NTNU
Norwegian University of
Science and Technology

Kickstarting an indie game development studio

Alexander Jonassen
Christer Rebni

Industrial Design Engineering
Submission date: June 2018
Supervisor: Trond Are Øritsland, ID

Norwegian University of Science and Technology
Department of Design

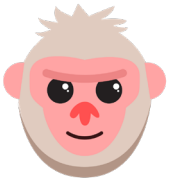
READY!

UTFORSKENDE SPILLUTVIKLING

På leting etter kommersialiserbare dataspillkonsepter ved hjelp av rask prototyping

INNHOLD

INTRODUKSJON



INNLEDNING

FORORD

Denne masteroppgaven presenterer resultater og refleksjoner etter et halvårig spillutviklingsprosjekt utført ved Institutt for Design ved NTNU, høstsemesteret 2018. Arbeidet som ble gjort i løpet av prosjektet har vært ment som en øvelse i alle aspekter ved spillutvikling og en mulighet for oss å skape potensielt kommersialiserbare spillkonsepter for tiden etter etter endt master. Det endelige målet har vært å opprette et spillstudio.

Vi føler oss meget priviligerte som har fått mulighet til ikke bare å jobbe med både spilldesign og spillutvikling, men også starte egen bedrift som en del av masteroppgaven. Hadde noen fortalt oss hva vi kom til å drive med om ti år for ti år siden hadde vi neppe trodd på det. Vi takker vår veileder, Trond-Are Øritsland, som har støttet og inspirert oss til å tørre å satse på spillutvikling som en karriere, Jonas Hyllseth Ryen fra Spark* NTNU, Norwegian Game Awards, communityet ved WorkWork i og alle som har vært med og testet, gitt verdifull tilbakemelding og pushet kvaliteten på spillene våre. Den største takken går til alle våre hvener ved Institutt for Design på NTNU, verdens beste industridesignstudenter, som alltid motiverer og eleverer humøret - en gjeng ekte players.

OPPGAVETEKST

NTNU
Fakultet for arkitektur og design
Institutt for design

Master thesis for Alexander Jonassen & Christer Rebni

Kickstarting an indie game development studio
Snar oppstart av et uavhengig spillstudio

The goal for this project is to experiment with video-game development and gain as much experience with game design as possible, the process from idea to publication and finally marketing. The purpose is to examine the feasibility of founding an indie video-game studio.

During the course of the project we will develop 7 individual games which will be published continuously on a dedicated itch.io page. Additionally, the project will be distributed on social media with the express purpose of establishing an online presence and building an audience. Each game is allotted two weeks, wherein we will be doing idea-generation, development, distribution and marketing. The project follows an open development-mindset, which means: Sharing the projects early and establishing a dialog with the audience.

The final delivery will, amongst other things: Present each game, present the feedback we received from the audience, and our own reflections.

This project will entail, amongst other things:

- Idea-generation and development of digital video-games/game-prototypes
- Publishing the developed games
- Gather feedback from players, in order to gauge the feasibility of founding an indie video-game studio, as well as using the feedback to grow as game designers.

The project is executed in accordance with "Retningslinjer for masteroppgaver i Industriell design".

Course supervisor: Jóhannes Blöndal Sigurjónsson
Department supervisor: Trond Are Øritsland

Starting date: January 12th 2018
Due date: June 7th 2018

Trondheim, NTNU, January 10th 2018

Trond Are Øritsland
Department supervisor

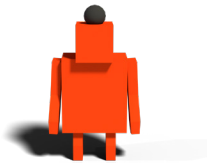
Ole Andreas Alsos
Head of Department

SAMMENDRAG

Målet med denne masteroppgaven var å anvende arbeidsmetodene vi har lært i løpet av studietiden vår for å lage spill, for så å opprette et levedyktig spillstudio basert på inntjening fra disse spillene.

Terskelen for å stifte et indie-spillstudio har de siste årene sunket betraktelig, og i dagens indie-spillmarked konkurrerer man mer på kreativt innhold og spilldesign, heller enn på teknisk kompetanse. På bakgrunn av dette mener vi at vår designutdannelse gir oss en god sjanse for å kunne skille oss ut i et indiemiljø dominert av spillutviklere med bakgrunn innen programmering.

