PROJECT:MASTER

Mikkel Blytt, Spring 2014

NTNU
Norges teknisk-naturvitenskapelige universitet

Fakultet for ingeniørvitenskap og teknologi
Institutt for produktdesign

Masteroppgave for student Mikkel Blytt

Utvikling av digitalt spill for økt forståelse av matematiske konsept

Development of digital game for improved understanding of mathematical concepts

Bruken av spill som et verktøy i læringsprosessen er ikke et nytt fenomen. Spill kan motivere og vekke interesse for å lære om ulike tema og konsept, men også bidra til en økt forståelse av disse. Utstrakt bruk av nettbrett og smarttelefoner, allestedsnærværende databehandling (ubiquitous computing) har gjort tilgangen til slike spill mye enklere via for eksempel "App Store" og "Google Play".

Med bakgrunn i personlig interesse for spill, spillutvikling og matematikk vil jeg utvikle et digitalt spill som har til hensikt å gi spillerne en økt intuitiv forståelse av forskjellige matematiske konsept på et ungdomsskolenivå. Det følger av dette at spillkonseptet skal være allment tilgjengelig og morsomt å spille for flere aldersgrupper, og ikke være et rent undervisningsverktøy, men heller et supplement. Utformingen av spillet skal derfor gjøres ved hjelp av teori om læring og undervisning, men også ut fra teori om spilldesign. For å komme frem til best mulig løsning skal det følges en interaksjonsdesign-prosess med sterkt fokus på brukerinvolvering.

Oppgaven vil innebære å undersøke teori for spilldesign og læring, og krever hyppig kontakt med målgruppen forut for og under utviklingen av spillkonseptet. Ut ifra dette skal det utformes et digitalt spill som er "scope complete", men ikke optimalisert fra et programmeringsmessig perspektiv.

Oppgaven vil blant annet omfatte:

- Informasjonsinnhenting og teori
- Papir og digital prototyping
- Omfattende brukertesting
- Test av endelig løsning på målgruppe
- Endelig spillbeskrivelse knyttet til teori
- Presentasjon og endelig prototype

Oppgaven utføres etter "Retningslinjer for masteroppgaver i Industriell design".

Ansvarlig faglærer: Trond Are Øritsland

Utleveringsdato: 17. januar 2014
Innleveringsfrist: 12. juni 2014

Trondheim, NTNU, 17. januar 2014

Trond Are Øritsland
ansvarlig faglærer

Casper Boks
instituttleder

"Tell me and I forget, teach me and I may remember, involve me and I learn."

- Bejnamin Franklin

Table of Contents

Figure 1. *ELEMENTICS, final prototype icon.*

Sammendrag

Målet med denne oppgaven var å utvikle et fullt fungerende spill for nettbrett som skulle være morsomt og engasjerende for en bred målgruppe. Spillet skulle også ha for sekundært mål å gi spillerene en bedre forståelse av matematiske konsept via spillopplevelsen. Spillet skulle derimot ikke oppleves som et læringsverktøy, og læringen fra spillet skulle komme gjennom vanlig spilling uten å trekke ned på spillopplevelsen.

"ELEMENTICS" er resultatet av denne oppgaven. Dette er et spill basert på sannsynlighetsregning og kombinatorikk på et ungdomsskolenivå. Spillet utnytter læringsteori, spilldesign teori og matematikkpensumet fra ungdomstrinnene for å bygge spillerns forståelse av disse konseptene gjennom spilling. Dette er blitt gjort gjennom en sammenfletting av det relevante matematisk pensum og engasjerende og interessante spillmekanikker. Det er også blitt arbeidet med å gjøre spillet tilgjengelig og spillbart uavhengig av forkunnskaper gjennom utvikling av grafikk og intuitive grensesnitt.

Denne rapporten omhandler prosessen og arbeidet som ligger bak utviklingen av et spillet "ELEMENTICS". Med dette menes hva som ligger bak spillets endelige utforming, hvilke grep som har fungert og hvilke grep som ikke har det. Den endelige versjonen av spillet har flere interessante aspekter både i forstand av spill som læringsverktøy og som spill alene, men har også enkelte problemer som gjennstår. Disse blir drøftet i denne rapporten og må sees på i kombinasjon med den endelige prototypen. Denne rapporten omhandler også de tekniske utfordringene som ligger bak utviklingen av spill, da dette representerer en stor del av arbeidet som har inngått i denne oppgaven.

Preface

This report along with the final prototype of the game “ELEMENTICS” make up the work done by me for my Master’s Thesis (Spring semester 2014, Industrial Design, NTNU). The goal of the project was to create a digital game that delivers a fun and engaging experience, and secondarily builds the users’ understanding of selected mathematical concepts through gameplay.

This report documents the process and theory that went into the creation of the game, an exploration of the game in its final state, how the game was received in user tests, how the game compares to the project goals, and my reflections on the game and the process as a whole.

The accompanying prototype represents the final iteration of “ELEMENTICS”, a A 2D Puzzle-Strategy game for tablets. The game is based on probability and combinatorics at a middle-school and early high-school level.

The direction of this project was self-motivated, based on my desire to learn programming, explore digital prototyping, and to learn about game design and development. The “building an understanding of mathematical concepts” part of the project goal was added primarily to give the project a direction from the get-go, and have some more quantifiable measures than the ever elusive “fun” to compare the final result against. Having no prior experience with game or other software development (except from the perspective of UX and Interaction Design) resulted in much of the effort going into just learning the coding and technical skills required to create the prototype.

My tutor for this project has been Trond Are Øritsland.

Thank you for reading,
Mikkel Blytt

Project Goal

Note that the text below is a free translation of the approved description of the thesis. The original copy (in Norwegian) can be found in the appendix.

PROJECT DESCRIPTION

The use of games in the teaching process is not a new phenomenon. Games can motivate an interest to learn more about various concepts, but also by themselves, through gameplay, increase the players understanding of these. The extensive growth in the use of smartphones and tablets, referred to as ubiquitous computing, has made access to these kinds of educational games much easier through platforms such as Google's "Android Play" and Apple's "App Store".

With a personal interest in games, game design, and mathematics serving as a personal background for the project I want to develop a digital game where the purpose is to improve the players' intuitive understanding of different mathematical concepts. These concepts are to be based on mathematics on a middle-school level.

From this it also follows that the game should be accessible and fun for multiple age-groups, and is not to be a teaching tool but rather a potential supplement. The design and development of the game is to be informed by learning theory, especially game-based learning, as well as theory from the field of game design. The project will follow an iterative design process with a strong emphasis on user involvement. Frequent user interaction, both prior to and during the development of the game will be necessary to ensure a solid game design. All of this will result in the creation of a "Scope-complete" game, albeit one not optimised from a technical perspective.

In summary, the project will require: Acquisition of information, creation of paper prototypes, extensive user testing, linking theory and gameplay, presentation and development of a finalised prototype. The project will be done in accordance to the "Guidelines for Master Theses in Industrial Design" and the tutor will be Trond Are Øritsland.

PROJECT GOALS

Design is a dynamic process. Since the project description was written prior to carrying out the project it was of course the subject to some change. Certain aspects of the task were more challenging and required more work than others, and also the reverse. Nonetheless, the core goals of the project stayed more or less the same throughout the process and can, in order of importance, be summarised as follows:

1. The primary goal of the project is to create a fully functional digital game, designed in accordance with theory on game-based learning and game design.

2. The game is to deliver fun and engaging gameplay. It is not to be targeted at either gender and should appeal to multiple age groups.

3. The game is to revolve around set mathematical concepts, and through repeated gameplay the user should get an improved understanding of these.

4. The game design and game development process is to utilise extensive user involvement and user testing at all stages.

5. The final prototype is to be fully playable, omitting no key feature of the game, but optimisation and minor add-on features not crucial for the gameplay will be implemented as time allows.

On a personal level, the goal is to acquire skills in software development (in this case, game development), which I hope to be a great asset in working as an interaction designer. Having no prior knowledge of coding will mean that the scope of the game has to reflect this. Also, of the challenges in this project, making the game "work" will no doubt be at the forefront. Secondly, this is also an opportunity to work with a project from start to an actual finished product (to a degree), as opposed to working exclusively on a certain stage or with a certain area of a bigger project.

"That's what games are, in the end.

Teachers. Fun is just another word for

learning."

- Raph Koster (2004)

I:GROUNDWORK

Definitions

In the field of game design, game development, learning theory, and pedagogy there are several terms that are either unique to their respective fields, or are given a different meaning when used in the context of the field. Below is list of terms that is used in this report, and a short explanation for each on what they mean in this particular context. Several of these terms are explained in more depth later in the report in addition to being listed here.

GAME BASED LEARNING TERMS

Game-Based Learning - *Learning through the act of playing games. Learning can be, but doesn't have to be, the intent of the game for it to be considered game-based learning.*

Edutainment - *A contraction of education and entertainment. In the context of game-based learning it is often noted as shallow in that it only offers teaching rote skills through repetition, or it is used interchangeably with serious games.*

Serious games - *Games where the primary focus is to educate or convey a message to the players through the use of game mechanics, making the game itself the means to a goal.*

Gamification - *The application of game design and game mechanics to non-game settings in order to improve motivation, efficiency or retention.*

Exogenous and Endogenous Fantasy - *By fantasy is meant the "fantasy" elements in a game such as audio, visuals, role-play. Endogenous fantasy is fantasy that is integral to the game's content, while exogenous fantasy is added on top usually to further "gamify" educational content. Prevalent in literature for serious games, but has later been refuted as a key to game-based learning (Ainsworth, Benfort & Habgood, 2005).*

Flow - *A mental state that is characterised by full immersion in an activity. In gameplay achieved through increasing mastery of the activity where the challenge and player skill maintains parity with both increasing (Greitzer, Huston & Kuchar, 2007).*

MATH-RELATED TERMS

Combinatorics - *In this project limited to enumerative combinatorics. This is the study of how patterns can be formed in terms of combinations or permutations.*

PISA - *Programme for International Student Assessment. An annual international study that maps the performance of 15 year old students in maths, reading and sciences.*

STEM - *An acronym for the disciplines of sciences, technology, engineering and mathematics. Competence in STEM disciplines is important in an increasingly technological society.*

LEARNING THEORY

Bloom's Taxonomy - *A 6-step hierarchical framework for how we learn and think. Divided into higher and lower order learning / thinking.*

Higher Order Learning - *The three upper steps in Bloom's Taxonomy. In successive order: Analysing, Evaluating, and Creating. These represent the higher levels of learning. Each prior step must be achieved to "move up" the hierarchy so to speak.*

Lower Order Learning - *The three lower steps in Bloom's Taxonomy. In successive order: Remembering, Understanding, and Applying. These are the lower stages of learning. Game-based learning is often criticised for not reaching the higher levels of learning.*

Metacognition - *The knowledge of one's own cognitive processes. Self-reflecting or self-questioning. For a game related example once can look at how players reflect and iterate their strategic approach to the gameplay. Metacognition is important to learning and problem solving.*

GAME DESIGN

Story - *The narrative, or story, in game design is much the same as in storytelling. It is the story that the game tells through gameplay, through events. This can be linear, branching, predefined or even emergent. One of Schell's four elements (Schell, 2008).*

Mechanics - *The framework of rules that the players have to work within. One of Schell's four elements.*

Aesthetics - *The looks and feel of the game. This is what allows you to express the game's story and function. One of Schell's four elements.*

Technology - *The materials and tools that enables gameplay. The technology is the medium of the aesthetics and mechanics of the game. One of Schell's four elements.*

Top-Down & Bottom-Up Design - *As taken from Mark Rosewater, current lead-designer of Magic: The Gathering (Rosewater, 2003). Top-down in game designs is where the mechanics are created from the narrative one tries to convey. While in bottom-up designs, the mechanics come prior to the narrative, which can result in the latter feeling tacked on.*

The Thin Zone - *Closely related or interchangeable with flow. It's the zone where the challenge vs. player skill is ideal. When in the thin zone, an increase in challenge results in player anxiety, while a stagnation or decrease results in player boredom.*

Fun - *Fun in games and gameplay is defined as just another word for learning (Koster, 2004). Koster further explains how fun is the feeling that arises out of mastery and comprehension.*

Game Atoms - *The series of feedback loops that make up a game. Each loop consists of input, action, feedback , and mastery and represents it's own mini-game. An useful framework for analysing and talking about games. Taken from Raph Koster's "A Grammar of Gameplay" (2005).*

Affordances - *Also known from the field of interaction design. Affordances means the perceived and actual properties of things, especially as it relates to how they can be operated. (Kaptelinin, 2013)*

GAME DEVELOPMENT

Unity - *Unity is a game-engine with an integrated IDE. Unity allows for creation of video games for multiple platforms and enables easier game creation by easy to use drag-and-drop functionality and inbuilt features such as physics, scripting, rendering.*

C# (or C Sharp) - *A programming language developed by Microsoft. Supported as a scripting language in Unity. This was the scripting language used in the creation of the game ELEMENTICS.*

IDE - *Integrated development environment. Assists in software development by providing automation tools, debuggers, and code compilers. Unity features it's own IDE.*

INDIE DEVELOPMENT - *Video game developers without extensive financial support. A large community has emerged around the Indie development scene in the recent years. In part based on Indie successes such as Minecraft and tools such as Unity and GameMaker which make hobby-developers capable to create more complex games than ever before.*

POST-MORTEM - *In game development, post-mortems are reflections done after the release of a game. This report could be construed as one large post-mortem. In this report it is the title of the "reflection" section.*

A Game?

In the book “The Art of Game Design: A Book of Lenses” by Jesse Schell (2008), Schell examines several definitions of “play”, “game”, and “toy”. In the end, Schell comes up with his own definition of “game”:

“A game is a problem-solving activity, approached with a playful attitude.” (Schell, 2008)

This definition is of special interest to this project, since it defines games and gameplay both in the terms of intent (a playful attitude), as well as content (a problem-solving activity). If one fits this definition of games to the problem the game designer is faced with, namely creating a game, would it entail that the a game must be designed to feature problem-solving content and to be full of fun? That is, do they have to be entertaining first and foremost? On the face of it, this would appear to invalidate the idea of a games as a vehicle for teaching, or for imparting a message or explore other areas.

Ultimately, any definition of “game” matters little except where it can be used to inform the creation of “better” games, but it is interesting to see that in Schell’s definition the game and the fun has to be central. This is in direct contradiction to “serious games”. In serious games, the education the game tries to impart is primary, replacing entertainment as the main focus (Sisler & Brom 2008). In serious games, the play is the method for developing new skills.

In this project, the goal is to design a game more according to Schell’s definition, rather than one in the vein of serious games. But it isn’t a dichotomy, but rather a spectrum of where to focus the design and what content to feature. All games teach something to their users after all, even if nothing more than just how the game is to be played.

These two things “games” and “serious games” are not the full extent of the interplay between learning and games, but serve as an useful spectrum on where “play” is balanced with “learning”.

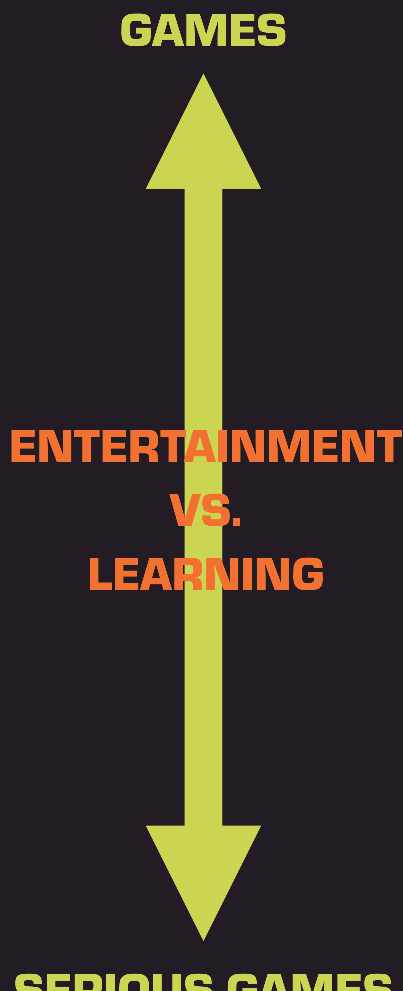


Figure 2. *Where to focus a design. A spectrum rather than a dichotomy.*

ON EDUTAINMENT

In this report, serious games is used over edutainment, as edutainment while often taking the form of games can extend to a much wider range of media, and in the literature edutainment when referring to educational games often means games that try to teach rote skills through repetition, and not higher order learning (Charsky, 2010). From other definitions, edutainment and serious games are interchangeable, or where serious games is but one type of edutainment out of several.

Games & Learning

Learning theory is an extensive field, where research is being done to model frameworks that explain how we as humans absorb, process and retain information as a part of learning. The field is multidisciplinary, drawing from neuroscience, psychology, anthropology, and more.

There is extensive research on game-based learning that is looking at how game design theory can create a better learning experience, and how learning theory can be incorporated in game design for the same purpose. By reviewing articles from the field, based on a searches in article databases such as SCOPUS and Google Scholar, the first thing that becomes apparent is that there is no consensus "best way" for how to design for game-based learning. There is even criticism of the very concept. Game-based learning is a doomed endeavor that will diminish our capacity to actually learn (Okan, 2003), or that it will not require the development of metacognitive strategies (which can loosely be identified as thinking about thinking, or thinking about why and what you learn). In a systematic literature review of serious games, by Boyle et al. (2012) it is concluded that there is a persistent difficulty in classifying the actual learning outcomes of the evaluated serious games, but not that the results are nonexistent.

In picking findings from the research, several important components to game-based learning are identified. These will help identify and sort ideas and features during both the early and later stages of the game design process.

TYPES OF LEARNING

All learning is not equal. In the revised version of Bloom's Taxonomy (Forehand, 2005) the cognitive domain, how we think, learn and reflect on a given topic, is split in two categories: Lower-order learning (or thinking) and higher-order learning. These represent different stages of cognition, in a hierarchical manner. A higher stage represents a better level of mastery of the topic.

The three lower order levels (in order) are: Remembering, Understanding, and Applying. The higher three are: Analysing, Evaluating, Creating. Their hierarchical relationship can be seen in figure 3.

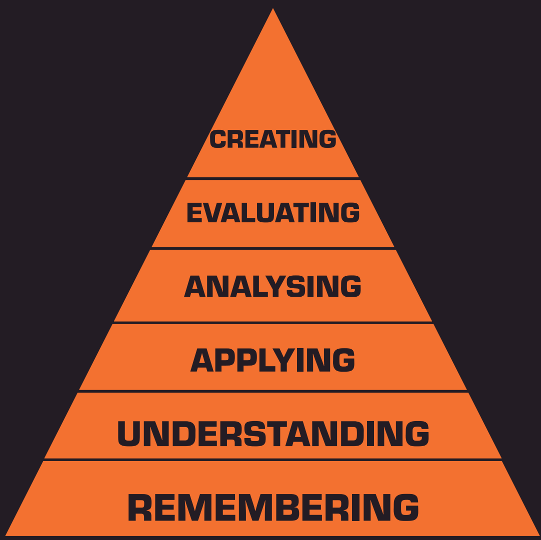


Figure 3. *Visual representation of Bloom's Taxonomy.*

While each of the steps are necessary to the mastery of a given topic, a game designed to teach would do well to consider it's scope and limitations. Gating content, progressive difficulty, or reducing the limitations on the player's actions over time are all examples of ways that several of the steps can be reached.

In typical gameplay for more complex games, the steps are integrated in the game design, often in the ways mentioned above, or in terms of a gameplay tutorial. This is done so that the players can slowly learn the game instead of being alienated or intimidated by the complexity. Some educational games might be happy only working in the lower steps, just teaching muscle memory, or rote-skills for a specific context.

Player Motivation

Motivation is a key factor in learning, and it is why educational games have existed for a long time. Good games capture both extrinsic and intrinsic motivational factors. Good games are fun, and having fun is motivating. This is the idea that has driven research into edutainment and serious games. That the use of games in an educational context will result in more motivated students, resulting in improved education and learning.

As described by Andersen (2012), a motivated learner focuses on developing, understanding and mastering knowledge. Which again results in enthusiasm and pride in their achievement, serving as a positive feedback cycle. If one compares this to Bloom's taxonomy, it is apparent that the motivated learner reaches a the upper levels of the pyramid. In comparison, someone that is not motivated might not bother going further than the lower stages. This motivation is split into extrinsic and intrinsic factors. Intrinsic, as in motivation that originates from within the individual, and as the opposite of extrinsic which is motivation that originates from outside influence.

Games, through their implementation, can create both kinds of motivation. By allowing exploration or self-expression, a sense of mastery and accomplishment, a game can create intrinsic motivation. While extrinsic motivation can take the form of social interaction (about or within a game), or simply competition, the latter which has always been common in the way of leaderboards or highscores even for games designed for single-player only, and the former which can be seen increasingly in the age of social media (Charsky, 2010).

Charsky (2010) explores why games, despite being such a great potential source for motivation in themselves, are difficult to use to educate. He argues that some educational games are nothing more than simulations, which are not the same as games (despite there being an overlap) and don't offer the same motivation, because they don't apply game characteristics. Compared to Schell's definition of a game, where these simulations could easily be considered games (solely on intent on part of the player, such as people "playing" with aircraft simulators as opposed to using them with the intent to train professional skills). Charsky argues that games must have a more "rigid" structure by using one or more out of several game characteristics to make it more than just free-form activity. Further, that only through blending these game characteristics, defined as: fantasy, choice, rules, and competition, can serious games help in motivating the higher levels of learning.

This idea of motivation in games, especially as it relates to serious games, is explored by Greitzer, Huston, and Kuchar (2007), resulting in an adapted Maslow's Pyramid to identify a hierarchy of the players' needs. This figure is seen below. The lower steps must be in place to derive enjoyment and thereby motivation and Greitzer argues that this pyramid also applies to "regular" game design. Fulfillment of the steps results in increased and maintained motivation for the players.

Figure 4. *Hierarchy of the player's needs, from Greitzer, Huston & Kuchar (2007).*

Serious Principles

This project's goal is to create a game designed to improve the understanding, even if just through implicit learning, of mathematical topics. However, in order to design the game in such a way that any game-based learning is at least guided through conscious and careful implementation, then looking at design principles from research into serious games reveals important considerations. What follows is by no means an exhaustive list, but instead a list of reoccurring principles identified in the literature. Some deal with the process of learning, while others deal with managing motivation.

THE KEY PRINCIPLE

The key design principle for serious games, as identified in the vast majority of the reviewed papers, is that the educational material has to be integral to the game. It can't be a separate element added on top of gameplay.

COGNITIVE PRINCIPLES

Greitzer, Huston, and Kuchar (2007) identify a series of cognitive principles they translate to design guidelines for the creation of e-Learning and training applications. They are as follows:

Stimulate Semantic Knowledge - *By which they mean that the material should relate to the learner's existing experiences and knowledge to facilitate further learning.*

Manage Cognitive Load - *Material should be broken down in smaller chunks, building up gradually to more complex concepts.*

Immerse in problem-centred activities - *Allow the user to immediately work on problems related to the material.*

Emphasise interactive experiences - *Encourage engagement and interaction with the material. This allows for higher order learning.*

Engage the learner - *This is done by maintaining the learner's Flow state, or "Thin Zone", where the challenge and learner's skills are proportional.*

VIDEO GAMES AND MATH

Young et al. (2012), is a meta review of trends in serous games for education. In their paper, they identify the trends on a subject basis, including mathematics. For serious games for mathematics they identify several constraints or concerns, they are paraphrased here as design principles or pitfalls.

Learning outside gameplay vs. learning within gameplay. *Players dislike it when they identify learning activities as such instead of as just more gameplay, this is in line with they key principle identified previously.*

Learning situated in gameplay. *This is the optimal situation, as players enhance their efforts and performance when committed to the gameplay.*

Gaming without reflection. *When players lack a reflection process it inhibits the learning process. To reach the higher orders of learning such a process must be encouraged.*

Boys versus girls. *When communication is facilitated, boys tend to focus on game-related conversations, while girls tend to emphasise and enjoy the social interaction especially.*

OTHER PRINCIPLES

In their presentation, Chan and Howlin (2007) identify that serious games can not lean on the fact that they are educational, but must produce engaging gameplay on the level of other commercial offerings, and that serious games must evaluate if the educational content they are offering is valid.

Much of the research into serious games considers how these games are to be used by educational institutions, and how to use them in the current educational framework. This project is not concerned with the use of games outside that of free-time gaming, and as such this research falls outside the scope of this project. Next we will look at mathematics, to identify interesting fields to consider for the game design.

Mathematics

The choice of mathematics as a subject area for this project was done because games and mathematics are closely linked. Most games, when abstracted and stripped of their narrative can be reduced to a series of mathematical expressions. Balancing gameplay is often a matter of adjusting the numbers of the underlying equations, which sounds easy on the surface but is a very nuanced and complex process. All of this is especially true when considering digital games, as is the case for this project, since these have to be expressible in code.

Undertaken before starting the project, as can be seen in the project description, was the choice to feature mathematics on a middle-school level. This choice was done without prior research, because it represents the most approachable level of mathematics to the widest range of users while still allowing for some complexity. Instead of creating a game centred around highly specialised mathematics it is more interesting to examine the foundations. Middle school mathematics also represents the last years where the curriculum is shared for all students (in Norway), and it also happens that the overall proficiency of the students is tested annually by PISA.

The full mathematical curriculum for 8th to 10th graders, as well as the first year high-school curriculum for both practical and theoretical mathematics can be found in the appendix.

CONSIDERED TOPICS

The game "Dragonbox" will be explored later in this project as it is a relevant example to the project. This is a game that teaches algebra and has achieved both commercial and critical success. In order to avoid constantly having worry about the comparison to this game, a conscious decision was made to avoid algebra as a topic for this project. Instead, the following four areas were considered to be of special interest:

Math in the "daily life" - *In the curriculum, students are to learn about real-life applications of mathematics, especially as it relates to economics (income, loans, interest, budgets etc.).*

Probability, Statistics and Combinatorics - *Students are to learn about probability, sample spaces, and basic combinatorics. In the first year of high-school this expands to dependent and independent events, and even binomial distributions.*

Geometry - *Students are to learn properties of two, and three dimensional structures and how these can be constructed. They are also expected to learn the use of coordinates and to experiment with logic based on geometrics. In the first year of high-school, basic trigonometry is introduced.*

Derivation and functions - *Derivation, and the identification of local minima etc., is first required at a first year high-school level. In middle school, students are expected to be able to draw and identify practical applications of functions.*

To inform the choice of topic for the game design, both PISA and an informal study was considered.

PISA

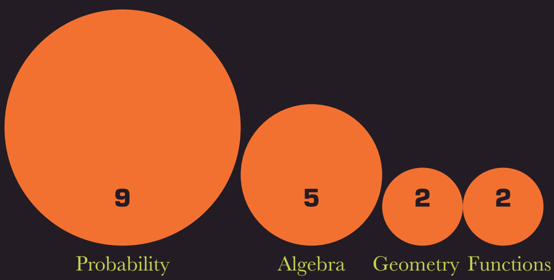
Norwegian students, in the 2012 results, performed around average in mathematics, but showed a decline from the 2009 assessment. Further, PISA shows that the amount of "top performers" in mathematics in Norway is less than the OECD average. OECD is an international economic organisation of which Norway is a member, and it is natural to compare the performance of Norwegian students to those of the other members. Out of 34 members, Norway ranks 22nd in maths.

For mathematics, PISA tests four overarching concepts that relate to numbers, algebra and geometry. These four are: quantity, space and shape, change and relationships, uncertainty and data, and they each primarily relate to basic computation, geometry, functions and algebra,and probability and statistics respectively. Norway performs below average in change and relationships, but above average in uncertainty and data. (All data is taken from the PISA 2012 report)

INFORMAL STUDY

To further inform the choice of topic for the eventual game design. A small questionnaire, where participants were asked in person, or through a form distributed in social media, was conducted. The full questionnaire can be found in the appendix, but the results are shown in figure 5.

HARDEST TOPIC TO UNDERSTAND, MIDDLE-SCHOOL MATHEMATICS:



HARDEST TOPIC TO EXECUTE, MIDDLE-SCHOOL MATHEMATICS:

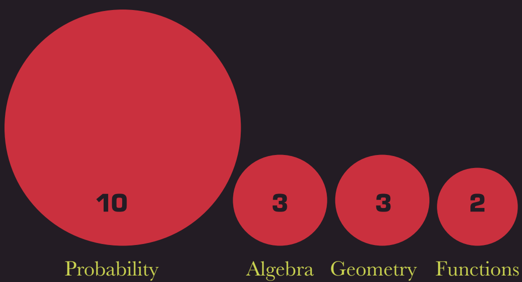


Figure 5. *Visualization of two of the study questions regarding math and difficulty.*

It is interesting that the PISA results correspond well to those of the questionnaire, except that here probability was identified as the most difficult subject. Since this is an area where Norwegian students performed above average, this appears contradictory. If this represents a change in the curriculum, a shift in educational focus, due to the small sample size, or just that the perceived difficulty for this topic is higher than the "actual" difficulty is hard to tell.

On a second note, when asked the same questions about first year high-school mathematics most answers remained the same, though derivation was a the second most frequent mention behind probability.

Shortening polynomial expressions, understanding the real-life application of derivation, understanding and modeling probability outcomes, and understanding the difference of independent and dependent events in probability were all emphasised by more than once test participant as particularly difficult.

CHOICE OF TOPIC

Of the four topics, only probability, daily math, and derivation and functions are going to be considered going forward. The exclusion of geometry, as a possible base for this project has been excluded on the basis that few participants mentioned it in the informal study, and because of perceived difficulties with the implementation of a digital game based on it. The need to draw, possibly move, and dynamically change and alter forms on the fly and comparing these shapes against each other is more difficult to do with a digital application, rather than say a board- or drawing-game. Also, a quick search for geometry based games reveals a wide range of existing concepts, from physics based puzzles to free-form exploration of geometric forms and properties.

The three remaining topics are all to be explored in the early stages of the game design process, resulting in the creation of a game design concept linked to each respective topic. Combining any of the topics is not going to be considered, in order to limit the scope and to keep the design vision "pure" in terms of what content the game will attempt to create an understanding in. Each topic already covers a wide range of areas, sufficient for any number of game mechanics.

GAME DESIGN

Like other fields of design, game design has no set of unconditional rules. This is not to say guidelines, grammar and frameworks for working with game design have not been established. These are tools for that help in the creating games, not unlike similar tools are taught and used in fields such as interaction or product design. Game design today is even taught at university levels, as is the case with game development, since the fields (jointly dealing with the creation of games) have grown more complex and mature. Forming the backbone of game design theory used in this project are three main resources:

Jesse Schell's "The Art of Game Design: A Book of Lenses"

Raph Koster's blog and book "A Theory of Fun for Game Design"

Gamasutra, a webpage (running since 1997) dedicated to game design and development featuring content by professional and aspiring amateurs in the fields.

Using these established frameworks and grammar for game design makes it easier to break down and explore the inner workings of games and the design process itself. Through the use of these, the process is moved to a level where individual aspects can more easily be tested, analysed, discussed and altered accordingly. Following is a short summary of areas that were especially important to the design process

THE FOUR ELEMENTS OF GAMES

Schnell's four basic game elements are visualised in figure 6. The four elements, Aesthetics, Mechanics, Technology, and Story, work together to create a game. They must work synergistically in order for a game to be fun, and neither one can be ignored without bringing the whole game down.

The diamond they form in the figure also shows how players experience them, showing how "visible" they are during gameplay. The aesthetics of a game is of course the most visible element, and it is with the aesthetics of the game that players interact (through GUI elements, in-game avatars or otherwise). Technology, at the rear of what's visible, is what ultimately enables the interaction, allowing the game to run. The choice of technology, such as the platform (or even just dice versus playing cards for that matter) informs the other elements and vice versa.

Mechanics separate games from other media. Mechanics, representing the rules and procedures of a game, make it so that games are unique when compared to books, movies or other linear media. A game can be linear in terms of story, but it is driven forward by interaction with the player(s). Lastly is story, which enhances the player experience by giving shape to the events that unfold during gameplay.

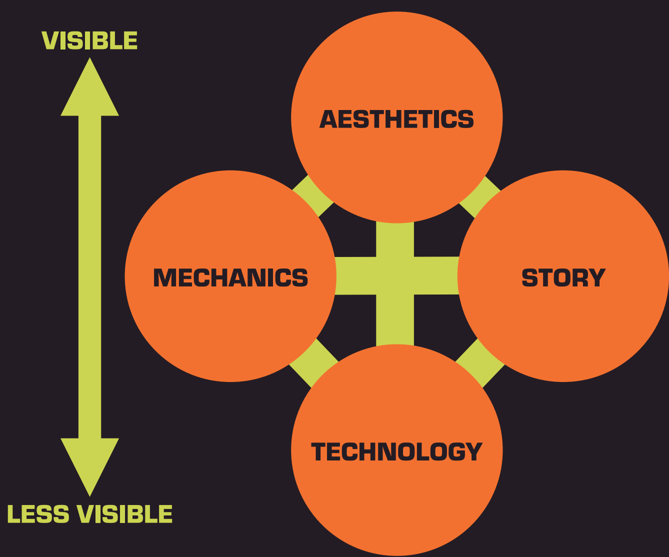


Figure 6. *Four elements of games (Schell, 2008)*

Schell's figure, while showing that the four elements are connected, doesn't tell about how they can or are to be approached. Top-down or bottom-up designs, represented by Aesthetics and Story first or Technology and mechanics first respectively, can influence the end result tremendously.

Approaching the game from a set idea, or overall concept (such as a story or feeling the designer wants to tell / invoke) can be hard to fit within the a game, but ensures an organic relationship between the "hard" and "soft" elements. An interesting game mechanic, or a type of "problem", can grow into a full-fledged game. This runs a greater risk that the story and aesthetics feel tacked on and arbitrary to the game, which weakens the game compared what it could be if the elements of the game make it more than just the sum of parts. The idea of top-down / bottom-up design for game design presented here is an adaptation from Mark Rosewater (Rosewater, 2003), lead designer of Magic: The Gathering, combined with Schell's four elements.

LENSES OF GAME DESIGN

Possibly the most important resource from Schell's book is the more than hundred "Lenses" included to help reflect on a game. These take the form of questions to ask yourself, when designing or play-testing, about the game. The questions are open ended, but cover topics that are closely linked to the hierarchy of player's needs (figure 4. Greitzer, 2007). These are used in the game design process to help reflect on "why" something is or isn't working in conjunction with Koster's model of Game Atoms.

THE ELUSIVE FUN

Fun is often the ultimate goal, on part of both player and game designer. What happens when the focus is instead on something has been explored earlier. Creating a fun game is the one of the primary goals of this project.

"A Theory of Fun for Game Design", by Raph Koster, delves into why games are fun, and what fun means to games and gameplay. He explores the rather sweeping statement that for games: "Fun is just another word for learning" (Koster, 2004). According to him, fun is had when a feeling of mastery is achieved. This is why gameplay ideally resides in the upper levels of the thin-zone, breaking into the area of "anxiety", where the challenge exceeds the skills of the player. The fun is found where the player is challenged slightly above his or her abilities, and is then allowed for a sense of progressive mastery over the challenge. The game is a model that challenges the player , but allows for mastery. So we play and learn, but once we exhaust the challenges that the model provides it becomes a rote exercise instead. In a sense, it stops being a game and the fun ceases. Also, the promise of mastery is not enough, which is how Koster explains why people aren't always motivated to learn, and why games can be too difficult. It is important to note that these ideas Koster speaks off are not objective, but instead subjective measures, experienced by the players themselves.

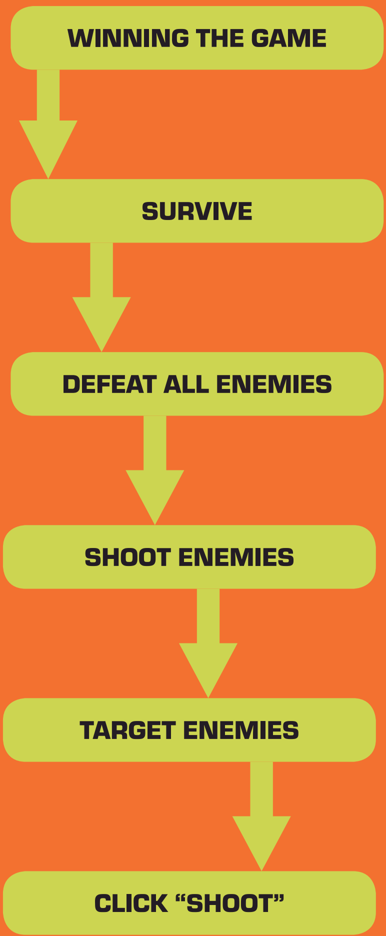
GAME ATOMS

Going further with this view, the question becomes how to create fun. For this, game atoms is a way of breaking a game into basic components to m make it easier to analyse, discuss, and to "debug" when something is or isn't not working.

In essence, all games follow a looping sequence consisting of input, action (or response, resulting from the input), feedback and mastery. Game atoms are what you find when drilling down into a game, finding these loops for the most basic components of the game. Most games feature multiple atoms, working in conjunction to form games of far greater complexity. By looking at a game as a series of nested loops or as fractals each nest or fractal is a subgame on its own. The innermost loops can be just simple GUI actions (press a button to rotate a block, to use Tetris as an example). Each of these nests or subgames, have to satisfy several criteria in order for the game as a whole to be fun.

The criteria required for the game atoms to be fun and a visualisation of how these loops function is shown on next page in figure 7. The theory of Game Atoms is taken from Koster's "A Grammar of Gameplay" (2004) and his blog.

A HIERARCHY OF SUBGAMES (NESTED LOOPS):



GAME ATOM REQUIREMENTS FOR FUN:

- Does the challenge require preparation?
- Does this preparatory step pass these steps as well?
- Does the challenge allow for multiple ways to prepare?
- Does the environment for the challenge affect the challenge?
- Are the rules of the challenge defined?
- Can the rules support multiple types of challenges?
- Does the challenges require multiple abilities to pass?
- Is there skill involved in using the ability? (and if not, is it a fundamental move, one of the innermost "nests")
- Are there multiple success states to beating the challenge?
- Do advanced players not get a benefit from sticking to easy challenges? (referred to as the mastery problem)
- Does failing the challenge have a cost?

GAME ATOM COMPONENTS:

INPUT ▶▶ ACTION ▶▶ FEEDBACK ▶▶ MASTERY

1. The player makes an input in an attempt to beat the challenge posed by by the "subgame"

2. The game calculates a response to the input. Could also be titled "result".

3. Success or failure, the game gives the player feedback on their action. The feedback should be variable.

4. The player learns from the feedback, achieving a higher state of "mastery" for the challenge. The Cycle repeats.

Figure 7. *Model of game atoms, criteria for "fun", and nested loops of subgames. Adapted from Raph Koster's blog (2012). Not intended as a "recipe" for fun, but critical areas to consider.*

A Digital Game

Gaming on smartphones or tablets, dubbed "Mobile Gaming", has seen an explosive growth since the emergence of these platforms. While the games for hand-held consoles or early cellphones have existed for many years prior to the introduction of the smartphone, the rapid improvements in technology has changed this field immensely in short time. Mobile gaming is a multi-billion industry (Capcom.com, 2012), and gaming represents more than a third of all time spent using iOS and Android devices, a figure that only increases when considering tablets only (Skillz.com, 2013).

For this project, the choice to design toward tablets over smartphones was made taken before the design phase had begun. Further, the choice was also made to design towards the Android platform over iOS and Windows Phone. This was done for the following reasons:

Access - *To design for a smartphone I would have required the purchase of a new device, while I already had a tablet device that could support the design and development of a game. Android is also open for new developers without a more lengthy (and expensive) process to access their development tools necessary to export and test games for the platform.*

Affordances - *The affordances provided by the larger tablet device allows for a higher level of complexity than that of the smartphone. This is backed up with statistics showing that the average session length for tablets is longer than that of smartphones (marketingland, 2013). Designing "simple" games is not an easy task from personal experience. Based on experience from designing boardgames, and on conversations with mobile "gamers" for both platforms; the level of complexity in a game correlates to how much "imperfection" the game can contain while remaining fun. This does not excuse bad design, but allows for a bit more leniency and a larger window of opportunity for the game to "hook" the user.*

Multiplayer - *While not given that any game would contain either, the possibility to include synchronous or asynchronous multiplayer on the same device was also a point in favor of the tablet, where the larger screen is more easily shared between players.*

DESIGNED FOR WHO?

The project description already puts in place several qualifiers that limits the design space for the game. However, it is qualified in the project goals that the game should be enjoyable for multiple age-groups, irrespective of gender. While not catering to any group in particular might shape the game around my own preferences, and thereby to my "age" to a greater degree, the latter goal is of special importance. Mobile gaming statistics show that female gamers represent 45% of the market. In regards to age, the average gamer is 28 years old, of which 68% are above 18 years (Skillz.com, 2013).

Much more important than designing for a target group based on statistics however, is to playtest with users that are representative of the potential userbase.

THE MARKETPLACE

For mobile games, the choice of platform also decides the potential market. The digital stores "App Store" and "Android Play" for iOS and Android respectively are the two largest marketplaces, and by far the most important ways to sell and distribute games for these platforms. However, while the industry is both large and growing, most games released to these do not recoup the cost of development, and most of the income is limited to a handful of games out of many thousands. Also many games are released with no intention to of making money (as is the case for many hobby developers) which only toughens the competition.

In following the dominant advice in the indie development community, this project does not concern itself with the market, where only established players have a chance to influence the outcome. Echoed in indie development forums is a sense of "release and pray" when it comes to new releases and marketing, and the advice is always to focus on the game and gameplay, which is the line this project follows as well.

Example Games

A quote with a long history (and unclear origins) is "Good artists copy; great artists steal". Of course this is only true when the original is improved upon or explored in novel ways. For this project, a select few games warranted closer inspection, if not for inspiration then at least in order to explore how they relate to the framework and perspective this project is working from. The games DragonBox and Kerbal Space Programme are briefly discussed here.

DRAGONBOX

The Norwegian game "DragonBox" has received much media attention (see under DragonBox) for its claim to teach algebra to children, or adults for that matter. So effectively, it is a game that has for intent to teach mathematical concepts, just like this project.

In the game, players are confronted with puzzles of increasing difficulty. The screen is split into two areas, and the ultimate goal of every puzzle is to have the "DragonBox" stand alone on either side, in the fewest moves possible. Also, players are only given access to a few possible moves. As the game progresses, the "fantasy" and aesthetic elements are slowly stripped away, and in the end, the imagery of dragons and boxes is completely replaced with letters and mathematical symbols. The elements on either field also cease being strewn haphazardly about, and instead stand in line with a "=" symbol separating the two fields. In the final form, the game reveals that the player has been solving equations all along, using only the fundamental mathematical rules hidden through aesthetics and interaction.

In this framework, and also seen in the game's marketing, is that DragonBox is a serious game. It carefully follows the player through the pyramid of Bloom's Taxonomy up to the step of analysis, and also utilises several of the cognitive principles for designing serious games. DragonBox has experienced success, and is a clear example of how serious games can work. It targets a specific audience through both aesthetics and content, and the no time limit puzzle structure of the game allows the player to reflect on what he or she experiences.

On the other side, the game model is fully deterministic with no chance influencing any outcome. Also, the rigid and progressive model of content restricts the game to the upper middle levels of Bloom's pyramid. Without the ability for more free exploration and interaction with the "equations" does the game teach understanding or just the practical steps to solving equations? Either way, "DragonBox" is a success to take into consideration.

KERBAL SPACE PROGRAM

Kerbal Space Program (KerbalSpaceProgram.com) or KSP, is a "sandbox" game where the player manages a fictional space program, designs spacecrafts and flies them while adhering to mechanics that are simplified but very close to those of real world astrophysics.

Also as a success, praised for being educational, KSP is a game first and foremost. The teaching it offers is very much a secondary result of the aesthetics and mechanics being derived from where they are. Criticised for a steep learning curve and little in the way of in-game tutorials, KSP forces the player to learn and engage with the game mechanics. The game requires learning on the upper levels of Bloom's Pyramid in order to progress since the game only gives you the tools to reach the first few stages. The cycle of design, attempt mission, debrief, clearly matches that of the game atom.

In many ways KSP is closer to what this project hopes to achieve, though at a much reduced scope. Learning is secondary to gameplay, but very much central to it nonetheless. However, KSP is unforgiving. The lack of clear goals and hard to grasp feedback can kill player motivation. The game also builds on something foreign (for most), amplifying these issues. It is my hope to create something that is more "accessible" as suits the mobile platform.

Game Dev

Separate, but closely related to game design is game development. This is the act of creating games (note that the term game development is only used when dealing with digital games) from a technical standpoint. For larger games there are many game developers, working in conjunction with designers and artists. This project is more similar to small indie or hobby developers, where art, design and development all fall to one person (me).

Just like the consumption of games has increased, so has the number of people making them. This has resulted in large communities where knowledge is shared amongst users, especially surrounding specific "Game Engines". Game Engines are software frameworks that aid in the creation of games by giving the developer a "head start", in that the most basic components don't all have to be made from scratch.

UNITY

Unity was the choice of engine for this project. Not only does it provide 2D-tools (like texture , animation, and sprite handling), but also full featured scripting, smart drag-and-drop functionality, and a nearly unrestricted free-to-use license.

The Unity environment allows for rapid prototyping, even for novice users, because it is forgiving to sub-optimal practices due to the way it's set up. Also, the extensive resource library built around it by the both the Unity developers and its users allows one to quickly find answers to the most basic and complex challenges one comes across. However, the ease-of-use can result in unwieldy projects that are hard to manage and prone to bug. Ultimately, it was Unity that made this project possible, by allowing rapid deployment of builds (versions of the game) to the web-player. These builds were easy to distribute and play over the web, which allowed for running user testing in parallel with game development and design, and rapid integration of user feedback.

STACKEXCHANGE

Stackexchange is a group of web-pages dedicated to an question-answer format for specific fields. In particular the stackexchange pages Mathematica and Stackoverflow (dealing with math related and programming related questions respectively) were used to find answers and ask questions as they arose during development.