I en iterativ prosess basert på sykluser kjent fra lean startup har vi utarbeidet fire spillkonsepter, i varierende grad av ferdigstillelse, samt jobbet videre på et spillkonsept som ble påbegynt i forkant av masteren. Av disse har to spill blitt publisert, to endt opp som spillbare prototyper, og det siste nærmer seg utgivelse.



ABSTRACT

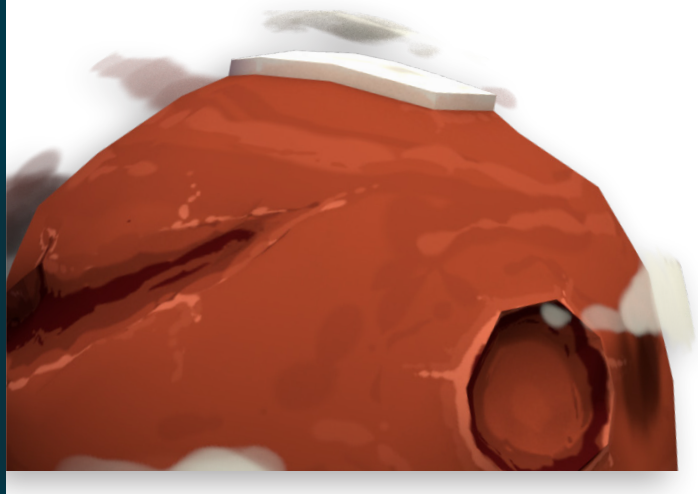
The goal of this master thesis project was to utilize the design methods and strategies we have learned to create video games, and, based on the income these games would generate, found an independent video games studio.

The obstacles and pitfalls encountered by newly founded indie-game studios have changed considerably in recent years. Earlier pioneers often faced technical hurdles, however the proliferation of affordable and powerful out-of-the-box game engines have created a market wherein creativity and novel game design, rather than technical brilliance, often is the kingmaker. Based on this we believe our design education gives us a fair chance to make a name for ourselves in an industry saturated by programmers.

Following an iterative process, based on the familiar lean startup cycle, we have created four game concepts, each currently in various states of completion, in addition to continuing work on a project started before the master thesis. Two of the games have been published, two exists as playable prototypes and the final one is nearing publishing.

HENSIKT

Hensikten med denne masteroppgaven var å utvikle våre evner som spillutviklere gjennom mengdetrening, med et endelig mål om å opprette et bærekraftig spillstudio ved endt studie. Vi ville undersøke om vårt lille team på to personer var i stand til å produsere underholdende spill i løpet av korte sprinter på to uker. Denne fremgangsmåten er delvis inspirert av en metode for rask spillutvikling fra utviklerne av bl.a. World of Goo (Kyle et.al., 2005), men det er også en fremgangsmåte som faller seg naturlig for oss som designere, da vi er godt vant med å jobbe raskt og iterativt. Det er dog begrenset hva to personer kan utrette på to uker, så å lære seg å avgrense prosjekter med hensyn på arbeidskapasitet og tid ville også være viktig. Ambisjonen var at to uker skulle være tilstrekkelig til idegenerering, design, implementasjon, markedsføring og publisering.



ORDFORKLARINGER

AI
Artificial Intelligence, ofte kalt kunstig intelligens på norsk, brukes innen spilldesign for å snakke om (ofte enkle) datastyrte systemer som utgir seg for å ha egen vilje.

GAME LOOP
Strukturen i spillets tilstander. Et spill kan ha flere ulike tilstander (meny, lobby, gameplay, game over osv.). En god game loop krever at det er en logisk struktur og en god rytme i overgangene mellom disse tilstandene.

GIF
Bildeformat som støtter animert grafikk.

LOBBY
Område eller meny i et flerspillerspill hvor spillere venter på at andre spillere melder seg opp til å spille.

PID-REGULATOR
PID-regulatorer er mye brukt i prosessindustri og robotikk. Hensikten er å holde en verdi på et stabilt nivå ved å bruke avviket mellom målt verdi og ønsket verdi som grunnlag for et effektpådrag.