USER TESTING

For this project user testing can be considered to be split into two categories of testing: bug-testing and playtesting. Testing all the possible interactions between components and unique corner-cases that might arise as a course of gameplay is difficult. When unintended events break gameplay, the bug is critical and will compromise the player's enjoyment of the game. Finding and fixing these were therefore crucial to the development process. More conventional playtesting, for testing the actual gameplay; the user experience and interactions, was also used continuously throughout the design process. User feedback is crucial to designing a good product, and games are no exception. In the appendix you can find forms used for these user tests. Tests were done both with or without supervision of me and with varying degrees of "formality" depending on what the design required at the time.

Figure 8. *Unity logo. The game engine is used by amateur and professional developers both.*

Design Goals

With the theory in mind, it is possible to clarify the design goals for this project. It is the goal of these pages on theory, or groundwork, to give a picture of why the resulting designs ended up the way they did and what this project has entailed beyond the creation of the prototype. Listed underneath is a revised and more detailed set of design goals.

1. The game's primary focus is to deliver a fun experience. Designing the game so that the game atom criteria (as shown in figure 7.) are satisfied and with a clear emphasis on creating a strong unity between the four elements of mechanics, aesthetics, technology and story, will give a better foundation to work from and take further with playtesting.

2. The game's secondary design goal is to teach the user about one area of mathematics. In order to do this, the design principles for serious games will be utilised. Structuring the game's progression so that complexity grows naturally, and providing the player with ample feedback to reflect on are two such principles. The game aims to ultimately engage the player at the upper levels of learning in Bloom's Taxonomy.

3. The area of math will reflect relevant curriculum, at a level that is approachable by gamers from a wide age-range. Derivation and functions, math in the daily life, and probability and statistics make up the three potential topics.

4. The game will be designed to be gender neutral, and based on tablet gameplay. This involves considering game length and the affordances provided by the technology.

5. In a conflict between the first and second goal listed here, fun will take priority over educational content where the two are exclusive.

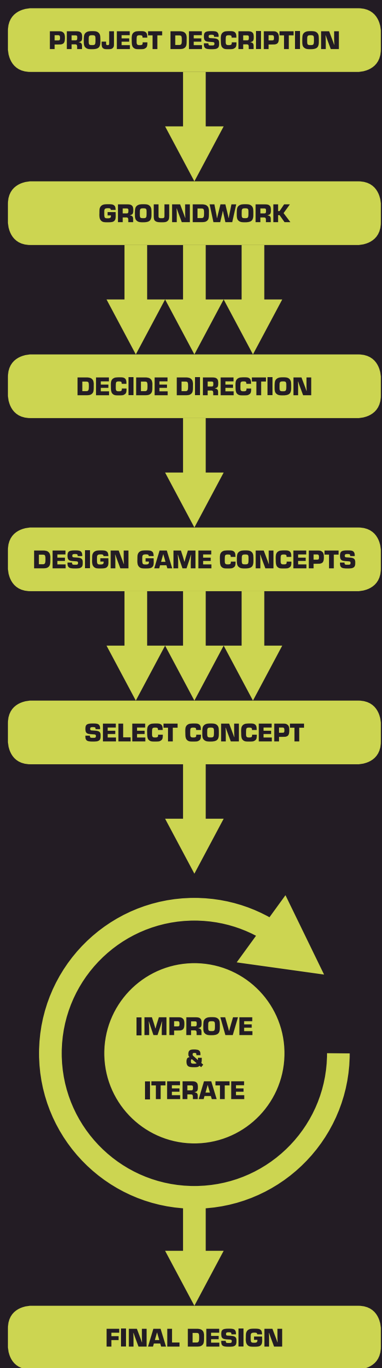


Figure 9. *Visualisation of the intended design process. Note that the process cycles between being divergent and convergent to best explore the potential design space.*

"The way to succeed is to double your failure rate."

- Attributed to Thomas J. Watson, IBM.

II:CONCEPTS

Early Stages

This and the following sections deal with the design of three separate concepts. It gives a truncated view on how these came to be and how they were tested and evaluated. Outside playtesting and informal conversations between my tutor and co-students, this work was that had to be done by me alone. Input on unfinished and unplayable ideas is hard to get and less valuable. Bringing ideas to life in terms of functional prototypes immediately adds more room for conversation and outside input and this was the goal of this early stage. Not shown here are the many ideas that never went beyond "doodles" in a sketchbook, where they proved to have critical problems (such as too deterministic, dominant strategies, or too removed from the mathematical concepts), making them non-viable.

INITIAL IDEAS

From the theory it is clear that any game will require the mathematical theory that it aims to teach be firmly embedded in the game's mechanics. Adding such content on top of non-relevant mechanics prevents player immersion, enjoyment and learning.

Another thing is that any game can't simply be presented as mathematical tasks just like they are in textbooks. This is not only doing a disservice to the capabilities of games as a platform, but will also keep players away. The central goal after all is to make players enjoy themselves, which is prevented if the "teaching" part of the game is too visible and distracting.

Using math as a basis for game mechanics results in a very much bottom-up style of design, where the aesthetics and narrative of the game are made around defined mechanics. This will require a careful hand when tying the "soft" and "hard" elements of games together.

To ensure this, the process cycled between a top-down and bottom-up view when brainstorming. First, when an interesting and relevant game mechanic was discovered, I immediately shifted focus to find a context or narrative that could support it. This again usually led to the discovery new potential mechanics, or additional features to the original mechanic. Then by shifting focus back to a mechanical level, I could again work to align and connect the narrative and mechanics.

THE GROWTH OF IDEAS

Often a promising idea for game mechanics grew when finding a suitable narrative or simply from reflecting on what might be "fun" additions to the gameplay. While several of these haphazard discoveries were interesting additions, they all to often strayed from the central premise in that they weren't linked to the relevant mathematical concepts. This was especially problematic when they then proved more interesting than the original mechanic, as this defeated the point of the exercise and proved the weaknesses of the original idea.

The game atom model was of great help in growing ideas organically, ensuring that each part of the emerging game at least functioned and had interesting aspects to consider. It also helped weed out the "bad seeds" that looked promising on the surface. Prominent problems were ideas that had issues with mastery, resulting in dominant strategies, or ideas that were too deterministic and were only thinly concealed mathematical problems that didn't have the room to capture a wide enough range of challenges. However, several ideas did emerge, off which three were taken further. One for each of the defined mathematical topics.

These three concepts are all presented here, and while they are playable to some extent, they were all rough versions. I didn't want to finalize too much of the design before moving to a digital prototype, but they still had to be playable to the point of being testable.

PROTOTYPING AND TESTING

One issue that I would like to highlight, looking back on the process, is how the early stages were never put to the test as digital games. All ideas were instead tested in the form of quick sketches and graphs, and then through a better defined paper prototype. Mostly, this was an issue of time. Paper allows for more rapid iterations and gets the conversation and feedback flowing much quicker. But also it was because in parallel to designing the game ideas I followed tutorials on coding and game development for Unity. At the first stages I did not have the capability to create rapid digital prototypes, at least not in a suitable timeframe. A third reason was an idea that by creating paper prototypes and then choosing the most promising direction to port to digital form I would have through the process of creating a functional paper prototype have made a game with a scope suitable for digital development.

Working strictly in a different media, analogue games, with the intent always being to end up with a digital game did affect the end result. While I did my best to explore the digital potential of each idea, something is always lost in the translation between media and something else is gained. As it happens, I think there were both strengths and weaknesses to the approach, and this is covered more in-depth under the "POST MORTEM" where I reflect more on the choices taken during the project.

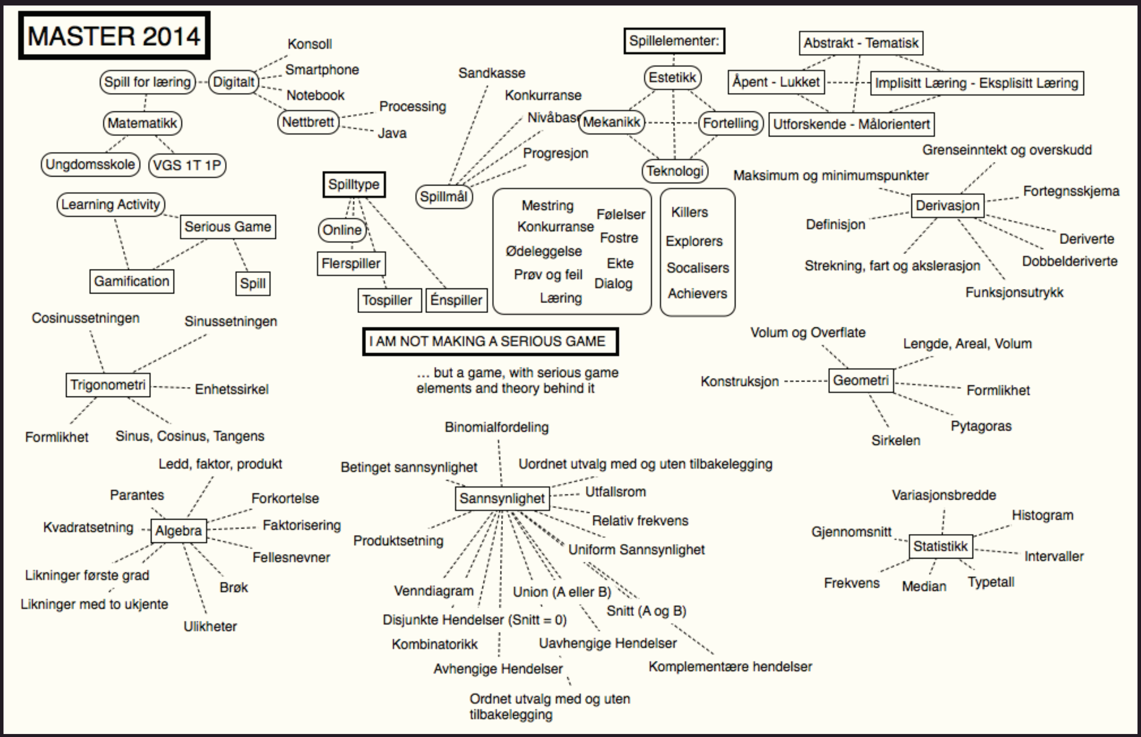


Figure 10. *Mind-map of math and some game design theory. Intended as a view into the earliest stages of the project.*

Concept I:

Functions and graphs are ways to express a chain of events and the relationships between the various variables involved. Being able to interpret these structures are important, and is what much of the middle grade curriculum deals with. The first concept that solidified deals with this area of mathematics.

Derivate Racer uses the narrative of a race to convey this relationship between variables. Designed to be a single or multiplayer game, the player(s) take on the roles of "racers" that try to get around a track as quickly as possible. Unlike in conventional race games, where the input - output (power to acceleration) is managed dynamically and continuously, requiring more hand-eye coordination on part of the user, the idea with Derivate Racer is to break it into clearly defined action-cycles. In other terms, a turn-based race game. This allows for a longer reflection and feedback period for each action and the formation of long term strategies over split-second decisions.

CENTRAL ATOM

The most central "Game Atom" in the game, is the management of speed vs. power. While most race games simulate a more or less realistic "input as acceleration" model, this game looks the relationship between the two more closely.

The player's piece, can not have its speed affected directly. But only through removing or adding "Power". As the turn then progresses, the power in the engine is added (or subtracted) from the current speed, and the player has to move the vehicle a length equal to the speed. This sequence shows the game atom as: Input - adding or subtracting power, Action - computing the resulting speed, Feedback - did the player crash, overtake another player, or hit a "special area" forms the central game atom.

While on the surface a rather simple decision, and it is for the short term (that specific cycle), but the complexity emerges in the calculation of long term outcomes. Because the change in speed cannot be adjusted directly, forethought is required on part of the player. This of course comes at a cost, and a potential area of frustration. The player can lose before actually losing if they aren't careful, and finding the deciding factor for the outcome is difficult.

To combat this area of potential frustration, and to break the game away from being too deterministic, randomness in the form of "field types" were added.

Field types explore altering the game rules, when the player is on them. Such as changing the direction, directly manipulating the speed, setting a speed limit, doubling the impact of power, or halving it are but some examples of how the game builds on the central game atom.

OTHER IDEAS

The game was tested as a multiplayer game, because it added another layer to the game by allowing players to test out the interaction of variables on someone other than themselves. This allows for "negative" effects to be enjoyable and worth incorporating into the player's overall strategy.

With the management of power and speed being done within the proposed constraints, the game can also build on the idea of derivation (and the relationship between distance, speed, acceleration) without actually requiring the know-how from players. This extends beyond the middle-grade curriculum, but is the natural "next stage" when considering the topics of functions, change, and relationships. To sever the connection with other "vector racers", this game does not concern itself with directions, leaving the navigation as a separate mechanic not tied to the underlying math. Next are the game's rules as it was played in the prototype stage.

Derivate Racer

PROTOTYPE GAME RULES:

Name: Derivate Racer
Tags: Turn-based, multiplayer, strategy, and resource managment.

Each player places their player piece on the indicated "Start Line", and sets their Speed and Power counters to zero.

Using the Power adjustment cards, each player decides on an action (+1 Power, -1 Power, + 2 Power, - 2 Power, No Change, or a "Field Card"). When each player has chosen, all choices are revealed and everyone updates their Power counter to reflect their action. Then, the power counter is added to the speed counter (subtracted if negative).

Each player has to move their piece in one direction equal to the length of their speed. If the player has to move into a "wall" or other player, they crash. Player's take their turns moving according to speed, the fastest go first and so on. If one or more are tied, the person most behind in the race moves first.

If a player "crashes" he or she has to reset both Power and Speed to zero and can only play field cards on the following turn. If a player is instead crashed "into", then that player sets their power to zero with no other effect.

Fields are divided into two types: Special fields, that give field cards when a player moves into the respective tile, and track fields, which are color coded and give a one-time effect according to the list below.

Green: On these fields, the player can change direction freely during movement, and not stick to the initial direction.

Red: The remaining movement is cut by two, and the Speed counter is reduced by two (down to a minimum of zero).

Yellow: Moving above the speed limit indicated on this field means that no change in direction can be made until the player crashes or starts the turn on a field not of this color. Example: The Speed limit is 4, and Jon hits it going 5, on the following turn, Jon can't change his direction from what he was going last turn.

Field Cards are used in place of changing your "Power" for that turn, and take place immediately when revealed, before players update their Power and Speed counters for that turn. Follow the instructions on the card.

Example cards: Reverse the intended Power change for this turn, Change the color of a "Track Field", or the targeted player can change power for one additional point until you use another Field card.

The game is played on a hexagonal tile board, and the track fields and field cards are distributed and shuffled prior to the game starting.

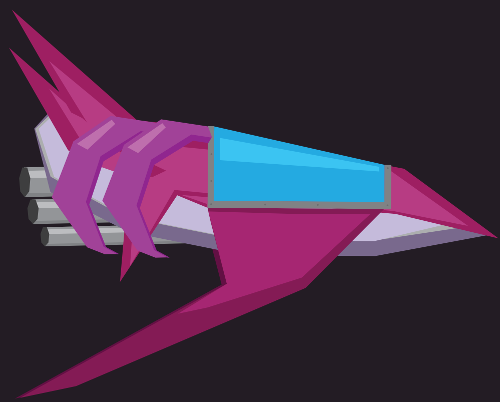


Figure 10. *Visualisation of the player piece. The figure was pasted onto thick cardboard and colored differently for each player.*

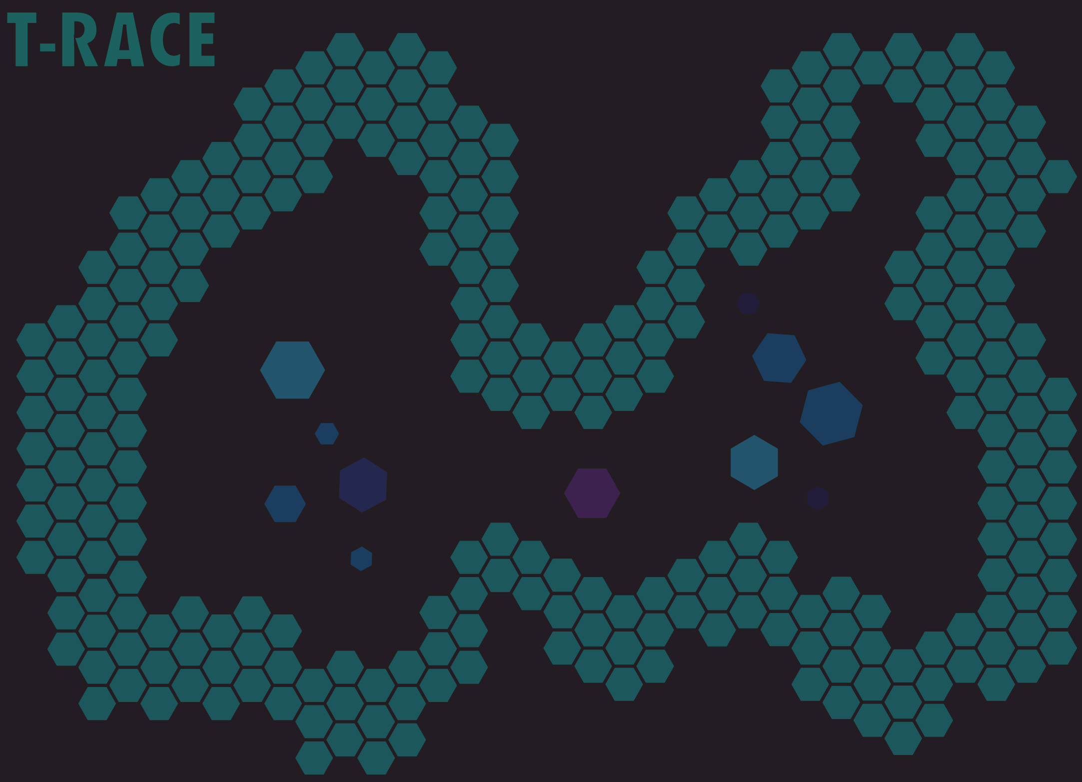


Figure 12. *Paper prototype used for playtesting "Derivate Racer", here called "T-Race". Not shown are field cards, special fields and track fields. These are placed on the board prior to the game starting.*

Testing

Derivate Racer was tested as a multiplayer boardgame following the rules listed on the previous page. The intent of the playtest was to discover if the game had sufficient promise to move ahead to digital prototying, and if it matched the design goals set for this project. Also, the playtest was to reveal any critical issues and potential digital features.

NOTES

Overall the game scored well on playability, and was judged to be interesting by the testers. The core game atom functions as intended, but offers limited room for growth on its own.

Track Fields did not all work as intended. The effects were either too powerful or too insignificant. They give deeper strategies to the game, but need more consideration and balancing. One worry is that they are hard to tie directly to the underlying math.

Multiplayer did in part overpower the focus on the game mechanics. The playgroup was especially interested in sabotaging and interfering with each others plans, and it was remarked that the crash penalty and "Field Cards" made strategic planning less viable.

Replayability was low when the track was setup the same way as previously, or when playing multiple laps on the same track for one race.

Counting tiles was an issue. Players were reluctant to test and explore different strategies in managing power and instead wanted to count and check how the game would run its course.

DIGITAL POTENTIAL

The game worked well as a paper prototype, and no player remarked that they felt the game required "digital" assistance to function. For a digital prototype, the main advantages will come in the form of automated handling of power and speed counters, greater potential for unique tracks for each playthrough, greater potential for handling different track fields and field cards.

CONCLUSION

This game was not chosen to proceed to a digital prototype. While the impressions were favorable, the most enjoyable components of the game did not align to the underlying game mechanics, but instead to the competitive aspect and between-players interaction. Also, the ideas at this stage that tries to add more depth to the game removes the focus from the intended mathematical topics.

Concept II:

Math in the daily life covers a wide range of topics in the curriculum. It covers economy, rough estimations, rationales for decision making, interest and more. The concept "Collapse" was the last concept to emerge and also the most refined concept of the three. Since it deals with the topic that is the least "theoretical" it also was the loosest concept in the sense of underlying mathematical theory.

Collapse tells the story of a collapsing cave filled with treasures where players try to get the most treasure for themselves before the cave collapses. Players have to conduct rapid decision making and optimise their strategies based on calculating the potential value of their actions. Players choose from a selected number of possible actions under time constraints without knowing what the other players will do.

CENTRAL ATOM

The game features simultaneous turn-based gameplay, and much like derivate racer the game follows a discrete rather than a continuous series of input-action loops. Again, by modeling this in a discrete fashion, one allows the players to reflect during gameplay, and more easily form and test strategies on the fly.

The second core of gameplay deals with optimising these input-action loops. The cave has treasures of different tiers of value, but each treasure has an area of use that can in itself be more valuable than the "gold" value of the card. This is but one area where the player has to judge overall value by comparing long and short term gains. Other areas appear when the player has to decide how much to chance when the cave is collapsing around him or her. Further from the safe-zones are the most valuable treasures, but also the more dangerous areas for collapses which punish the player. Also, players can instead of picking treasures from the cave opt to try and take them from their fellow players. All of these decisions have to be made under time pressure, and each turn attempts to be of sufficient low-impact that the player is tempted to play with differing strategies.

OTHER IDEAS

The game was tested as a multiplayer game, just like Derivate Racer. This adds depth to a game, but again draws the focus from the underlying mathematical concepts, which in the case of this game were already unclear.

This game has the least direct connection to specific parts of the middle-school curriculum, but math is integrated at every level of the gameplay, albeit in rather covered forms.

The game's least mature ideas are related to how to mange the collapse of the cave, which has to be carefully balanced in terms of risk vs. reward. As is always the case with games where the immediate outcome is not obvious is to make decisions feel impactful. Also, when the game tries to balance several decisions against each other, the risk is that dominant strategies emerge. When identified, such strategies kill enjoyment of the game, as mentioned in previous theory.

Figure 12. *Risk vs. Reward is the central conflict in Collapse. Players constantly have to consider the long term benefit of their choices under pressure. The challenge with such gameplay is to avoid dominant strategies and "bottom-feeding", where skilled players are rewarded for taking the "easy choice".*

Collapse

PROTOTYPE GAME RULES:
Name: Collapse
Tags: Multiplayer, rough estimations, game theory, basic summation.

Each player takes a player card, indicating the player's carried treasures. Only in the starting tile can players move carried treasures over to their "safe pile" that counts toward their total gold at the end of the game. Each player can only carry a maximum of 3 treasures at any time.

The game board takes place in a collapsing cave, and the game ends when the cave is fully collapsed or empty of treasures. The winner is the player with the most stored gold.

For each turn, all players have to choose 3 actions, in order, and place them face down. Each of the actions can be one of the seven following actions:

- Move (from a tile to another)
- Take treasure (from the current tile)
- Examine treasure (from tile or carried treasures)
- Steal or fight (alternates depending on the opposing player' actions)
- Unlock door (locked doors prevent movement between certain tiles)
- Dump treasures (put any number of carried treasures down into the safe zone or the current tile)
- Use treasure (use a treasure to get its effect)

Players are limited to 15 seconds to decide on their 3 actions, otherwise their actions are forfeit. Then everyone reveals their first action, and carries it out, then the second and then the third. Different actions have different priorities, and they go as follows: Use treasure, steal/fight, take treasure, examine treasure, walk, unlock door, dump treasures. Note that treasures are on a per card basis, and some treasures can be used at any time.

Under every treasure card is a hidden number, and every tile has a "collapse" counter on it. When a treasure is taken, the number underneath is added to the global collapse counter. If this counter matches or exceeds the collapse counter for a specific tile, that room collapses, forcing everyone in it to lose all treasure and start back at the safe zone. The tile is also blocked from being moved through, which can lead players to be trapped. For any trapped players, they also lose all treasures and have to start back at the safe zone.

Note that treasures that haven't been identified can still be used, but the player has to use them regardless of the effect they prove to have.

Fighting occurs when two players on the same tile decides to use the "steal" action simultaneously. Unless modified by treasures, the player with the least treasure wins the fight and gets to a treasure from the loser (loser's choice). Either player can give up their subsequent actions for the turn for an advantage. Example: Jon and Lisa both try to steal. Jon has two treasures and Lisa has one. This is the first of the three actions of the round. Lisa will win the fight unless Jon forgoes the last of his three actions, but Lisa can counter this by doing the same herself. In the case of a tie, both players have to skip the following turn. Otherwise, when only one player "steals" he or she can take a random treasure from the victim.

The game is played on a board where each tile represents a "room", and all treasures are distributed by the value levels indicated on the cards and rooms. Treasures are placed face-down and can not be turned face-up unless a player identifies or otherwise uses the treasure.

Figure 13. *Image of the rough paper prototype used for Collapse. Not shown are the different treasure and action cards.*

Testing

Collapse was tested using the rules listed on the previous page. As with Derivate Racer, the intent of the playtest was to discover if it had sufficient promise to move ahead to digital prototyping, and if it matched the design goals set for this project. Critical issues and the potential for digital features were also focused on. A summary of the playtesting notes are given here.

NOTES

Overall this game played the most like a finished game, with the exception of corner cases and balancing (of treasure values vs. effects primarily). Also, certain features such as stealing were not balanced enough, leading to the discovery of some dominant strategies.

Even more than in the case of Derivate Racer was the overpowering aspect of between-player interaction. The diplomacy and knifing between players strongly overpowered any focus on the underlying math. However here, at least the between-player interactions were more shaped by and aligned to the overall goal of the game.

Replayability was high, in that each game had a great deal of variation from how players decided to go about with their strategy. Players appeared to follow a specific strategy for each "run" at the treasure. Referring to the cycle of the player leaving the safe zone, collecting treasure and then returning to the safe zone to dump it.

When asked, players were not able to identify that the game had any agenda or underlying material it tried to teach. Instead they spoke of similarities to other board games leaning on the same type of decision making (The popular board-game Munchkin being one such example).

DIGITAL POTENTIAL

This is where the "finished" feel to the game worked against it. Playtesters said that they preferred the idea of the game concept as a boardgame over a digital alternative, because the digital alternative would reduce the more fun aspects such as cooperating and betraying one's fellow players. The key thing to explore in a digital version would be the implementation of shorter cycles, making the game run closer to real-time, which can't be emulated in paper prototypes.

CONCLUSION

This game was not chosen to proceed to a digital prototype. While probably a potentially fun and enjoyable boardgame, the concept is too removed from the idea of teaching mathematical concepts. The second main strike against it is the lack of inspiration for a digital solution. While several ideas, such as a greater variation of board layout, incorporation of AI entities, treasure functionality was brought up, none felt essential to the game idea.

Concept III:

Probability is a basic concept to grasp. Representing the chance of an event or set of events happening. In practice however, probability can be very hard to calculate and often counter-intuitive as can be seen by the popularity of the “Monty Hall Problem” and “Boy or Girl Paradox” that showcase these issues. Most games feature chance in one form or another, usually referred to as RNG (which stands for random number generation). This is one of the primary ways of ensuring a varied gameplay and not have the game follow a deterministic model. Very few games feature no randomness, for good reason.

Dreamweave was the most difficult concept to make, requiring several iterations on the paper prototype to reach even a semi-playable version. While probability is common in games, designing mechanics on probability theory proved harder than anticipated. In Dreamweave, the players play as spiders, spinning webs to capture dreams. The game’s mechanics are based on probability and combinatorics. The complexity of gameplay has the capacity for exponential growth as the game progresses, and rewards insight and understanding of the underlying topics.

CENTRAL ATOM

Again, the game is designed around discrete “turns”. While all three concepts do this intentionally, it is also due to the limitations of prototyping in paper. However, I do believe the advantages of this approach is justified by the theory, since it does lead to more reflection due to the immediate feedback it can provide. There are of course degrees to this approach, where “DragonBox” separates each turn as a full level, while for the three concepts I’ve explored, the turns all happen within a full game. Dreamweave however, goes one step further than the other two, and splits the game into two phases, each with its own subset of input-action loops. This game is also very much linked to mathematical theory, more so than the other two concepts.

Each of the two phases have its own central atom. For the “Day” stage, the players build their webs, where the atom revolves around selecting nodes to form webs. This means managing resources, and has consequences if done poorly. However, due to the cyclical nature of the two phases, new strategies can be tested and tried during gameplay without too severe consequences for failure.

In the night, the central atom revolves around selecting the “webs” to match an arbitrary task. This is an adapted but recognizable variant of the “urn problem”, which is central to combinatorics at a middle grade and first year high school level (represented by different coloured marbles in an urn, drawn at random and then either put back or not).

A second area of interest is the separation of “points” as means to winning the game, and “power” as the means to create webs. One is chance based, and the other can be awarded to the players on a more consistent basis, which creates a desireable dynamic where experienced players have an advantage, but new players can still experience moments of success.

OTHER IDEAS

The game changed immensely from the first variation on the idea. Originally the game centered more around resource management and the probability mechanic was secondary. This is shifted in the prototype version, where the game follows a structure closer to a level-based puzzle game with its use of the day / night cycle. The game was also tested as a multiplayer game, but nonetheless functions as a single-player variant without further alterations.

Dreamweave

PROTOTYPE GAME RULES:
Name: Dreamweave
Tags: Singleplayer, multiplayer, probability, combinatorics, turn-based, puzzles.

1. Shuffle the "task" cards and put them facedown. Roll a dice to decide the starting player and go clockwise from there.

2. Each player draws a "web" by linking three nodes, repeat this until each player has three webs.

3. Flip the top "task" card, and the game beings. The game cycles between day and night cycles until the stack of task cards is emptied.

Each task consists of one or more elements, where each element can be one of the four types (squares, triangles, stars and circles).

Night:
1. Each player has to decide which of their webs they want try matching the task with. For each web they intend to use, the player puts a counter indicating how many draws he or she intends to use the web for.

Note: A web can only be used for X draws per task depending on the web size. X is 1 if the web has 2 or less active nodes, otherwise X is the number of active nodes minus 2.

2. When each player has decided how they want to use their webs, check for each player if they succeed. For each draw, the web yields an element. The type of element is decided randomly based on the active nodes in the web. Example: A web consists of two triangles and a circle, this web has 2/3 chance of yielding a triangle and 1/3 chance of yielding a circle. The node of the yielded type is then made "inactive" and no longer count as a part of the web for the rest of the night phase.

3. If a player succeeds, he or she gains points according to the point chart (1,3,5,8,12 points depending on task size). If all players "pass" on the task, the game reverts to the day phase. Each player that attempted to solve the task, gets 1 power if the task size is less than three, and two power otherwise.

4. Draw a new task, and repeat the cycle until everyone passes on a task. The game then reverts to the Day phase.

Day:
1. Each player is awarded 3 power and each web is restored (all nodes become active again).

2. Players can then build additional webs, or expand their existing webs at the cost of 1 power per node. Each web has to be a minimum of three nodes.

3. When everyone has spent the desired amount of power the night phase starts again.

Power can also be used during gameplay to re-roll the generation of an element, change the task, look at the future tasks or double the points received for a task. The specific costs are listed on the player cards.

Figure 14. *Example of the aesthetic and narrative that informed the prototype version of "Dreamweave" where players play "magical" spiders.*

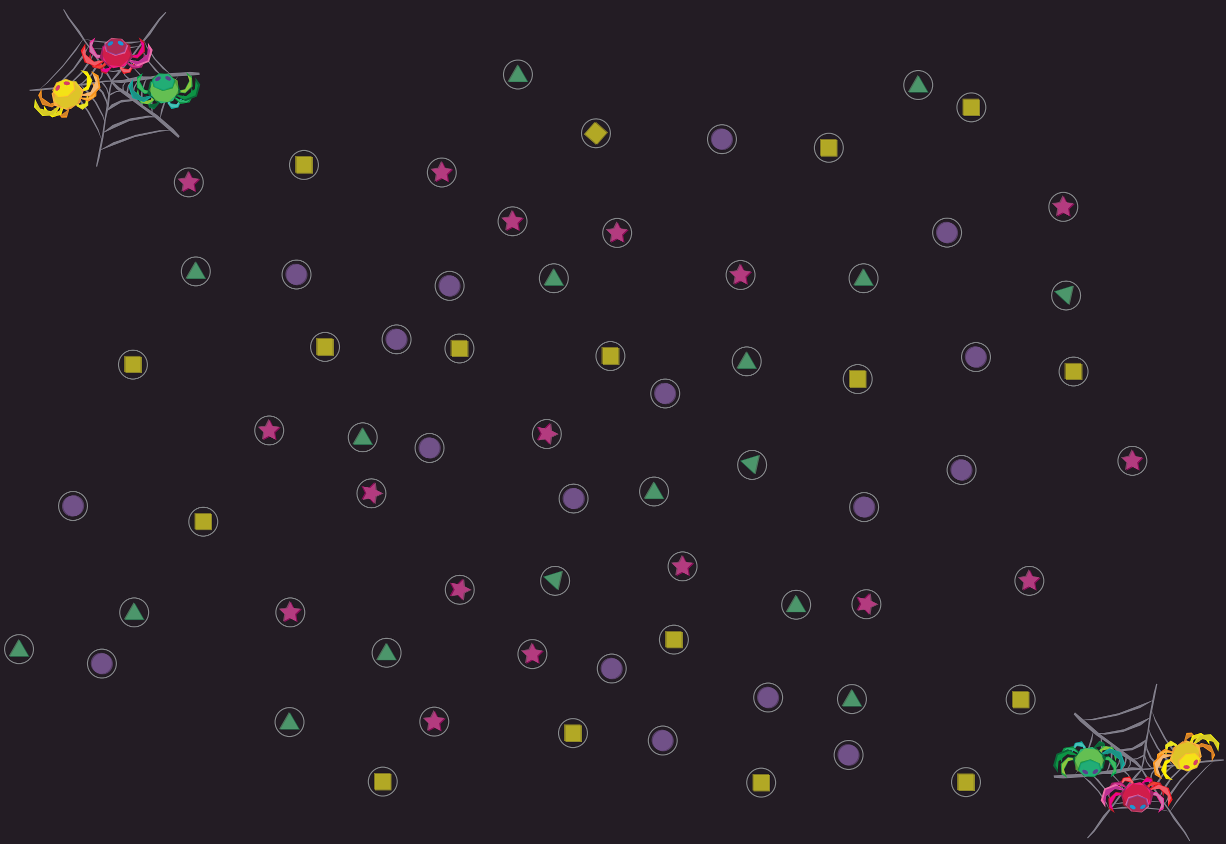


Figure 15. *Image of the rough paper prototype used for Dreamweave. Players drew their "webs" on with pen, and used coins to mark which symbols were exhausted.*

Testing

Despite three variations before even resulting in a somewhat playable concept, even the final version, adhering to the rules listed on the previous page, proved hard to test. Testing for "successes" and "failures" took a long time, and made the game run longer than intended. However, the testing gave a lot of insight into the game, and what worked and what didn't.

NOTES

Both core game atoms, for the day and night cycles respectively, were enjoyable and held promise. The night cycles took a long time to complete, because every random generation required rolling of several dice and interpreting a rather cluttered board. These atoms proved to have sufficient depth, and no obvious dominant strategies. If anything, the impact of the players' choices sometimes carried too little impact on the outcome. The balance between chance and skill has to be balanced for the players to feel they are rewarded based on the mastery they achieve and not just due to fluctuations in chance.

The game was tested with both single and multiple players, and while the multiplayer experience proved more enjoyable overall this was due to the added layer of social interaction. Within the context of the game, the interactions were limited, and players focused mostly on their own decisions, only cheering other's misfortune (or extraordinary luck).

The additional abilities had some issues and corner cases in regards to which player had priority and when some actions could be taken. These needed to be resolved on a case-by-case basis, and detracted from the gameplay when the player's found "loopholes" that gave unintended advantage.

The players easily identified the underlying principles of the game to be centered around probability, but did not specify that they felt the game had an agenda. When asked, they identified the game as a puzzle or strategy game.

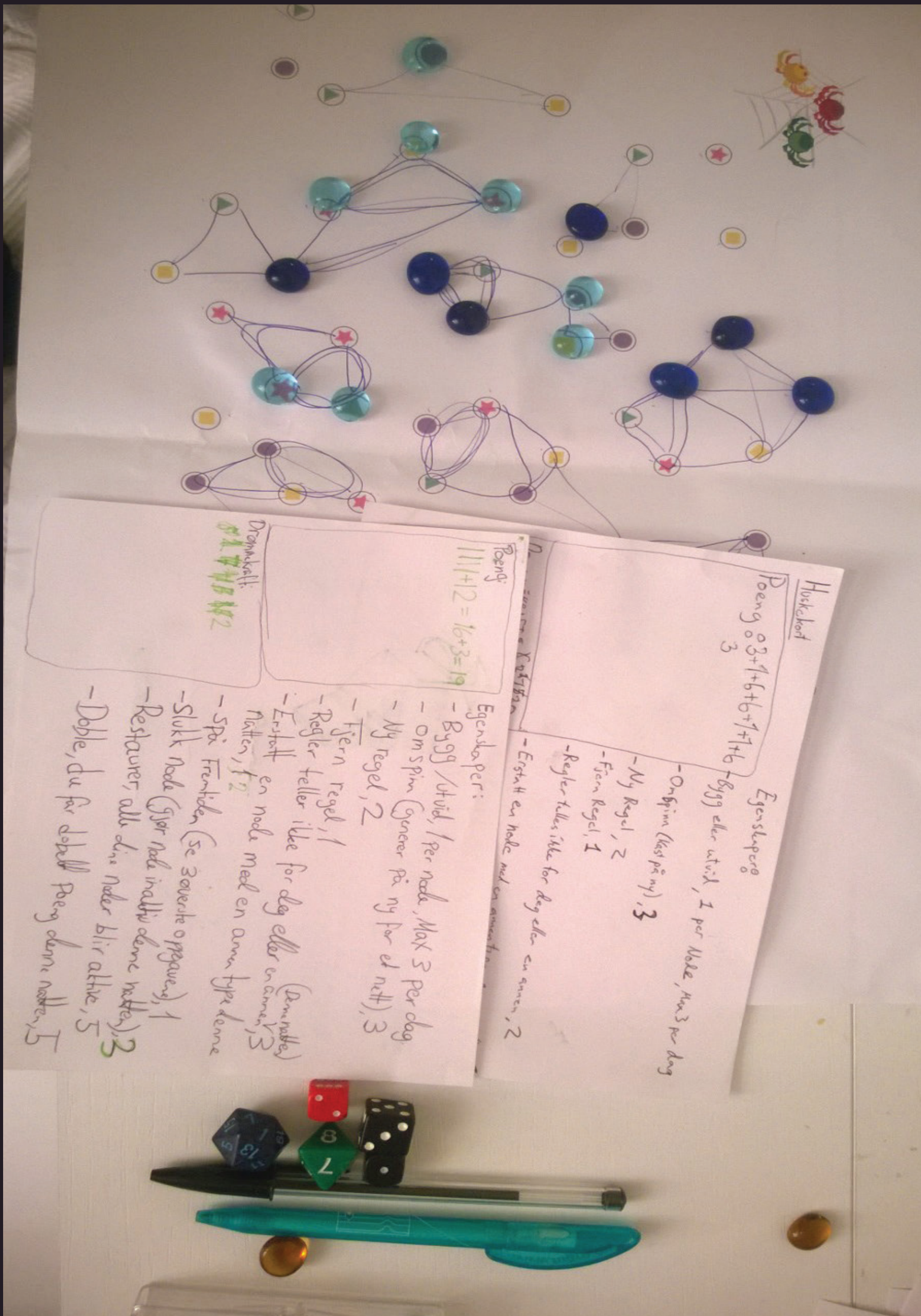
DIGITAL POTENTIAL

Unique to this concept was that most players agreed the game would do better as a digital game. Mostly because of the time it took to calculate the success / failure for each task, but also because the ideas they wanted to implement required such technology to be feasible. Ideas such as continuous gameplay (removing the turns), rewarding players based on their chance of success, or rules that changed how the webs functioned from night to night (such as certain compositions acting differently, or some symbols being "worth" more et cetera.) are all examples to consider.

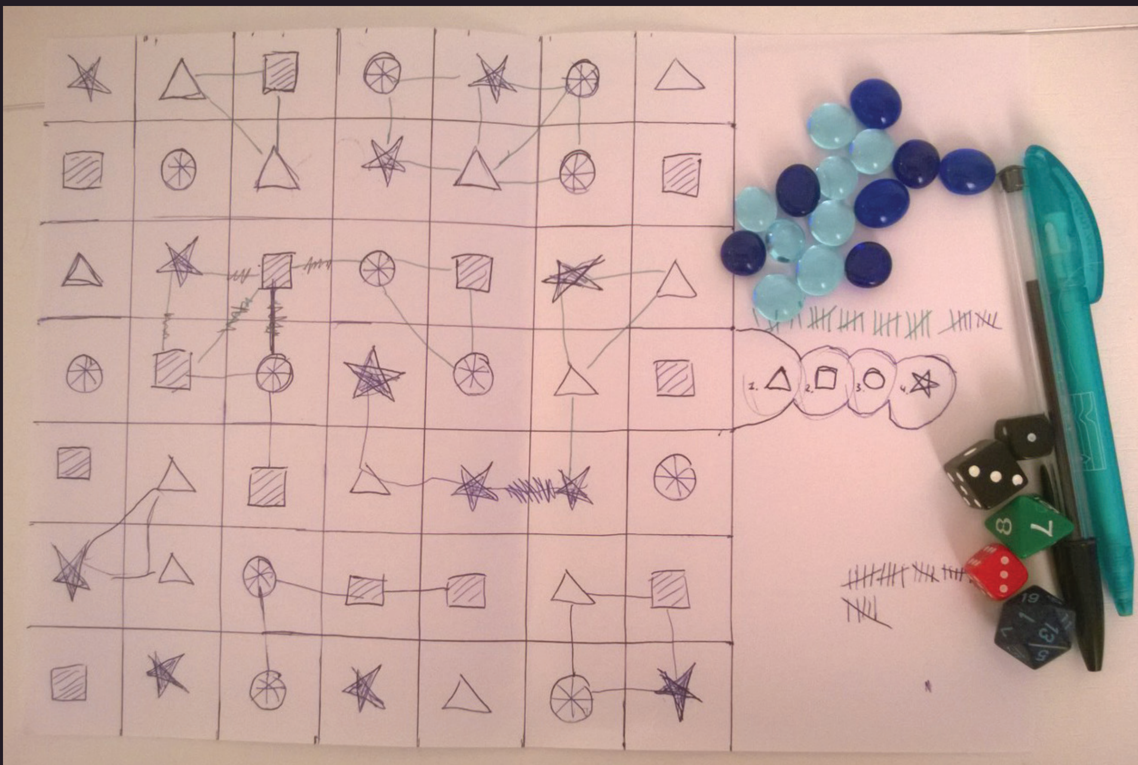
CONCLUSION

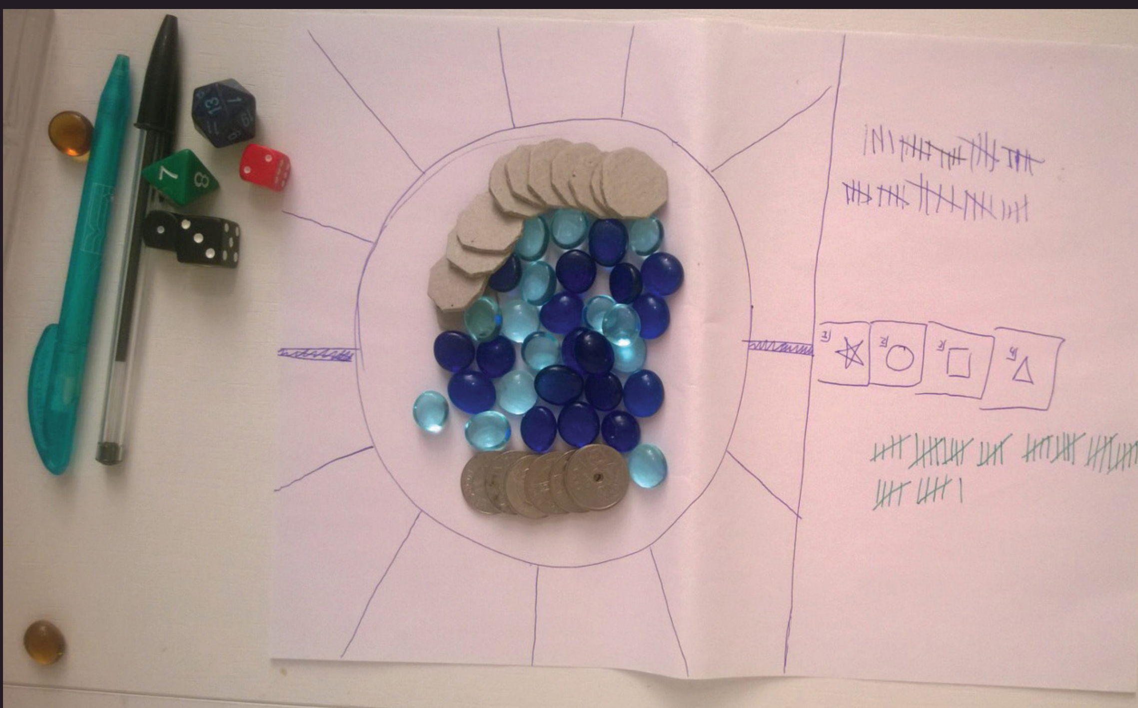
This was the selected concept, and was to be made into a digital prototype. The alignment and clear integration of gameplay and mathematical principles is the strongest for this concept, and this is the main reason for the choice. Secondary was the clear "benefit" that the game could receive by moving to a digital format. Being able to move a lot of the probability checking to occur "under cover" so to speak, and also speeding up this process would have been a great asset to the paper prototype and is but one of the features that a digital version can provide.

Dreamweave:



Paper Prototypes





Moving Forward

In the appendix you can find the general manual used for each of the paper prototype playtesting sessions. In addition to having their gameplay observed, each player was asked to fill out a form rating the game on different axes such as overall enjoyment, how interesting they found the concept, and if they would prefer the game in a digital or analogue format are some examples. The second part of the form had some open ended questions, where the players could fill in as they pleased.

I believe this somewhat informal way of playtesting yielded the best results given the state of the paper prototypes. By following less rigorous constraints, players were more encouraged and willing to talk and discuss the game both during and after game-play, resulting in several interesting areas for future consideration.

The paper prototypes tested at this stage did not represent the last playtests featuring paper prototypes. As the concept "Dreamweave" was brought further, single features as well as full play-throughs were playtested using the same method as described here before being implemented into the digital prototype. This was of particular importance because the implementation of each feature in digital form represented a large investment of time and effort, and by at least playtesting some form of the feature prior to implementation saved more time than it took.

CHOICE OF CONCEPT

I believe each concept could result in an interesting game that would have satisfied the goals of the project. However, Dreamweave provded to be the concept most in-line with it's close integration of probability theory and gameplay mechanics, and a natural balance of input-action-feedback for each central atom. It also didn't rely on multiplayer, which I knew would represent a major hurdle in digital development, and as a concept had, on the surface, the most to gain from moving to the digital format.

EDUCATION AND GAMEPLAY IS ALLIGNED

GAMEPLAY PROVIDES ROOM FOR REFLECTION

SATISFIES THE GAME ATOM "CRITERIA"

ADJUSTABLE COMPLEXITY AND PROBLEM-CENTERED MECHANICS

OPEN TO PLAYER EXPLORATION. ENGAGEMENT ON UPPER LEVELS OF BLOOM'S TAXONOMY

Figure 16. *Some of the theoretical princples for game-based learning, and "fun" that Dreamweave satisfies to some extent. Retaining, or better, improving, these qualities in the game will be crucial for future development.*

“The most disastrous thing that you can ever learn is

your first programming language.”

- Alan Kay, computer scientist

III: GAME DEV

Starting Point

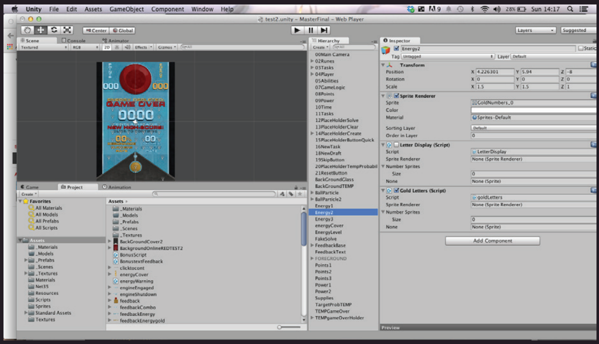
Despite the choice of concept being made, several playtest sessions and design iterations were required before game development in Unity could begin. But, since the choice of concept was made and the core game atoms were deemed adequate, the design process could be more focused. All design decisions could be judged and undertaken more quickly because of the existing game framework and a basic familiarity with what the game was about and what I thought it could become.

In parallel to these iterations, the implementation of core functionality, such as selectable objects and creating groups of objects, was begun in Unity both as a way to save time and to learn more about the workflow. This way of working in parallel also helped inform me of whether or not an addition or change to the gameplay was feasible within the timeframe for the project.

These next pages showcase a shortened version of the design process that took Dreamweave from it's first paper prototype to the final playable tablet-game. It also explores why certain features were added, changed or cut altogether, and where bottlenecks occurred in the development process.

WORKING IN UNITY

Working in Unity takes place in two different workspaces, that of the Unity engine and that of MonoDevelop, the accompanying IDE. In the first you modify and position game objects, which can be represented by anything from 2D images, 3D models, particles or even invisible game objects. In the inspector, you can assign properties to the objects and manage the hierarchy of objects in your game. The engine also keeps track of all assets used for the game, such as image files, animations, scripts et cetera. The MonoDevelop IDE is where you write code for the scripts attached to the objects in your game. It is through these scripts you ensure that the objects behave in accordance to the game mechanics, and that the whole sequences of player input - action - feedback that the game revolves around are handled properly.



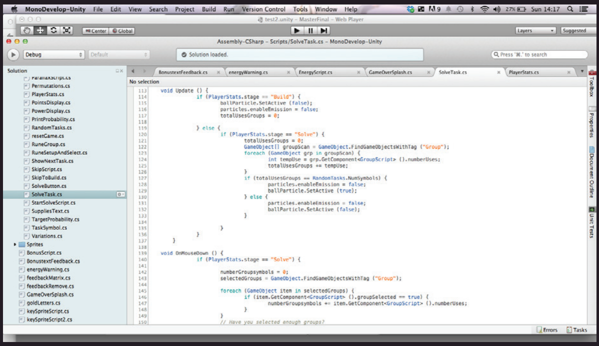


Figure 17. *Top screencapture is from the Unity work-space where game objects are positioned and linked, bottom is from the Unity scripting IDE where the coding is done.*

Working in Unity is not dissimilar to working with other software tools such as CAD programs (such as SolidWorks) or graphic software (Photoshop, InDesign and similar). Familiarity with such tools made it easy to familiarise myself with the ins and out of the Unity engine.

The scripting environment, shown in the bottom image of figure 17. was something else entirely. Here, all work is done in a scripting language (in this project C#) where syntax and "correct" practices are of vital importance to a whole different extent, otherwise the game will not run at all. Learning the language of code, and being able to express game functionality and rules through "code" represented the bulk of the work for this project.

PAPER ITERATIONS

The images shown in the previous chapter are some of the paper prototypes used to test “Dreamweave” prior to and in parallel with the development process. Listed are some of the main changes done to the game as a course of this testing, including cut and added features:

Free-form webs vs. grid structure - By changing the way to visualise the “webs”, from free-form structures to a square grid where each tile were given its own element, represented a major aesthetic change. This was done during playtesting as a way to improve the overview of groups and “used” elements during gameplay. The top image of figure 18., compared to figure 15. illustrates how this made the gameplay more intuitive.

- Abstracting the game aesthetic. In addition to the grid based structure, all references to the “spider” narrative and aesthetic were eliminated. I was unhappy with the chosen game aesthetic, as playtesters did not see a connection between it and the gameplay. As a temporary solution, the aesthetic was abstracted. This helped with development, since mechanics no longer relied on a special theme to communicate function. The intent was to re-create a more suitable narrative and aesthetic later in the process when gameplay functionality was more or less fully implemented.

- Eliminating multiplayer. Despite several of the paper protoypes being tested in a multiplayer setting, the idea of multiplayer was quickly cut in development. The immediate benefits came in the form of simplifying the GUI and player controls and reducing technical difficulties linked to handling player-to-player interaction in realtime during gameplay. However, while players mostly focused on themselves during gameplay, the mere presence of multiple players in a setting where they didn’t explicitly cooperate, resulted in the players engaging less with the game mechanics. This felt counterproductive to the game’s secondary goal, and is one instance where this goal was prioritised over the first (albeit hopefully at a minor cost to it). Further, I believe the change is justified by how games are played on mobile platforms since I would not be able to feature online multiplayer, but only single-device multiplayer functionality.

- Introduction of the “probability check”. This was always one of the considered features for a digital version, and was quickly added on the development to-do list. In taking advantage of the technology by checking the actual probabilities for success allows rewarding “good” play and strategies explicitly, without just them being represented with an advantage over time (by having better chance of success than less “good” decisions). It also adds another level of user feedback, that allows the players to explore just how their ideas work out.

- Time-based gameplay. Without a stack of “task” cards to represent the game length, and other players to encourage “speedy” decisions, I needed a natural way to end the game and to add pressure to the decision-making. Without the pressure to make decisions, players were free to take focus of the game mechanics, reducing player immersion. The addition of a time counter, that counts down during the “Night” phase, was my solution to this problem, and added the secondary goal of keeping the game “running” by having solved tasks give a time bonus.

AESTHETICS

Shown over the next pages are also some of the aesthetic directions that the were considered through the course of the project. Not all of them were given much consideration, but it reflects how the aesthetic and narrative elements of the game changed in order to fit them to the game mechanics so that they could all support each other in accordance with Schnell’s four elements of gameplay.

Several of these aesthetic changes were also the reasons for accompanying functional changes such as the placement of GUI buttons and gameplay gestures. A good example of how all the gameplay elements can be used to inform the others.

DevLog I

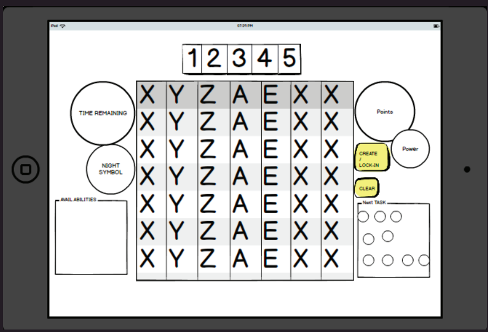


Figure 18. *Early conception of the game interface, showcases the "grid" structure over the free-form webs of the first prototype. Made in Balsamiq*

Figure 19. *First game objects. These textures represent one of the four element types and shows how the sprite for the unselected and selected version of the element "Zetta"*

Just like having the core game atoms of a game be fun and interesting, it is of immense importance that the basic player inputs work flawlessly. Going by personal observations and experiences, most people who have used smartphones, tablets or just about any type of computer software, have experienced the annoyance that comes with an interaction being just a bit "off". A slight delay in the response, a hitbox being mismatched to the visual object, or a function that doesn't communicate it's intent clearly, or anything that doesn't match up to the host of "unwritten" rules about just how such and such features should work, are all examples of such occurences.

So creating these elementary interactions was the first order of business for developing the game. Not only are they essential for gameplay, so they would have to be made in some form or another before the game could be testable, but by making them as "finished" as possible immediately, they could also be subject to continuous testing for as long as possible. This way they would stand to get the most tuning out of all the game's features.

"BUTTONS"

Drilling down to the fundamental moves in the game, the game atoms of the game "Dreamweave" (which wasn't referenced as such during development, but is so here for sake of consistency) are basic GUI actions. By having it so that these actions are nothing more than pressing buttons instead of making use of more complex gestures, the game avoids distracting from the gameplay that matters. Buttons are also very integral inputs for touchscreens, and are familiar to all users of such devices.

Unity supports creating buttons rather easily, and allows for calling events in scripts using the OnMouseDown function. This means that when the game registers a player touching on the"collision box" of a button, it runs a the attached sequence of code once. More advanced methods, where you require the button being pressed and then released over the same area makes it easy to have any "button" element animate according to the input, giving appropriate feedback to the user. Figure 19. shows one of the four types of elements that make up the grid structure in the two states of selected and unselected. Each of the elements in the grid is a button on it's own, and when the user creates groups of these elements, all the grouped elements form a new button over these.

Getting the many buttons of the game up and running was the first stage of the game development, and once it was accomplished a rudimentary game could be played without using a paper version of the grid (but still simulating other mechanics and randomness with dice and such).

DevLog II

The second layer of features to be implemented were that of a random “Task” generator, and allowing for groups of selected “Elements” to be created. The latter represents what the first iteration referred to as creating “webs”. These two features were also the first major hurdles in the development process, requiring a more extensive “logic” than that of state-changing or one-shot effect buttons.

With the implementation of the grid structure it also followed that groups had to be made of adjacent elements, so that game visual display could be read at a glance. Visually separating the groups from each other was then required, and was a major issue in the earliest playable versions of the game. For generating tasks, one immediate problem was that the game couldn’t have the process be completely random. With a stack of cards representing the tasks, the game could “stack the deck” by having the easier tasks come with high frequency at the start and then slowly progress to more extensive tasks (with more elements in them). A similar feature of semi-randomness was required for the digital game.

DYNAMIC LOGIC

By naming each element in the grid according to its placement (in terms of row and column number), the game could check if the element the player was clicking was adjacent to any of the previously selected elements. This formed the basis of the logic for selecting elements for groups.

However, to ensure that players couldn’t unselect certain elements and then form a group with non-adjacent elements (the same group forming one or more islands in the grid instead of one connected shape) the functionality to “unselect” elements had to be removed. Instead of simply tapping already selected runes, which in playtests was the most commonly used method, a button with a “clear all selections” functionality was the only way to unselect elements. A clear case where the my ability to code caused a compromise in the desired gameplay.

The earliest versions also used simple block shapes of colors behind the elements to show which group they belonged to. But by checking each element in the group for it’s position and adjacencies, a dynamic system for drawing the group “backgrounds” was implemented. This better separated the groups from one another, relying on the visual form over simple colouring to separate them. This use of form (and also often colour) to separate game objects ensures that gameplay is available to everyone, whereas colour alone is insufficient in the many cases of colour blindness or deficiencies.

Through this dynamic logic, each element is assigned one of 15 possible backgrounds depending on its position and adjacency when forming a group.

SMART LOGIC

The envisioned game and the direction of the game design, made it so that the game was very much divided into discrete steps and required few continuous updates. However, sometimes the game had to update itself when certain conditions were met; by hiding buttons during gameplay that were of no use, the GUI could keep from being overly complicated at any time, with only useful information and input-options being visible at any time. Sometimes the game also needed to transition automatically, without input from the user.

This required “smart” logic, that constantly checked the state of the required transitions. In the early stages of game development, these were kept to a minimum, but from playtesting it was always found that players did not like to make an effort to progress the game themselves. Doing so, such as clicking buttons to proceed, felt unrelated and distracting to the gameplay. Creating the “tasks” the gameplay centered around was one such feature. In the end incarnation, no input is required to generate such tasks, and the player is automatically given a new task after solving or attempting to solve the one that came before.

DevLog III

The third and last layer of functionality to be implemented took the longest time to get working. The first and second layer represented the essential gameplay, with which the game could be tested with full "playthroughs" and replace the paper prototypes for this purpose. The third layer of functionality represents the gameplay features that could only work in a digital version, and came at a stage in the development where testing had revealed several critical issues with the gameplay as it were. Explored here are the many changes to the core gameplay that were made and implemented at this stage, and the problems encountered with their implementation.

FINDING THE CHANCE

Despite the goal and intent to make the game check the probability of the player's selection succeeding against the current task, this proved more complex than initially assumed.

The basic mathematical theory for a specific scenario where the task size, task content, the content of each group, and the number of "draws" per group is known does not exceed the curriculum by much. It touches on something known as hypergeometric distributions, which is first introduced at a high-school level, when the user draws multiple elements from any single group, but beyond that, the tasks were solvable with pen, paper and time for the most part. However, to have the game check and calculate this for every possible scenario proved a major hurdle and was only implemented at a late stage in the process. The problem was one of recursion. To be "smart" the logic had to act differently for each scenario, and this proved too much for me to handle. In search of an answer to the problems, I asked around on the mathematical and programming Stackexchange websites dedicated to answer problems of this kind. The questions and the answers I received can be found in the sources under Stackexchange.

The final solution was one of using brute computer power to calculate every potential outcome based on the users selection, and compare these against the task. This involved using mathematical libraries and plug-ins to allow for calls on these types of functions (such as get combinations, and calculating binomial coefficients).

This cluttered process is hidden from the user, and since it did not result in a performance issue remained in this incarnation even in the final version. The end result is a number representing the users chance of success, based on his or her selected groups and the current task. This feature then became the basis of a secondary objective in the gameplay and it is the most important change to the core gameplay mechanics that occurred during the development process.

TRACKS OF OBJECTIVES

With the aforementioned feature, the "solve" phase (previously called "night") underwent a major change. The primary task remained unchanged, this is where the players try to generate a selection that is equal to that of a task, however, in the early stage of the "Solve" phase the player rarely has the opportunity to guarantee success. The final outcome is left to chance, and even when the player made the best decision they could they would be left without anything to show for it should chance be against them. This was undesirable both from a "fun" perspective, but also from the secondary goal of teaching the player the underlying theory. I wanted to reward the player for understanding his or her actions, while still making the game playable for those that simply cared little and less about their decisions.

By adding a second "task", in the form of a specific probability outcome (10%, 33%, or 50% for example) the game could introduces a mechanic that is less chance based and clearly encourages an understanding of the underlying mechanics. No change on part of the user's input was made, but now their selection was not only checked against whether or not it matched the original task of elements, but the probability of success was matched against the new "probability target" task. If the chance of success falls within a certain

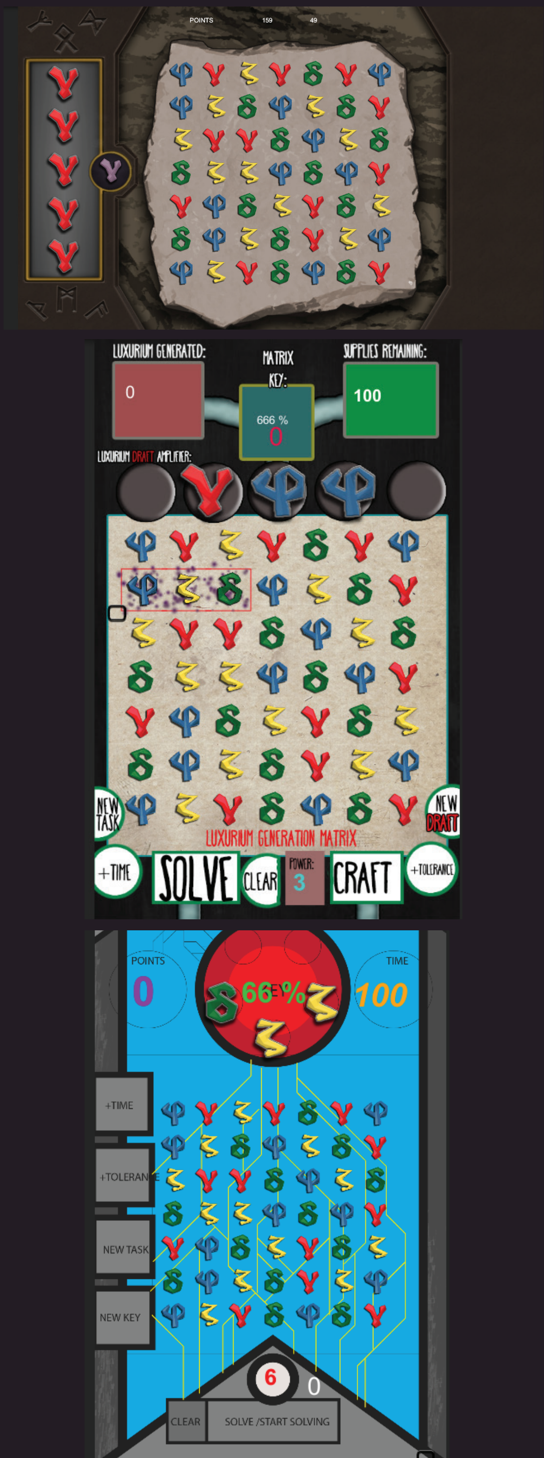


Figure 20. *Iterations of the aesthetic and functionality in the game. From the first versions down to a more recent.*

threshold of this target, this represents a success for this secondary task.

While adding a lot of complexity, and in the first playtests proving a less-than-intuitive feature, the added feature gave additional strategic depth to the game and an additional reason for the player to engage with the mathematical topics of probability and combinatorics.

GROUP AND USE CHANGES

Once of the major areas of playtesting, was to check if the player's decisions had sufficient impact on the chance based outcome of the primary task. One of the early findings was that with a group size of minimum three elements, and the maximum number draws for each group only being one for groups of this size (increasing by one for each increase in size beyond three), left the player with vanishingly small chances of success all too often. Also, the introduction of the second objective meant that players needed a greater ability to influence the outcome. By playtesting alternatives, the end result was that users could create groups of two or more elements, and the maximum number of draws was decided by group size minus one (and one for each group with only one active element remaining). This functional change was not difficult to implement, but resulted in much smoother gameplay and less frustration from constant "failures".

TIME TO ENERGY

Described earlier, the game implemented a time-based system to end the game naturally and to keep the player focused on the game during play. In playtesting, this felt needlessly punishing on players that simply wanted to think through their decisions, and it being a single player game made them want to decide their own pace. To fix this, time was converted to another measure of "energy", that ticked down for each start and end of the "solve" phase of the game. The ability to prolong gameplay by good decisions did however remain unchanged.

DevLog IV

The devil is in the details; this is certainly a true tenet for game design and development. The last stage of development and testing emphasised the smaller details of the game mechanics, and it mostly involved cutting back on features that had been added with too little thought, and making alterations to smoothen the gameplay. Also, this was the stage where most of the aesthetic work was done, and a lot of effort was put into making sure that the aesthetic and narrative supported the game mechanics.

CREATING GROUPS

Over the course of development, the "build" phase, where the players created their groups did not feel impactful. Also, being locked into the choices made at the start of the game was a source of annoyance, especially when the player realized he or she had made a mistake with their creation. To solve both of these issues, this phase was revamped. For each cycle the player's groups were reset, which encouraged trying out different approaches within a single game.

REDUCING FEATURES

First on the chopping board were the "abilities" to re-roll and change the tasks at the cost of "Power" which was otherwise only used for creating groups. This led to the players facing the dilemma of creating groups, which was otherwise seen as fun since it prolonged the "solve" phase, not being the optimal decision. Instead, it became more attractive to save this currency all but guarantee a successful result that would recoup the energy and gain points. Due to this, the abilities were cut back to a single instance per solve phase of changing both the probability target and the original task. This helped smoothen gameplay, remove "Power" as a third currency too keep track off at all stages in the game (along with points and energy), and also simplified the GUI. All of which were desirable.

ON FEEDBACK

The game atom model stresses the need for balanced feedback. Highlighted by Koster is that disproportional feedback compared to the input and result, is a way of cheating the player in the long term. While the gratification on part of the player might occur, the gameplay in the long-term will feel stale and cheap if the feedback is disproportionately large, and hard and pointless if it is too small.

Much of the work went into finding out what type of feedback the users wanted, and what they required to engage with the game mechanics on a deeper level, eventually resulting in mastery. No one wants to read a full report for every minor input, but players wanted to know the result of their actions.

In the end variation, when the player confirms their choice and presses what is effectively a "This is my answer"-button, the game, in addition to calculating the outcome, also provides the user with the end result for both types of tasks and the rewards (based on this outcome) on a splash. This again gives more freedom to the player on how long they want to reflect on the outcome and their choices. It also led to some players trying guess their outcome prior to seeing it, further engaging with the mathematical theory.

ON ART

As can be seen in figure 20, the game underwent major changes over the course of development. Multiple narratives and aesthetics styles were considered at one point or another, from the first abstract versions, to something which invoked the idea of runes, shamanism and predicting the future (borrowing heavily on tropes from fantasy literature) back to something abstract and machine-like. The key functional change came from turning the format to portrait over landscape. Not only did this change create a better gameplay flow, where the user could see how the grid, the tasks and the buttons interacted with each other, but it is also more in the style of this kind of game. When asked, many playtesters commented that they expected a more flowing and dynamic type of game with the landscape format, and a more "thoughtful" puzzle game in the portrait format.

The final art style is a blend of futuristic and abstract elements. The word choices and the graphical style tries to capture the narrative of the player working with some sort of futuristic machine. In the "build" phase, the player embeds circuitry into the machine, and in the "solve" phase the engine turns on and the player has to solve tasks to keep it running. The currencies of "energy" and "points" are also color coded consistently in all stages of gameplay, further communicating various functionality to the player.

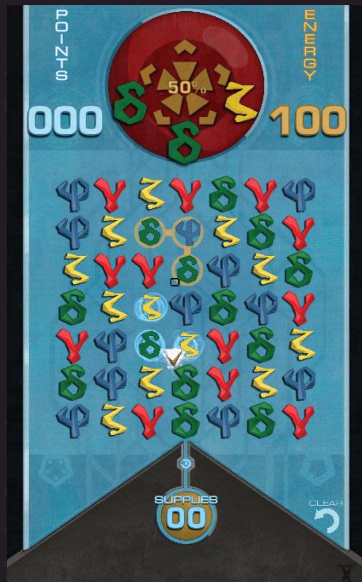


Figure 21. *Gameplay from one of the latest versions, from the "Build" stage of the game.*

ON TESTING

Playtests were done in parallel to game design and development. Most of the playtests with digital versions of the game were done using the Unity webplayer instead of on the intended end form of a tablet. This of course did lose some of the functionality in the translation, such as using a cursor instead of a finger to select and press buttons, but ensured a consistent experience that otherwise couldn't be guaranteed without me being in the same room as the tester. This ability to test at any time, with access to a wide range of playtesters, was of great importance and I believe that since this type of testing was supplemented with testing on an actual tablet that the end result didn't suffer because of the differences in media.

Figure 22. *Gameplay from one of the latest versions, from the "Solve" stage of the game.*

Testing using the webplayer does not imply that I wasn't able to observe the playtests. Through screen-sharing software (a functionality in Skype) I was able to observe the users actions without distracting. I could also prompt the users when something of interest was observed. Overall, the impression of players grew more positive for each iteration of the game. Which encouraged progress and continuous tweaking of scripts and variables.

It bears repeating that the notes on development listed here is just the highlights of a longer journey that required the better of three months from start to finish. If it even can be said to have finished, as the game still has its share of kinks and flaws (outlined on the following pages).

Issues & Fixes

As the development process drew to a close along with the project as a whole there were still issues in the game that remained. Outlined here are some issues that at the time of writing, have not been fully dealt with along with a proposed series of fixes that will be included in the final prototype as time allows.

Some of the issues mentioned here have no proposed fixes, and what this means for the game-play and why this is the case is reflected upon in the "Post Mortem" section of this report. Most of these deal with gameplay from a perspective of "fun" rather than evaluating the learning. More reflection on part of the learning aspect of the game is also explored in the "Post Mortem" section.

INTUITIVE GAMEPLAY

One problem, exacerbated by the choice of platform, is that the game mechanics are not immediately clear and intuitive. For mobile games this can be a strike against the game, at least if the player is not swiftly brought up to speed and feels confused. Players don't like feeling stupid, and are not always willing to read the "manual".

The final prototype suffers in particular due to the "Resonance" mechanic (previously referred to as the probability task), where the player has to match the probability with their selection with the predefined "Resonance" probability. The feedback splashes, shown after attempting to solve each task did teach users over time, but seldom before the user reached an undesirable level of frustration.

The "fix" for this is to implement a tutorial. Preferably, the tutorial would be an interactive version that is forced on the new players without being to obtrusive. This is the way most games deal with ramping up complexity; by stepwise introduction of new game mechanics. However, given the remaining timeframe, a simpler form of tutorial will have to be implemented and the first few series of tasks are to be made simple enough so that the user not feel overwhelmed.

UNSOLVABLE TASKS

The way of creating new tasks, of both kinds (referred to as "Matrix" and "Resonance" in the game), is done on a semi-random basis. Beyond a variable checking how many tasks the user has solved (which adjusts the number of symbols in the task), there is nothing else that is used to make the creation of new tasks "smart". As such, there are cases where the user has to make a selection with the chance of success being zero, and the game is not able to check for if or when this is the case. For most uses, scenarios in which this occurred was a source of frustration, especially the first time when playing. They would typically use their one-time ability to change either or both tasks, but if the scenario occurred again they were stomped at how to proceed. In the end they simply made a haphazard selection and pressed "Solve" as it became clickable.

This is one of the issues where no immediate fix is apparent. These events are not without strategic value to the gameplay, since players can use it to "prune" their groups to have better control over the results of subsequent tasks. One fix that will be tested is the ability for repeated uses of the "Change Task" functionality, at the cost of Energy. This results in a shortened gameplay if used too frequently, but keeps the user in control over what he or she finds the most annoying, trying to solve an impossible task or pay with "energy" to avoid it.

PLAYER CONTROLS

At the time of writing the script controlling the start of the game, where the user has to pull up a "curtain" in order to start, has problems with its implementation. The control requires the user to drag slowly or not be able to start. The problem resides in how the script handles the moving object, and a fix is simply to tweak the method used.

From an aesthetic consideration it is desirable for the movement to have a sense of inertia, since the "curtain" is supposed to represent a heavy lid that covers up the game "board" so to speak.

GAME LENGTH

In the last full-scale playtests the game went past what both I and the users considered optimal. It is always the case that a game is best served if it ends while the players are enjoying themselves, rather than to drag on past the point where they stop caring. If a game reaches this point, those that don't put the game away start making ill-advised random choices which nullifies any learning they might have acquired earlier in the gameplay.

Again, this is a fix that requires additional playtesting to ensure it isn't over- or underdone, but is in essence a tweaking of costs. Perhaps a solution that combines with a way to "skip" unsolvable tasks will sufficiently shorten the game to the point where it ends on a high note.

CONTROLLING THE DIFFICULTY

In its current incarnation, the game is too sporadic in its difficulty. This does not so carry so much impact on the fun of the gameplay, but still does affect it to some extent. Mostly it results in the learning part of the game being undermined, and is what I believe to be the primary weakness of the game from a learning perspective.

By more careful control over task generation, the difficulty can be gated, allowing the game to stay more consistently in the "thin zone", instead of skipping in and out of it without a set pattern. More consistency on part of the task generation also makes it so that the user feels they have more strategic control and the ability to make more impactful decisions.

The major challenge with a fix to this part of the game is that it would involve fixing a very integral script which is referenced by most other game elements. However, it can be done, and such a feature is high on the list of desired fixes.

IN CLOSING

The final version of the game that was subjected to more extensive playtesting was received rather well overall, despite the issues mentioned here.

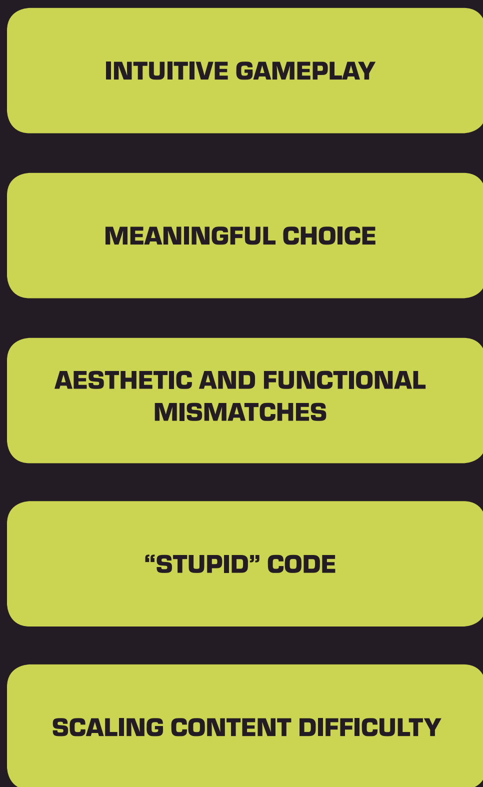


Figure 23. *The main categories of existing flaws. In the next section the game will be inspected in greater detail.*

"Before we start, however, keep in mind that although fun and learning are the primary goals of all enrichment center activities, serious injuries may occur."

- GLaDOS, Portal (game)

IV: "ELEMENTICS"

Blurb & Logo

*“**ELEMENTICS**” is a puzzle-strategy game where the goal is to balance chance and logic. Operating the engine is a hard thing, rewarding those that learns its ins-and-outs.*

*In “**ELEMENTICS**” you as a player have to balance your strategy between managing the engine energy levels and the engine output by thinking ahead. The game is based probability and chance, but luck can only get you so far!*

INTRODUCTION

The version of ELEMENTICS subject for the last major set of playtests is available as a webplayer version here:

http://goo.gl/3hjHJx
(https://googledrive.com/host/0BzsWrA66m6rvX0ZISmxSbW83ak0/Finalizer01.html)
(requires the installation of the Unity Webplayer, note long loading time the first time it’s run)

The actual final version, which corrects most of the issues detailed in the last chapter is available at:
goo.gl/Sh4W3C
(see Notes: Prototype)

The top link is the version of the game where the set of issues detailed in the last chapter are still in place. The supplied tablet version, and the second link features updates and fixes to the gameplay not implemented and is the “actual” final version of the game.

The game centers around probability and combinatorics on a level which should be familiar for most players and still remain accessible for those who don’t. The game revolves around finding a balance between maximizing points and the combo multipliers against keeping the game running by maintaining the energy levels above zero. To do so, you as a player have to solve two types of tasks, sometimes choosing one at the cost of the other. One involves trying to match the Matrix symbols, while another involves manipulating the chance of a successful outcome. Smart play is rewarded, so stay focused and the opportunities will come!

Figure 24. *The opening screen as seen by the players on starting the game.*

Figure 25. *The logo and texture used for the tablet icon.*

Gameplay Outline

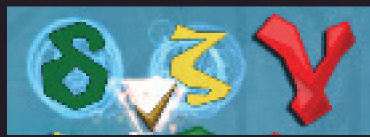
While there is no replacing trying out the game itself, these next pages try to explain how the game functions and how it's played. Further, it's shown how the theory and game design aligns.

GENERAL GAMEPLAY

The game is divided into two phases named "build" and "solve" respectively. The goal of the game is to maximize your points before the engine runs out of energy. This energy is drained whenever the game has to change between the phases, and is regained by solving the probability task called the "key". Points are gained when you manage to solve the matrix, the set of symbols surrounding the key.

In the build phase you use the third currency, "supplies", to create groups of symbols. Groups of different sizes offer different advantages for both types of tasks, and it is important to ensure a variety of groups. Once the supplies are exhausted you can turn on the engine and begin solving the tasks. This removes all non-grouped symbols. To solve the tasks you have to select which groups to use and how many times you want to use each of these groups. Once your have chosen enough symbols to solve the task you can initiate the solving sequence. After a moments calculation you are given the results of the process. For each group, one symbol is selected for each use of the group. If the total outcome across all the selected groups matches the matrix, the primary task is a success. If the chance of a successful outcome matches the Key probability target, resonance is achieved and some of the energy is restored to the engine. A new task is immediately available to be solved after each solving sequence. The generated symbols are used up, and do not count for future tasks.

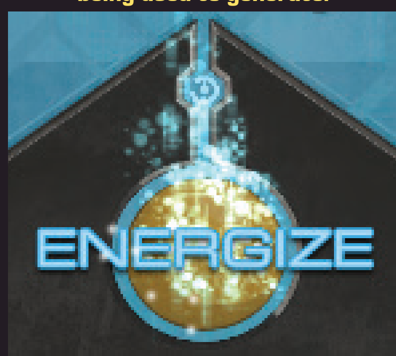
When each of the groups are exhausted and you don't have enough symbols left to solve the matrix, the engine shuts down at the cost of energy. When this happens, each group is broken down and the supplies are restored. The cycle then repeats, with the tasks growing more complex and the accepted deviation from the key shrinks for each iteration. When the energy runs out, the game is over.



Selecting symbols, the create group button appears automatically.

Key and Matrix task, the Matrix changes after every solve attempt, the Key changes every full cycle

A created group and the number of symbols it is being used to generate.



Start the engine and begin solving! This button appears automatically when the supplies are exhausted.

Figure 26. *Images from the build phase of the gameplay.*

Game Atoms

The game in its final form can be broken down into game atoms. From the upper levels, where the goal is a more abstract thing, such as getting the most points, down to the most basic GUI actions. Here I'm going to discuss some of the two most important atoms, and why I believe they satisfy the criteria laid out by Koster.

CREATING GROUPS

The central action in the "Build" phase of the game revolves around creating groups. Each build phase, the player loses whatever groups they had, but get supplies that allow them to build new ones. Each time they come back to the build phase they receive more supplies than for the last round, which allows them to keep solving more tasks even as the tasks grow harder. The most basic action of this process is selecting elements from the grid to make into groups. This is a GUI action, requiring the user to tap the symbols with their finger to select them. The top image in figure 26. shows how this action looks in the game.

INPUT
Touch an element on the grid during the "Build" phase.

ACTION / RESULT
The game checks if the player has sufficient supplies to select an element, if the element already belongs to a group, if the player has selected other elements and then if they have if this element is adjacent to any of these or if the player has selected this element before, and then if this was the last element they selected previously.

Each of these states result in a different outcome. If the element is adjacent to previously selected elements, and the player has enough supplies remaining, the element is added to the selection. Without sufficient supplies, or if the element is not adjacent or already belongs to a group, no change happens. If the element was already selected, it is then deselected if it was the last previously selected element and the player regains 1 "Supplies".

FEEDBACK
The feedback here is visual. If the element becomes selected, it changes shape. The "floating" version of the element is embedded into the surface of the game and a shining particle effect surrounds it. If it is unselected the reverse happens.

Further feedback arises when this brings the number of selected elements up to two or greater. If it does, the "create group" button appears in the lower left, shining in a different contrasting color. If the element is the first selected, it also turns on the "Clear button" in the lower right screen.

If the game does not allow the selection to happen, the element "pops" back into place, as the player lifts their finger off it, showing that their action was unsuccessful.

MASTERY
The stages of mastery are linked to how the player is able to understand the "rules" controlling the selection process, such as adjacency and the ways of deselecting. By seeing the feedback, they eventually learn which ways the elements can be selected. On an even higher level the mastery comes from the user reflecting on whether or not selecting this or this element is "smart" from a gameplay perspective. This type of mastery requires feedback from the game in other areas, such as from the "Solve" phase of the game.

The challenge of creating groups has to fulfill several requirements, which I believe it does and why this is in part why it is a valuable part of play.

First, the challenge requires preparation in the form of forethought and strategic planning. The user is able to see the first task they will have to solve, and can plan at least with that in mind. There are also multiple ways to prepare, in that groups can be of different sizes and compositions. The environment for the challenge is the grid, which can alter the challenge based on previously made groups blocking the user from creating the desired "composition" for their new groups.

The challenge requires basic GUI actions, but there is a skill element to the overall strategy that the user has to consider for lasting success. Failing can not be defined in this context alone, but has to include solving tasks, this delays the “mastery” aspect of the atom and separates it from the Input, Action, Feedback cycle. However the creation of groups has “fail” states, when the user makes ill considered groups and can’t solve as many tasks they were hoping to.

NOTES
Something that is left unmentioned here is the secondary purpose of the build phase. Reflection. The build phase allows the player to regain their balance and do a series of lower impact actions (selecting elements and creating groups) compared to the actions of solving tasks in the solve phase of the game.

By resetting the users groups with each full turn of the gameplay cycle, these actions also take the form of something akin to free-from exploration. The user is free to simply make aesthetic configurations, or try out unique compositions that they want to play with and test out. The theory on learning stresses this need for reflection in gameplay, and this is the primary reason that the game did not go the direction of continuous gameplay (where input is constantly required) but split into two phases.

By looking at the hierarchy of players’ needs in figure 4. this phase represents a “safer” and more easily grasped phase than the more complex solve phase. I believe that breaking the game into cycles of high and low levels of engagement prevents the player from “burning out” on gameplay too quickly, and lets them come up with strategies and plans in their own time.

- Does the challenge require preparation?

- Does this preparatory step pass these steps as well?

- Does the challenge allow for multiple ways to prepare?

- Does the environment for the challenge affect the challenge?

- Are the rules of the challenge defined?

- Can the rules support multiple types of challenges?

- Does the challenges require multiple abilities to pass?

- Is there skill involved in using the ability? (and if not, is it a fundamental move, one of the innermost “nests”)

- Are there multiple success states to beating the challenge?

- Do advanced players not get a benefit from sticking to easy challenges? (referred to as the mastery problem)

- Does failing the challenge have a cost?

Figure 27. *A repeat of the requirements for game atoms, set by Koster (2004). Not a recipe for fun, but intended as vital components for any game atom.*

Game Atoms

SOLVING TASKS

If the central atom in the build phase of the game is to create groups, the central atom in the "Solve" phase of the game is solving tasks. This requires the groups created in the build phase, and is again simple GUI actions in the form of pressing the groups that the user has made. Each time the user touches on a group, the lower left side counter increases by one, this means that the group will be used to generate one element for solving task. Since how many times each group can be used is decided by its size, the user can potentially press the group multiple times to increase this number. Alternatively, they can loop the counter back to zero or press the "Clear" button to deselect all groups. This counter is seen in the third picture from the top in figure 26. After selecting sufficient groups, which is decided by the number of elements in the "Matrix" task surrounding the red circle, the user can press the "Solve" button as it starts glowing bright. It is this sequence of events that make up the bulk of the game and require the most thought and consideration from the user.

INPUT

Again, simply touch the groups that remain on the grid. When the player is in the solve phase, the rest of the elements that aren't made into groups become invisible.

ACTION / RESULT

The game calculates if the player has made sufficient selection, and if the group can be used for as many "draws" as the player is trying to. If a sufficient selection is made, the "Solve" button becomes active and usable. Pressing the solve button results in the screen changing drastically, and the full set of feedback is given to the user.

FEEDBACK

The counter adds or reverts back to zero, depending on how much the user can and is drawing from that particular group. A group which is used for one or more "draw" glows faintly, and has a counter that indicates just how many.

When pressing solve, the player is shown a wide range of feedback. The outcome of their selection with which elements they managed to generate, and the probability they had to succeed compared against the Key probability target. Further, they see how many points and how much energy they are awarded, and what their current combo multiplier is.

MASTERY

The types of mastery involved in solving tasks are several. Ranging from a mastery over the basic rules governing how many times any group can be used, and how the basic input system works (such as clearing the selection, selecting a total equal to the task size to proceed etc.), to the much more nebulous areas of understanding the probability that governs their chance of success. The feedback screen will give the user an idea of what their selection resulted in, and why it turned out the way they did. They can judge their fortune (or misfortune), and learn from it to improve their task solving abilities as the game progresses.

In essence, solving tasks is also where main part of the "learning" is made apparent. If the user improves in their task solving ability over time, they are also showing improved skills in calculating probability and combinations, whether they are explicitly aware of it or not.

This part of the game required the most attention, as it is what by which the game stands or falls. I do believe that, despite certain flaws such as the issue of "unsolvable" tasks and problems with scaling the challenge and complexity correctly over time, that it is works as one of the core gameplay atoms.

To go by the criteria, the challenge does require preparation and planning, with multiple ways to achieve this. How to select and which groups to select is essential to solving the tasks successfully. On a second level, the topology of the challenge, or environment, is informed by the build phase and the task itself. By what you have to chose from, and which choices you can make.

The challenge is clearly defined in the form of the two types of tasks, the key and the matrix, and these challenges can take on a wide range of difficulties. And the challenge requires layers of skills on part of the player. There are strategic considerations to how to balance energy versus point gain when the Key and the Matrix are exclusive (aka, scenarios when you can only really solve one or the other), and balancing gambling with chance compared to making the "safer" more guaranteed choices. Further, failing has a cost and players can fail or succeed by degrees.

NOTES

The solving of tasks represents the critical game mechanic for teaching the player about probability and combinatorics. The underlying model for each task represents the probability and combinatorics model of drawing without replacement. The task, removed from the context of solving math with pen and paper, is made intuitive through visuals. This ensures that people with no or little knowledge of the theory can easily come up with a solution that at least has a chance to succeed.

Since a chance of success can often be rewarded with actual success, players who don't understand the underlying math are still drawn into the gameplay and can rely on the probabilities shown under the "Resonance" (as seen in the lower image of figure 27.) to guide their decisions.

From the earliest versions, the solving tasks mechanic has remained more or less the same (with the late addition of the probability task known as the "Key"). It is a mechanic that is aligns naturally with the desired mathematical theory, and can still provide a wide range of challenges. The difficulty and complexity that occurs by introducing more elements to the task is exponential, but through visualising the problems, the user can make educated guesses and become more informed about what worked and what didn't by the feedback they receive. This builds a mastery of not just the game atom itself, but also an understanding of the underlying mathematics!

Selecting from the created groups. Solve button turns on automatically.

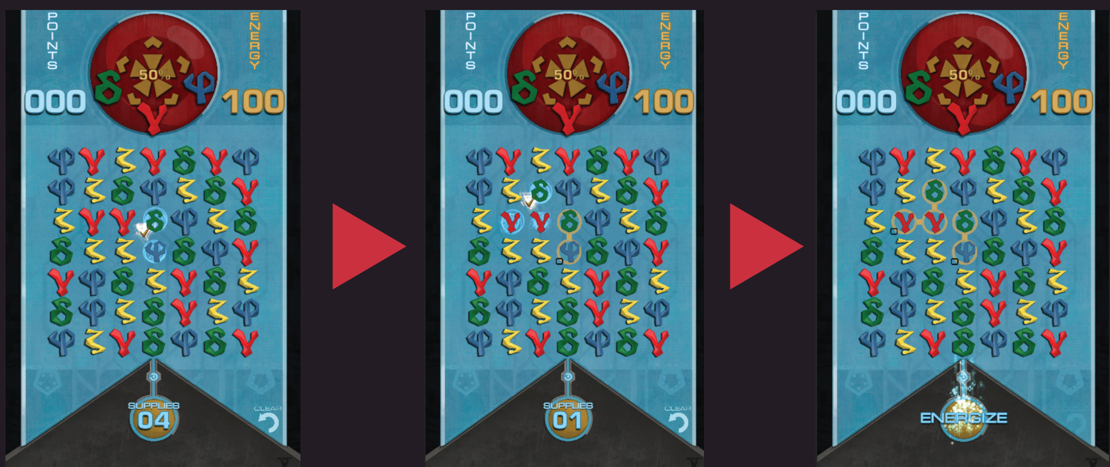
The feedback splash screen. Here the user can judge the outcome of their actions.

Figure 27. *Images from the solve phase of the gameplay.*

Overview

WHAT: Starting the game. The player can read instructions, or pull up the "curtain" to start gameplay. The player starts with 100 energy, 6 supplies and 0 points. The matrix and key tasks are shown on the red circle.

WHY: No learning content or core game mechanics until this stage.

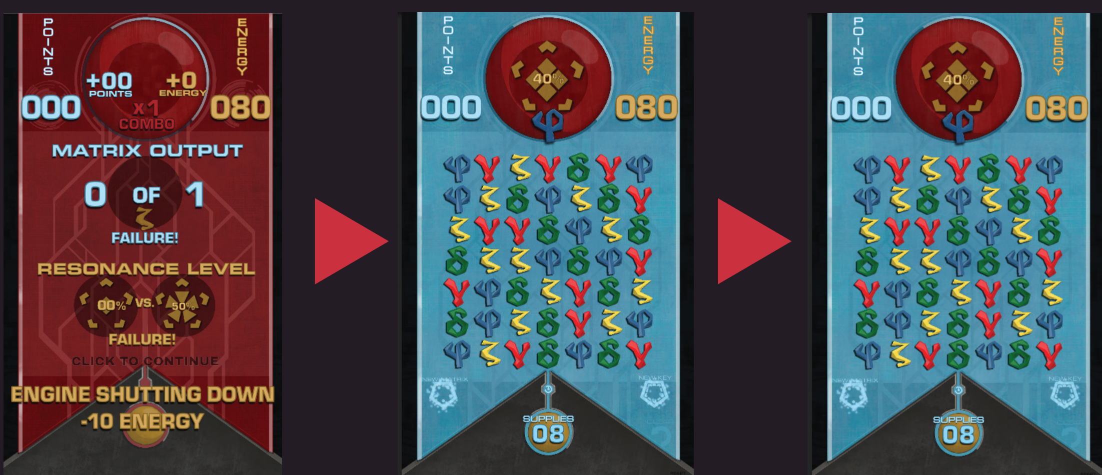


WHAT: The build phase, the player creates groups (with 2 or more elements) using their "Supplies". By pressing the gold create button, a group is formed. When the supplies are used up, the "Supplies" graphic changes to a button. Pressing the "ENERGIZE" button initiates the solve mode at the cost of 10 Energy.

WHY: This is the secondary phase of the gameplay. It gives the player a break from the demanding tasks, and instead allows the user to relax and think strategically about how they want to make their groups and prepare for the next "Solve" phase. As the game progresses, the users gets more options in the form of more supplies. This phase allows for experimenting with different compositions, so that they can see how this affects the probabilities in the Solve "phase" of the game. The phase is structured so that even haphazard group creation isn't completely detrimental to the Solve phase, but good play is rewarded.

WHAT: The solve phase clears all the ungrouped elements, leaving only that which those that the player made in the build phase. These groups are then selected, with a counter indicating how much the player intends to use each group for. When the total number of uses matches the number of symbols in the task, the "SOLVE MATRIX" button starts glowing. Pressing it takes the user to a feedback screen which shows how he or she did, and the rewards of their performance. Continuing, the player gets a new matrix task (the key stays the same) and the previously generated rune are "used up", for the remaining of the cycle. When this capacity to solve is used up, the "Engine" shuts down at the cost of 10 energy, and all groups are reset. The player starts a new build phase, this time with some more supplies to make groups from.

WHY: The essential game mechanics in solving the tasks are built on probability and combinatorics. The distribution is akin to a hypergeometric distribution (drawing without replacement) and is made intuitive for the players by visualising the content of the groups and the task. This allows for intuitive leaps of judgment and a way to "guess" at a good answer irrespective of the ability to actually calculate it. Players get enough feedback for it to entice them to try alternative paths to victory, and will then get to see how drawing multiples or different group compositions and sizes affect the probabilities involved. This is where the game shows off the "learning content".



Overview

WHAT: The game repeats, with each cycle involving on the whole more complex tasks (more symbols in the matrix). Also, the player has more room to try different group sizes and compositions. Unless the player is exceptional, the energy slowly trickles down to zero and the game warns that the last cycle is engaged.

WHY: The game slowly increases overall complexity, and reduces the acceptable threshold for hitting the probability task (the key). This keeps the player from feeling that the game is too easy as they master the game mechanics. Also, as complexity increases, it requires a higher understanding of the underlying principles to maintain good results. This means that the player is forced to engage with the learning content on the upper levels of Bloom's pyramid, using the basic theory they've accumulated to form the basis of understanding the more complex content.

WHAT: The last feedback the player receives as the game ends shows how well they did, both in total and for each of the two task types. The players can then put the game down until the next time they feel like playing, or use the accumulated knowledge to try and beat their own score or that of a friend.

WHY: The feedback also allows the player to experiment with alternative approaches and better judge the overall outcome. Such as focusing primarily on one type of task over the other or similar. Ideally the game ends with the user eager for more, instead of them tired of the game because this cheapens the impact of the content.

Gameplay Strategy

The game operates on multiple levels of strategy, informed by the interplay of the two types of tasks. This balance between the two, the "random" matrix task and the more deterministic key task, game mechanics.

The game is playable with the player ignoring this dynamic, but gains strategic depth when the player starts to consider them. The game design tries to balance this process of mastery, by working up the steps of Bloom's pyramid of learning and using the principles outlined in the theory section. This section will deal with some of the strategies that aren't necessarily visible to a first-time player.

EFFECT OF TASK SIZE

One of the things that players quickly learn is that generating one element per group results in a rapidly diminishing chance of success as the task size grows. If the task consists of four elements, and even if the player uses four groups each with two thirds chance to generate a desirable element, the resulting chance of success is 16 / 81, just below 20%. Task size results in an exponential increase in difficulty, in terms of probability. The probability of smaller tasks, with one or two symbols, can to some extent be calculated on the fly, but once the task size grows the best the player can hope is a rough estimate.

This is intended, since it is not desirable for players to actually calculate the resulting probability, and it adds necessary depth to the game. When players can consistently solve smaller tasks, the larger tasks still present a challenge.

MEANINGFUL CHOICE

In order to ensure that the players still feel that they can impact the outcome, even for large tasks, a lot of work has gone into calculating the probabilities of different scenarios.

One of the strategies that players learn through the feedback splash screen is that drawing multiple elements from a group is a requisite to alter the probability significantly. This means that players soon start testing out how larger groups alter the outcomes and make it possible to hit the "Key" tasks consistently.

KEY VS. MATRIX

The game requires that the player doesn't ignore either task type if they want to reach the highest potential score. Key tasks and resonance allows the player to keep going, but only solving the matrix task gives the player points and builds the combo multiplier.

The game helps this realisation come naturally to the player by the elements becoming "inactive" when used to generate a solution for the matrix. This means that as the "Solve" phase progresses, each group becomes more and more drained of available symbols, resulting in them eventually only having one type of symbol remaining. This gives the player an excellent way to guarantee that they are able to solve the matrix. The key is, however, only successful if the player hits a specified probability. So when the player can guarantee success for the matrix, it often means that the key can't also be a success. Which gives the player a choice, if they want to purposefully reduce the chance of getting points, for the possibility of gaining energy and keep playing.

To counteract this dynamic, at the start of each solve phase, the player has a much better range of options, and has a greater chance of finding some combination that satisfies the key task.

CHANGING THE TASKS

Something the players also discover later in the gameplay is that they can change both the key and the matrix once per cycle. Knowing when these are most efficient isn't always obvious. Sometimes the player gains more by not skipping an unsolvable task, if it is small enough, but if the player delays using the abilities they risk losing out their "free roll" for each round.

"The universe is not required to be in

perfect harmony with human ambition."

- Carl Sagan

V:POST-MORTEM

Reflections

This post-mortem section deals with how the game, in the incarnation explored and detailed in this report, was received, how the different mechanics work (or don't), and a reflection on the overall process.

While the final game is fully playable, and even enjoyable to some extent, there are several issues that I feel are worth exploring and explaining further. I do believe the game offers valuable ideas and insights into what works and what doesn't for a game like this, where there is a secondary intent that goes beyond just making a fun and enjoyable game.

I would also again like to mention that some development and tweaking of the game was conducted after the time of writing. By comparing the final game to the one described here the reader can get insight into how the last set of "fixes" work out when compared against the issues that were addressed in the "Game Dev" section.

Ultimately, I am satisfied with the end result, despite its flaws. Not only has the project been an interesting venture into the field of game design, a field which I feel is very much essentially interaction design at its purest, but also because it has given me the opportunity to learn what I hope will be valuable and important tools for my future working as an interaction designer.

For the these pages of reflection, I will revisit some of the game design and learning theory that went into the creation of the game, as the were important contributors in the game ending up as it did.

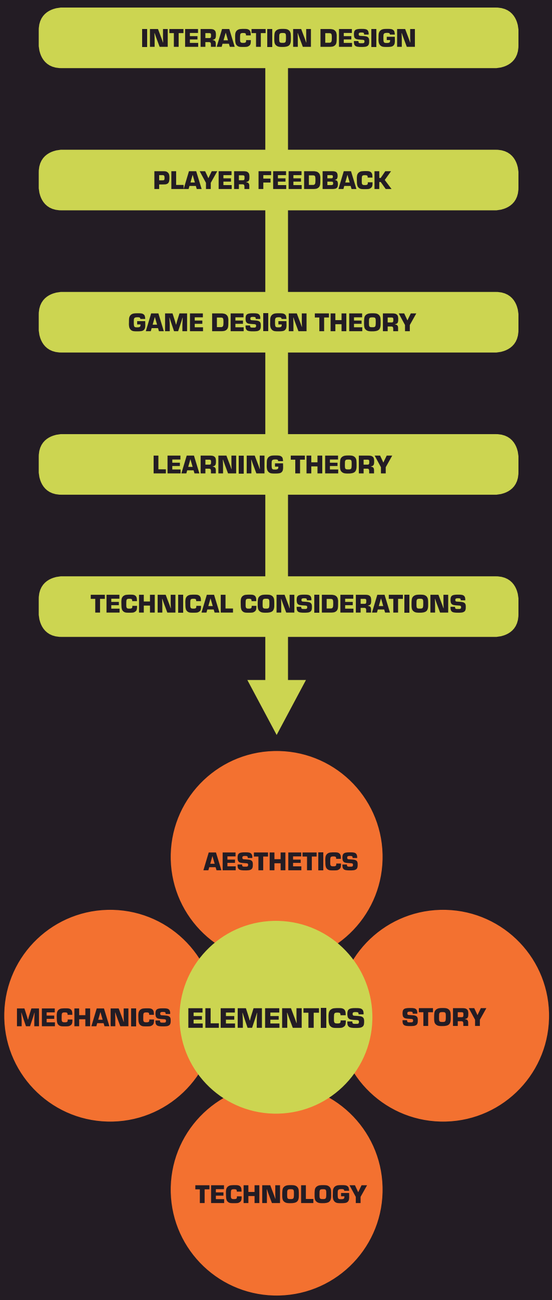


Figure 28. *What went into the creation of the final game.*

Game Elements

The initial concept, "Dreamweave", shared some similarities with what would become the final version, but underwent major changes from the early paper prototypes to the final digital - which I hope this report has illustrated. Using Schnell's four elements to illustrate these changes, it becomes easy to see why the game couldn't have stayed in its initial form. The four elements are, or should be, connected in order to create the best possible game. For example when the game was converted from the analogue to the digital, the game would have suffered if it had tried to stay the same, and this change of technology had to be reflected in the other three elements (which was also the case). Next we take a look at each of the four elements and what the game tries to do with them.

TECHNOLOGY

Working with the technology was a big part of this project. Unity, coding, game graphics are all what makes the game playable. At several points in the process, both before and after the game had a functioning digital version, either I or one of the many playtesters would come up with features that seemed interesting only to find out it would have to be altered due to the technology. The reverse also occurred, and many times the technology enabled features that in any other media I would have had to ignore for being infeasible.

Even playing the game in the web version compared to the intended tablet version reveals how technology shapes the experience of the three other elements.

MECHANICS

For this design the mechanics came first. It can't be denied that "ELEMENTICS" represents a bottom-up design process at the core. The need to use mathematical theory to base the game mechanics on did initially remove the focus on the story and the narrative in particular. This was something I worked hard to correct in the later stages of development, which succeeded in part. I believe that the eventual aesthetic and story do help the user grasp the game mechanics more easily, but this relationship could and should have been strengthened. The relationship between the game mechanics and the technology is much more satisfactory, in that these two work off of each other rather well. Through brute computing power, the game can check hundreds of combinations and generate the required "randomness" in an instant, which make several of the game mechanics possible.

STORY

Just like the two "hard" elements combine well, the two "soft" elements do the same. Both the story and the aesthetics of the game shapes the other. The story chosen for the game, tries to communicate the cyclical nature of the user, and the central "struggle" that the player is facing - *the engine is losing power, and you need to do your thing to keep it running* - The story tries to add a need to care beyond just the desire working the game's mechanics. When the story and the game mechanics, align and come together the player feels immersed in the gameplay and cares beyond just wanting a better high-score.

AESTHETICS

The way the aesthetics work with the mechanics and the technology is primarily to create intuitive controls. Communicating to the user which elements are "usable" and which are decorative, or inactive at any point in the gameplay. While the aesthetics work for the most part in terms of communicating function, the story they try to create doesn't adequately bring extra tension to the gameplay. Users played the game, and often they thought the mechanics were interesting, but they didn't go away with a good sense of purpose or narrative to it. Like an arcade-game. This is not all negative, as it ensures that the user instead engages with the game mechanics (where the educational content resides), but a better balance could have been reached. Also, creating animations, even 2D ones, was a major hurdle in game development, and was down prioritised due to the time investment they would have required. A clear case of the game not fully utilizing the technology from an aesthetic and story perspective.

Game Ideas

Searching around, rather extensively, on the App Store and Google Play for games that centered on probability mechanics ended in surprisingly few results. Beyond the obvious, such as games centered around gambling (of which there are a whole host), most of the games that make use of probability simply use it to add unpredictability to another mechanic.

Few games besides those in the gambling category use probability theory as a basis for central game mechanics. So despite some of the issues that ELEMENTICS suffers from, I think it centers around an original and unique premise. This was not just an observation by me, but also supported by comments from playtesters. I think that some of the techniques used by the game are interesting beyond just the game itself, but also from the perspectives of game-based learning and game design.

REWARDING THE PLAYER

One of the problems I encountered when trying to create game mechanics based on probability theory, was the problem that they all boiled down to chance. Random chance, even when it can be "adjusted" either positively or negatively by the player can be too much. Something you find in several gambling style games, like poker, is that the game is not all about chance and probabilities at all, but often add a social component to the gameplay. Purely chance based games like slot machines,could be argued to not be games at all, depending on the definition of games you adhere to. I believe this boils down to an issue of rewards. Where in the gameplay is the player rewarded? The problem with most of the probability themed game mechanics was that they simply rewarded the player on the final outcome. The player could give themselves some chance of success, but beyond that it was only if they were "lucky" that they would get any sort of reward.

What the first "Dreamweave" prototype took steps to solve this. By whittling down the elements in each group or web, the player could usually reap

Figure 29. *Layers of rewards. Often, you can't guarantee an outcome, but you still won't be rewarded if the action was the right one. This can always be taken up one step further in the chain of causality.*

some guaranteed successes. The final game builds further, using the technology to explicitly reward the "intent" of the player and not just the outcome through the use of the "Key" probability task.

This is an advantage of the choice of platform. Computers today are capable of computing variant scenarios at a much greater rate than people, at least if they fit into a mathematical model, which games often do. This means that it is possible to reward the player for making good decisions, and then further reward them for "getting luck". This reduces the player's frustration when they know they are doing the right things but still fail on a chance based outcome. Further, it helps motivate the player to engage with the game mechanics, and strategies about what is and isn't a good choice since this is now another way to be rewarded all in itself.

I believe this combination of rewards is a very interesting aspect that is very suitable for chance based games such as ELEMENTICS. It makes the game something more than just "gambling" and adds depth to the game. I find this idea / mechanic to be the most interesting idea from ELEMENTICS from a game design perspective. Interestingly, this mechanic is also reflected in math tests, where the student is rewarded for a good "try" even if the result is flawed.

TWIN TRACKS

Not unique to ELEMENTICS is how the players must juggle several “currencies”. In the case of this game, the player sometimes has to reduce his or her chance of gaining points, the primary goal of the game, to gain energy which keeps the game running. Of course, keeping the game going sounds like the obvious solution, since it will give more opportunities to gain points later. This is however counteracted with the fact that the “Key” tasks which award energy are the harder of the two, especially for tasks with multiple elements. Observed in the playtests, and when playing the game myself with a good understanding of the underlying models, was that players could fairly easily get an intuitive understanding of good and less good choices for solving the “Matrix” tasks, but translating this knowledge into matching a specific probability was much more difficult.

But even so, to avoid the player feeling frustrated at having to chose between one or the other. The two tasks have a dynamic relationship where at the start of each “Solve” phase, the player has the most options and potential to solve the “Key” task, but as the phase continues the ability the ability to solve these go down but the ability to guarantee success for the “Matrix” tasks. This ensures that the player more often than not has one path to “victory”.

ON ABILITIES AND COMPLEXITY

In the “Groundwork” section of this report, I remarked that complex games were more forgiving. In truth, they are also easier to make. Even the initial paper prototype had more “abilities” that the players could utilise, like re-rolling , doubling points, or setting up arbitrary game rules such as one type of element having larger chance of occurring for the remainder of the phase, compared to the final version of the game. As the game design matured, most of these features detracted from the core gameplay mechanics rather than improved on them, and when examined further, they usually did not satisfy the requirements for game atoms which they could be modeled as. For each iteration of the game some of these features were cut, and it almost always resulted in an improved more streamlined game experience. Simplicity is a boon, but while finding which features to cut wasn’t so difficult, it was much easier to try to patch gameplay issues by adding more complexity. The early implementation of “time” as a way to add tension to the game is a prime example of one such flawed band-aid approach. Especially since it also came with time-based mechanics, where the player could buy more time by using points or power (the previous name of “supplies”). The change from time to “Energy” did come at a loss of tension, but felt more natural to the story and aesthetic elements of the game, as well as streamlined the gameplay experience.

In the end it might be that even the ability to change one or both of the “matrix” and “key” tasks once each full cycle of the game is too much. I do believe that this focus on a few solid core game atoms keeps the player engaged with what is the educational content of the game, and is also in line with the choice of platform and type of game. Players expressed a similar sentiment, and this aim to focus and simplify the game, led to them understanding the game more quickly.

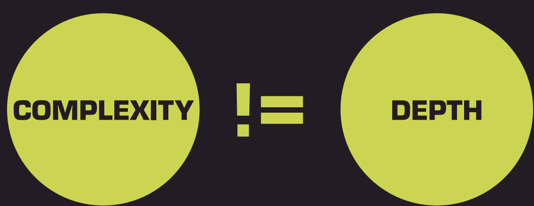


Figure 30. *Complexity is not depth, and the two should not be confused. Cutting complexity that doesn’t satisfy the criteria for game atoms isn’t the same as cutting depth. This does not mean that games can’t be both, but each feature needs to be examined if it is “empty” of true depth or not.*

Reception

The ease of sharing and distributing the game in its web-version, allowed me to get input from a wide range of players many whom had no reason to go "easy" on me (except for a general bias for people to err towards the positive in such scenarios, a factor that shouldn't be ignored). Overall, the reception was decent, and people seemed to enjoy at least the first couple of games sufficiently for me to be positive. However, I'd like to address what I consider to be the two biggest problems with the game, which reflects the most common criticisms from playtesting, how this impacts the gameplay, and what I think should have been done to prevent them from happening.

NEEDS INSTRUCTIONS!

This was the most common complaint by far in the cases where I purposefully left out any instructions on how to play the game. While I, for most of the playtests, simulated a tutorial by providing a short written summary on the gameplay I did on occasion leave it out. Also even in the later playtests, where an in-game tutorial was present, I sometimes instructed the players to ignore it until after their first playthrough (or they did it without me asking). This revealed that despite the cutting of unnecessary features and complexity, the game wasn't always intuitive to the players, and required some time to figure out. Without a tutorial, most player misunderstood one or more mechanics of gameplay until later in their first playthrough. Some of these players reaching the stage where they gave up rather than continuing. Needless to say, this is undesirable from both the perspective of fun and that of game-based learning.

This one of the limitations that came about from the way the project time-management was handled. The game didn't reach it's final incarnation until too late in the process for me to implement a "forced" tutorial that gates the player through the core mechanics. By adding an "optional" tutorial or manual a band-aid is at least in place for this issue, but it's not uncommon for mobile gamers to ignore such features. This results in some potential players giving up on the game without ever engaging with the game content. The playtest sentiment is clear - the game needs some sort of instructions, and this should preferably be pushed on them in the form of say the first "build" and "solve" cycle of the game guiding the user in a non-obtrusive manner.

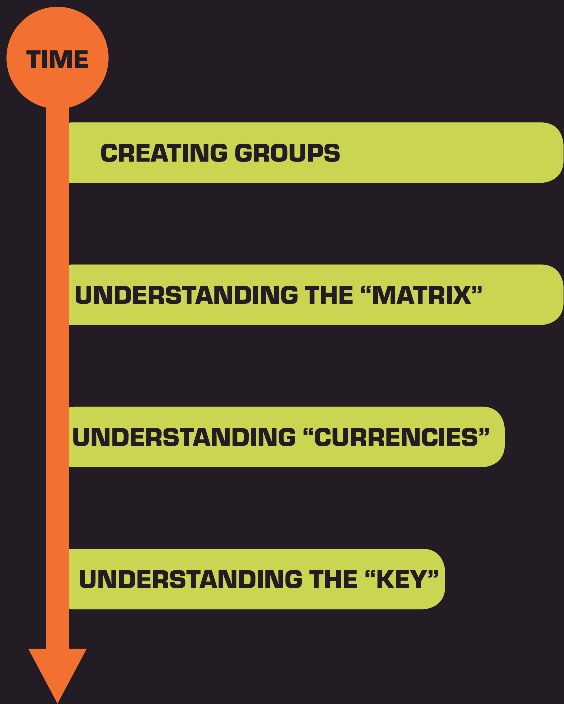


Figure 31. *The crucial game mechanics the player has to learn, where they are presented in the game, and how large impact they have. For the upper two, not understanding them quickly leads to quitting the game, while the latter severely reduces the player's enjoyment. Ideally, the game would guide the player to understand them without telling, so that the user feels the enjoyment of "mastery"*

LACK OF TENSION

A second complaint was that the players felt the game lacked tension as it came closer to the end. This is in part intended due to the "puzzle" style and nature of the game. However, observed in the playtests was that the players peak engagement was reached in the middle of the gameplay. At this point, the players are given enough supplies to create multiple group variations and have usually worked out some strategy that they apply. Despite the game gradually scaling the difficulty however, this engagement did not keep steady or increase, but rather diminished as the player neared the end of the game.

When the players interest in the game waned, they seldom put it down for later, but finished the round by starting to make worse choices, sometimes even clicking at random just to finish it. This contrasts greatly with how they played at peak engagement. Then the players had fun trying to work out the optimal solutions for the Key and Matrix tasks, exploring various approaches. This made them learn and build an understanding of the probability mechanics corresponding to the upper levels of Bloom's pyramid. So when they went past this point, they regressed in terms of engaging with the educational content, and at best engaged only at the lower levels by applying memorized gameplay.

This illustrates a problem of the game mechanics and story not coming together well enough, which the theory did warn about for bottom-up designs. The late implementation of the idea of the "Engine" as a central narrative I think is a big part of this problem. It doesn't adequately build tension, expressed through feedback to the players or altered mechanics and aesthetics. Perhaps in part because the end outcome of the game is always the same, the player will run out of energy. While perhaps some other measure of success could be added, an alternative victory state if the player manages to keep going past a set point, or fail state if they fail too quickly? This would give the player another reason to care, rather than just the idea of beating a previous high-score.

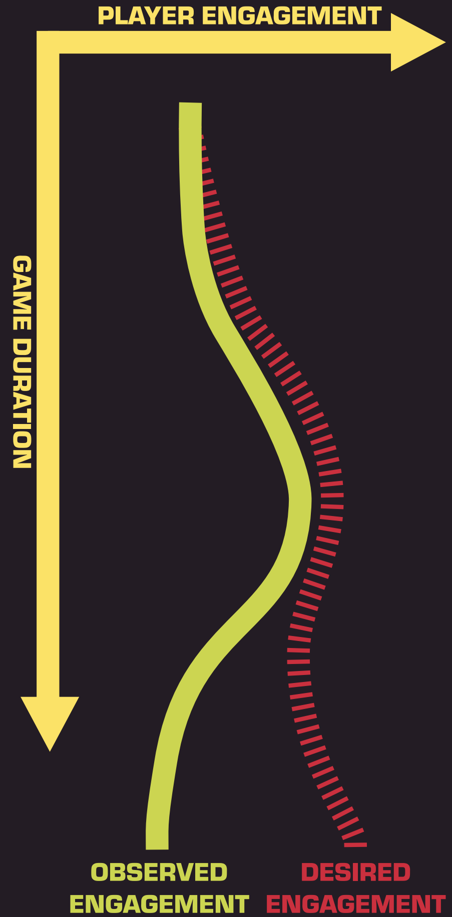


Figure 32. *The observed player engagement compared to what would be most desirable. The "band-aid" fix the game has implemented for this issue is to shorten the game length, to ensure that the peak drop-off doesn't kick in before the player is finished. A more proper fix would be to ensure player immersion and engagement through the story element of the game.*

The Process

The process of creating the game "ELEMENTICS" has been long and interesting. Not only has it required me to learn the basics of game development, something which I've had personal interest in learning about for a long time, but it has also allowed me to work on a project from the very beginning to what can be considered a finished product. While not optimised, the final game is in a state where it could be uploaded to the target marketplace it has been developed for, the Android marketplace "Google Play". However, I would like to remark on some of the steps taken over the course of the project and how they have impacted the final outcome.

DIGITAL TRANSITION

One of the most important and influential parts of the process was the early stages of paper prototyping. Here I worked out and tested various game concepts based on different parts from the math curriculum. However, the intent was always the end goal, and is something I did not consider sufficiently in the early stages of the process.

While there is extensive overlap between what is and isn't enjoyable gameplay for both types of media (analogue versus digital), there are several elements that do not carry over. One example I would use is the rolling of dice, as this was central to the game "Dreamweave". This is a form of interaction that is much more enjoyable in physical form, than pressing a button and getting the same level of randomness immediately. The full extent of the act doesn't carry over automatically. Breathing on the dice, slow-rolling or crossing your fingers as they roll over the game board are all things that add enjoyment to such a simple act. In the first digital versions, the player was however treated to the outcome immediately on the touch of a button. The essential outcome is the same, some random element is being selected, but the acts are and feel completely different to the players. This is but one of many such "mistranslations" that I had to consider in the transition to a digital media.

While the process in part had to be conducted in this manner because I had no way of creating a digital version of the game at that stage, I do believe it was a flawed way to go about things. A process that more thoroughly tried to simulate digital gameplay, at the cost of having the paper prototypes function on "their own" so to speak, would have resulted in the transition going a lot smoother and maybe to a better end result. As it were, the "feedback" parts of each game atom had to be altered to find ways to be as fun as the paper versions, which wasn't always as easy.

NAIVETÉ

Learning about software development took a lot more time and work than anticipated. There are some things that are hard to make and express in code, that one doesn't think about for boardgames. However, this was a valuable lesson from the process and something I wouldn't be without. Trying to envision mock code for any new game mechanic to judge its feasibility would have saved time and effort, but at least for this project my way of going about things were not hamstrung by thinking or knowing that certain features were infeasible. This led to a rather free-form early process and allowed me to explore a greater range of game mechanics, at the cost of a significant time investment. In the end, my lack of prior knowledge in software development impacted the final result for both good and bad.

TUTORING

As this was a one person project, being able to bounce ideas off my tutor was of immense benefit. It was easy to get tunnel-vision on certain features and ideas working alone, but communicating with someone who'd followed the process from the start helped me steer clear of most such pitfalls and was invaluable. Of course the importance of player feedback cannot be understated, but they couldn't give an informed opinion on the direction of the process as a whole.

Learning & Goals

The start of this report explained the goals for this project, and at this stage I'd like to revisit them to show how the end result compares. Based on my own opinion and that of the playtesters, I think that the game satisfies the goals I set for the project. Ideally, this is where I would present hard-data to back up this assertion, but the work such an undertaking would have involved has instead gone into further game design and development in an attempt to try and improve issues dealt with in this report. The goal was always the creation of a functional prototype, rather than testing the theoretical and actual learning outcome of the game, which is how I defend this choice.

FUN

The game was intended to be a fun and engaging game over anything else, and despite the issues I've covered so far, the game was engaging enough for most playtesters to volunteer for multiple playthroughs without me asking. The game's flaws in balancing the four game elements does impact the fun and engagement, but these are fixable issues for the most part. Overall, I consider the game satisfactory but still lacking a bit in terms of the original project goal.

LEARNING

While a secondary goal, this is where I think the strength of the game design resides. On the creation of a game with the secondary intent of teaching probability and combinatorics, I am very satisfied.

The game is solidly founded on probability theory and combinatorics, something which was recognized by most playtesters without any of them feeling that the game's main focus was to "teach". Through mastering the game mechanics, the players are in many ways solving smaller mathematical problems - only they are not felt or perceived as such. The game is able to cover an extensive part of this theory, from simpler concepts such as what is percentages or fractions, to more advanced concepts such as drawing without selection. All of this without overwhelming the user or demanding prior knowledge in these areas. Gradual increase in difficulty and complexity to manage cognitive load, as well as natural "breaks" for reflecting on the gameplay are two examples of implemented learning theory for game-based learning. The less rigid structure and depth of the game mechanics also gives the user room to work up the levels of Bloom's Pyramid. Though there is a need for a somewhat ham-fisted tutorial to get the basics of th gameplay in place, once the players get past this hurdle, they can quickly try out and "play" with the games mechanics to test ideas and strategies.

It is hard to quantify the learning without significant testing, and further if the resulting understanding is transferable to other contexts (such as a math test dealing with these theories, or other real life applications). Some level of tacit knowledge is clear going by playtest observations where players steadily improve their "skills". Some could also discuss the reasoning behind their actions, giving hope that some of it is also "explicit" and transferable learning derived from play.

FUNCTIONALITY & PLAYERBASE

For the issue of functionality, I refer to the final prototype. In testing it has not revealed any critical bugs or crashes, and the important interactive elements are efficient and intuitive.

For the issue of the playerbase, the game has been tested on a wide range of players, down to those just starting to attend middle-school. In no playtest did a complaint or comment surface that the game felt targeted at any particular group in terms of both the aesthetic profile and the gameplay mechanics. I take this as a confirmation that the game is of sufficiently "neutral" character that it could interest a wide range of players and that such an interest would instead be mostly governed by the type of game or gameplay these users were to desire.

With this, the bulk of this project report comes to a close, it is my hope that it has given an adequate idea of what has gone into creating "ELEMENTICS".

Credits // Sources

REFERENCED SOURCES:

Ainsworth, S. E., Benfort, S., Habgood, M. P. J., *Endogenous fantasy and learning in digital games*, SAGE, 2005

Andersen, K. A., *The use of game elements for increased motivation in game based learning*, NTNU, 2012

Boyle, E. A., Boyle, J. M., Connolly, T. M., Hainey, T., MacArthur, E., *A systematic literature review of empirical evidence on computer games and serious games*, Elsevier, 2012

Brom, C. & Sisler, V., *Designing an Educational Game: Case Study of 'Europe 2045'*, Springer-Vederlag, 2008

CapCom.com, online resource, available at http://www.capcom.co.jp/ir/english/business/market.html (retrieved June 2014)

Charsky, D., *From Edutainment to Serious Games: A Change in the Use of Game Characteristics*, SAGE, 2010

DragonBox, online resource, available at http://www.dragonboxapp.com/ (retrived March 2014)

Forehand, M., *Bloom's taxonomy: Original and revised. 2005*, In M. Orey (Ed.), Emerging perspectives on learning, teaching, and technology. available online at http://projects.coe.uga.edu/epltt/ (retrieved February 2014)

Greitzer, F. L., Huston, K., Kuchar, O. A., *Cognitive Science Implications for Enhancing Training Effectiveness in a Serious Gaming Context*, ACM, 2007

Hoi, C., Howlin, C., *How to Build Serious Games*, available online at http://www.dcs.warwick.ac.uk/~nikos/CS406/StudPres2007/HowtoBuildSeriousGames.pdf (retrieved February 2014)

Kaptelinin, V., *Affordances*, In: Soegaard, Mads and Dam, Rikke Friis (eds.), "The Encyclopedia of Human-Computer Interaction, 2nd ED.", Aarhus, Denmark, The Interaction Design Foundation, available online at http://www.interaction-design.org/encyclopedia/affordances_and_design.html (retrieved June 2014)

Koster, R., "A Theory of Fun for Game Design", O'Reilly Media, 2004

Koster, R., A Grammar of Gameplay, Game Developers Conference, 2005, available online at http://www.raphkoster.com/gaming/atof/grammarofgameplay.pdf (retrieved February 2014)

Koster, R., An atomic theory of fun game design, 2012, online resource, available at http://www.raphkoster.com/2012/01/24/an-atomic-theory-of-fun-game-design/ (retrieved February 2014)

Marketingland.com, online resource, available at http://marketingland.com/report-3-64362 (retrieved June 2014)

Okan, Z., *Edutainment: is learning at risk?*, British Journal of Educational Technology, 2003

PISA, 2012, online resource, available at http://www.pisa.no/hva_maaler_pisa/matematikk.html and
http://www.pisa.no/pdf/pisa_2012/pisa-rapport2012.pdf
(retrieved March 2014)

Rosewater, M., *Some nontraditional card design, Top Down and Goal*, DailyMTG.com, 2003. available online at http://www.wizards.com/magic/magazine/article.aspx?x=mtgcom/daily/mr75 (retrieved June 2014)

Schell, J., *"The Art of Game Design"*, Elsevier Inc., 2008

Skillz.com, online resource, available at http://skillz.com/blog/2013/03/05/65-mobile-gaming-stats-to-impress-your-friends/ (retrived June 2014)

Young, M. F., Slota, S., Cutter, A. B., Jalette, G., Mullin, G., Lai, B., Simeoni, Z., Tran, M., Yukhymenko, M., *Our Princess Is In Another Castle: A Review of Trends in Serious Gaming for Education*, SAGE, 2012

ADDITIONAL RESOURCES:

Stackexchange.com, online resource. Questions (by me) and answers available at:

http://stackoverflow.com/questions/23656191/all-combinations-over-multiple-lists?noredirect=1#comment36354204_23656191

http://math.stackexchange.com/questions/786472/formula-when-drawing-from-multiple-urns-probability

Gamasutra.com, online resource, used for Post-mortem examples, game inspiration, game design and development theory and tips. Links with particular relevance available at:

http://www.gamasutra.com/blogs/MarkFilipowich/20140130/209706/Marrow_Mechanics_the_Strength_of_Simple_Games.php

http://gamasutra.com/blogs/JohnKrajewski/20140114/208612/You_Have_Died_of_Dysentery_How_Games_Will_Revolutionize_Education.php

http://gamasutra.com/blogs/AlexandreMandryka/20140129/209620/Fun_and_uncertainty.php

http://www.gamasutra.com/view/feature/2190/the_designers_notebook_.php

Unity, online resource, used for everything related to the development of the game. Available at: *http://answers.unity3d.com/index.html*

Matematikk.net, online resource for mathematical curriculum, available at:
http://matematikk.net/side/Hovedside

Akison, A., Permutations, Combinations, and Variations using C# Generics, available online at: http://www.codeproject.com/Articles/26050/Permutations-Combinations-and-Variations-using-C-G
(retrieved, may 2014, used to create the probability task mechanic)

Math.NET Project, plugin for numerical computations, used in calculating binomial coefficients, available online at: http://www.mathdotnet.com/

CREDITS:

Trond Are Øritsland, Tutor for the duration of the project

Bjørn Baggerud, Master's Coordinator

All the playtesters for the game ELEMENTICS and all its prior versions, and the Indie development communities, without which this game would not be possible (reddit.com/r/Unity, r/Unity2D, r/Gamedev, r/Gamedesign)

LINKS:

Linked below are previous versions of the game, available for comparison to the final version:

https://googledrive.com/host/0BzsWrA66m6rvX0ZISmxSbW83ak0/QuickTest.html

https://googledrive.com/host/0BzsWrA66m6rvX0ZISmxSbW83ak0/QuickTest2.html

https://googledrive.com/host/0BzsWrA66m6rvX0ZISmxSbW83ak0/QuickTest3.html

https://googledrive.com/host/0BzsWrA66m6rvX0ZISmxSbW83ak0/webtest1.html

Notes: Prototype

Since the writing of this report, the prototype has underwent several minor (and some more impactful) changes. Below is a list of these changes.

CHANGELOG

- Tutorial, language and layout fixes. Tried to go for a simpler language and more to the point than a "descriptive tutorial".

- "Pull Up" mechanic to start game, fixed script and animation so it works better.

- GUI Tweaks, improved consistency of layout, simplified language especially for feedback given to the player. Changed Resonance to "Key Task" for consistency.

- Glitches and bugs, fixed minor bugs causing the elements not to reset properly from transitioning to the build-phase.

- Supplies capped at 18, to avoid the Build phase feeling overlong and cluttering the grid.

- Sounds and audio implemented, using freely available resources with a free-to-use license.

- Implemented a stricter control on generating new Matrix tasks. Ensuring a gradual increase in difficulty and allows the player to consistently "use-up" his or her groups before the Solve phase ends.

- New Task and New Key abilities are now available for repeated use at the cost of energy. This allows players who feel too annoyed at "unsolvable" tasks to go past these without having to exhaust their groups.

FINAL PROTOTYPE

In addition to this report, the final version of the game is available as a .apk file (for installation on a suitable Android platform). This is the version that delivers the intended experience, as touch controls make the GUI more intuitive and explorable.

The web version, with all these changes and newly implemented features is available at:

goo.gl/Sh4W3C
(https://9c24c76a96af35456029c6fa95b8a531bddfca87.googledrive.com/host/0BzsWrA66m6rvX0ZISmxSbW83ak0/FinalWebVersion02.html)

Please note that the game requires the installation of the Unity Web-player, and takes some time loading the first time especially.

The old version, which was subject to the most user tests (resulting in this new updated version) is available at:

http://goo.gl/3hjHJx
(https://9c24c76a96af35456029c6fa95b8a531bddfca87.googledrive.com/host/0BzsWrA66m6rvX0ZISmxSbW83ak0/Finalizer02.html)

APPENDIX

Project Text

NTNU
Norges teknisk-naturvitenskapelige universitet

Fakultet for ingeniørvitenskap og teknologi
Institutt for produktdesign

Masteroppgave for student Mikkel Blytt

Utvikling av digitalt spill for økt forståelse av matematiske konsept

Development of digital game for improved understanding of mathematical concepts

Bruken av spill som et verktøy i læringsprosessen er ikke et nytt fenomen. Spill kan motivere og vekke interesse for å lære om ulike tema og konsept, men også bidra til en økt forståelse av disse. Utstrakt bruk av nettbrett og smarttelefoner, allestedsnærværende databehandling (ubiquitous computing) har gjort tilgangen til slike spill mye enklere via for eksempel "App Store" og "Google Play".

Med bakgrunn i personlig interesse for spill, spillutvikling og matematikk vil jeg utvikle et digitalt spill som har til hensikt å gi spillerne en økt intuitiv forståelse av forskjellige matematiske konsept på et ungdomsskolenivå. Det følger av dette at spillkonseptet skal være allment tilgjengelig og morsomt å spille for flere aldersgrupper, og ikke være et rent undervisningsverktøy, men heller et supplement. Utformingen av spillet skal derfor gjøres ved hjelp av teori om læring og undervisning, men også ut fra teori om spilldesign. For å komme frem til best mulig løsning skal det følges en interaksjonsdesign-prosess med sterkt fokus på brukerinvolvering.

Oppgaven vil innebære å undersøke teori for spilldesign og læring, og krever hyppig kontakt med målgruppen forut for og under utviklingen av spillkonseptet. Ut ifra dette skal det utformes et digitalt spill som er "scope complete", men ikke optimalisert fra et programmeringsmessig perspektiv.

Oppgaven vil blant annet omfatte:

- Informasjonsinnhenting og teori
- Papir og digital prototyping
- Omfattende brukertesting
- Test av endelig løsning på målgruppe
- Endelig spillbeskrivelse knyttet til teori
- Presentasjon og endelig prototype

Oppgaven utføres etter "Retningslinjer for masteroppgaver i Industriell design".

Ansvarlig faglærer: Trond Are Øritsland

Utleveringsdato: 17. januar 2014
Innleveringsfrist: 12. juni 2014

Trondheim, NTNU, 17. januar 2014

Trond Are Øritsland
ansvarlig faglærer

Casper Boks
instituttleder

Curriculum

The untranslated middle-school curriculum, as taken directly from Matematikknet.no (see Additional Resources) is shown here. This shows which areas that were considered for game mechanics and what the final game is based on.

LÆREPLAN UNGDOMSTRINNET

Tall og algebra

Mål for opplæringen er at eleven skal kunne:
- sammenligne og regne om heltall, desimaltall, brøker, prosent, promille og tall på standardform og uttrykke slike tall på varierte måter
- regne med brøk og utføre divisjon av brøker samt forenkling av brøkuttrykk
- bruke faktorer, potenser, kvadratrøtter og primtall i beregninger
- utvikle, bruke og gjøre rede for metoder ved hoderegning, overslagsregning ogskriftlig regning tilknyttet de fire regneartene
- behandle og faktorisere enkle algebraiske uttrykk, regne med formler, parenteser og brøkuttrykk med ett ledd i nevner
- løse likninger og ulikheter av første grad og enkle likningssystemer med to ukjente
- sette opp enkle budsjetter og gjøre beregninger tilknyttet privatøkonomi
- bruke, med og uten digitale hjelpemidler, tall og variabler i utforskning, eksperimentering, praktisk og teoretisk problemløsning og i prosjekter med teknologi og design

Geometri

Mål for opplæringen er at eleven skal kunne:
- analysere, også digitalt, egenskaper ved to- og tredimensjonale figurer og anvende disse i forbindelse med konstruksjoner og beregninger
- utføre og begrunne geometriske konstruksjoner og avbildninger med passer og linjal og andre hjelpemidler
- bruke formlikhet og Pytagoras' setning i beregning av ukjente størrelser
- tolke og lage arbeidstegninger og perspektivtegninger med flere forsvinningspunkter ved hjelp av ulike hjelpemidler
- bruke koordinater til å avbilde figurer og til å finne egenskaper ved geometriske former
- utforske, eksperimentere med og formulere logiske resonnementer ved hjelp av geometriske ideer og gjøre rede for geometriske forhold av særlig betydning innenfor teknologi, kunst og arkitektur

Måling

Mål for opplæringen er at eleven skal kunne:
- anslå og beregne lengde, omkrets, vinkel, areal, overflate, volum og tid, og kunne bruke og endre målestokk
- velge passende måleenheter, forklare sammenhenger og regne om mellom ulikemåleenheter, bruke og vurdere måleinstrumenter og målemetoder i praktisk måling, og drøfte presisjon og måleusikkerhet
- gjøre rede for tallet pi og bruke dette i beregninger av omkrets, areal og volum

Funksjoner

Mål for opplæringen er at eleven skal kunne:
- lage, på papiret og digitalt, funksjoner som beskriver numeriske sammenhenger og praktiske situasjoner, tolke disse og oversette mellom ulike representasjoner av funksjoner som grafer, tabeller, formler og tekst
- identifisere og utnytte egenskapene til proporsjonale, omvendt proporsjonale, lineære og enkle kvadratiske funksjoner og gi eksempler på disse funksjonenes tilknytning til praktiske situasjoner

Statistikk, sannsynlighet og kombinatorikk

Mål for opplæringen er at eleven skal kunne:
- gjennomføre undersøkelser og bruke ulike databaser til å søke etter og analyserestatistiske data og utvise kildekritikk
- ordne og gruppere data, finne og drøfte median, typetall, gjennomsnitt og variasjonsbredde, og presentere data med og uten digitale verktøy
- bestemme sannsynligheter gjennom eksperimentering, simulering og beregning i dagligdagse sammenhenger og spill
- beskrive utfallsrom og uttrykke sannsynligheter som brøk, prosent og desimaltall
- vise med eksempler og bestemme antall muligheter i enkle kombinatoriske problemer

Questionnaire

At the start of the project, a quick informal questionnaire was conducted to see which mathematical topics people considered the most difficult at a middle-grade level. This questionnaire was conducted both in the form of informal interviews and over the internet. The form is in Norwegian, since this was the language of all participants.

QUESTIONS

- Hvilken del av matematikkpensumet syntes du var vanskeligst å forstå på ungdomsskolen?

- Hvilken del av matematikkpensumet syntes du var vanskeligst å utføre oppgaver for på ungdomsskolen?

- Hvilken del av matematikkpensumet syntes du var vanskeligst å forstå i første klasse på videregående?

- Hvilken del av matematikkpensumet syntes du var vanskeligst å utføre oppgaver for i første klasse på videregående?

- Var du glad i matematikk på ungdomsskolen?

- Var du glad i matematikk på videregående?

RESULTS

A total of 18 people answered this informal study.

Distribution for question 1:
Probability - 9
Algebra - 5
Geometry - 2
Functions - 2

Distribution for question 2:
Probability - 10
Algebra - 3
Geometry - 3
Functions - 2

Distribution for question 3:
Probability - 5
Algebra - 3
Geometry - 1
Functions - 4
Not applicable - 5

Distribution for question 4:
Probability - 7
Algebra - 2
Geometry - 1
Functions - 3
Not applicable - 5

Distribution for question 5:
Yes - 9
No - 4
Undecided - 5

Distribution for question 3:
Yes - 7
No - 4
Undecided - 2
Not Applicable - 5

Playtest form

What follows is the form used when playtesting, the first part are the guidelines I set for myself, and the second part deals with questions for the semi-structured interview and a Likert scale type form that I used to get a measure of quantifiable feedback.

In some cases this form was the only feedback I received, but in most of the cases I also observed the user during play and was able to end each session with an additional conversation about the game and gameplay.

Observation was done with me observing both in person and through shared-screens over the internet, depending on the prototype used.

GUIDELINE

1. Introduce myself and the fact that this is a voluntary process, the user is free to leave at any moment.

2. OPTIONAL STEP: Quickly show and describe the game and the most important game mechanics so that the user has some idea of what to expect.

3. Express the limitations of the prototype, but the fact that feedback for the most part should treat the game as if it was a finalised product (NOTE: not for the earliest paper prototypes or digital single-feature tests).

4. Express the intent of the playtest, which is to test functionality and observing the players interaction and engagement with the game and gameplay. Note that the users are not the ones being "tested", and that it is impossible for them to fail, any fault that arises is to be attributed to either me or the game.

5. Express the desire for the user to also answer a few questions and fill out a form after the playtest. Remind the user that all data will be anonymised and not shared or used beyond the purpose of improving future versions of the game.

6. During testing, remain unobtrusive, only step in when the user is on the verge of giving up from being stuck over a longer period of time. Allow the user to note questions and comments while playing.
7. After the playtest and the questions are answered, ask if the user would like to be contacted for future playtests later (or potentially a second round of playing the game if appropriate, to test on-the-fly changes).

QUESTIONS

The following questionnaire was given to all users, note that the questions were given in Norwegian.

Er du glad i matematikk?
Hvor godt gjorde du det i matematikk på ungdomsskolen?
Hva tenker du om spillopplevelsen?
Hva føler du spillet er bygget på?
Hva likte du best med spillet?
Hva likte du dårligst med spillet?
Har du andre forslag eller innspill?

Also the user was always welcome to remark or comment outside the scope of these questions.

FORM

The following scale was also answered after most usertests. Users answered the following statements on a scale from one to five, where five meant agreement and one meant disagreement with the statement. Three indicated a neutral stance.

The game was fun
The game was interesting
The game was repetitive
The game felt finished
The game was lacking
The game was demanding
The game needed prior knowledge
The game felt educational
The game has potential
The game is best as a digital game
The game is best as an analogue game

The aim was to cover both positive and negative aspects, to avoid getting biased feedback.

Screendumps

Shown here are pictures taken from the Unity workspace from the early to the later stages of the game development. As can be seen this reveals major alterations both in terms of functional layout and aesthetic profile. The final aesthetic profile attempts to be a neutral, semi-abstract representation of a science fiction like setting.