REDDIT
Et forum på internett delt inn i underforum (subreddits) for et vell av ulike temaer og interesseområder.

SUBREDDIT
Se Reddit.

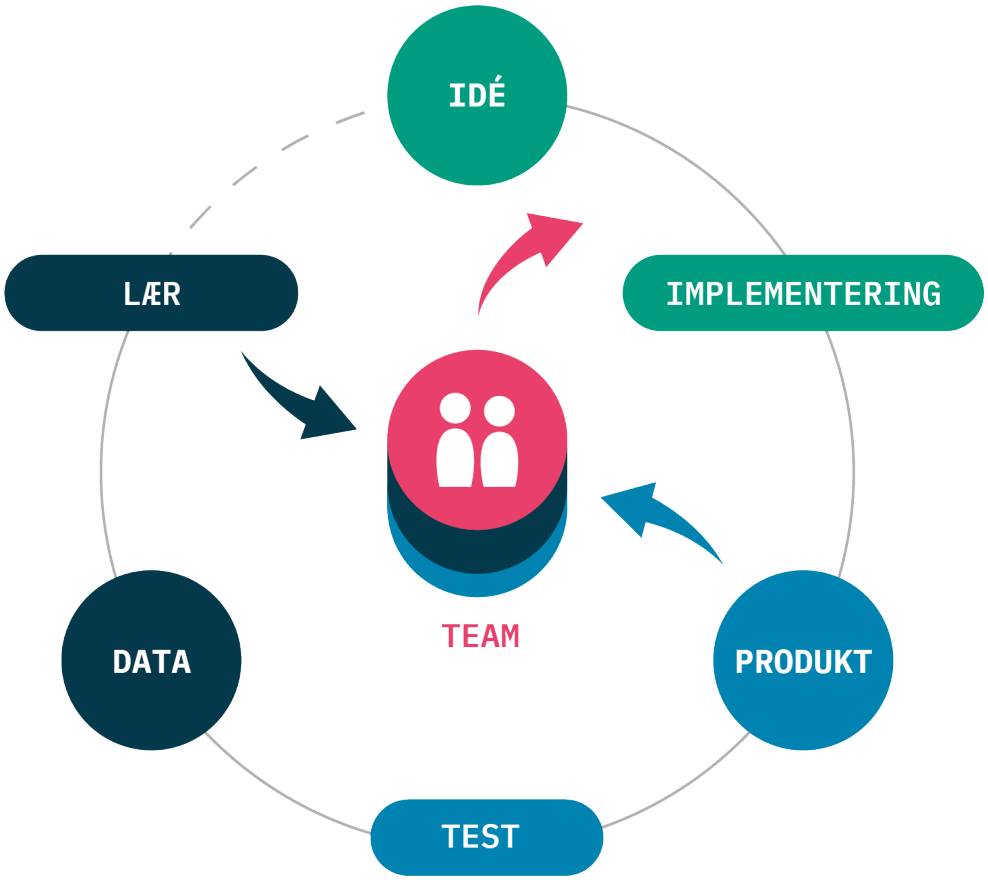
UPVOTE & DOWNVOTE
Måte å stemme opp eller ned innlegg og kommentarer på Reddit. Poster med mange upvotes blir mer synliggjort på forumet.

METODE

METODE

Metodene som har blitt benyttet i løpet av dette prosjektet spenner fra klassiske designfaglige aktiviteter, som idégenereing, brukertesting og iterativ prototyping, til mer formelt tekniske arbeidsoppgaver, som koding i C#, opprettelse av utviklerkontoer og enkeltpersonsforetak, til kreativt arbeid som musikkproduksjon, tegning og 3D-modellering. Siden vi bare er to stykker på teamet, og spillutvikling er et så allsidig virke, har vi vært nødt til å spille mange forskjellige instrumenter for å komme i mål.

Den overordnede prosessen har tatt form som en klassisk lean startup-prosess, hvor vi starter med en idé, implementerer den, før vi så samler data om denne for å kunne iterere videre på den samme ideen igjen (Ries, 2011). Forskjellen har vært at vi har tatt lærdommene våre og videreført de inn i nye prosjekter, i stedet for å fortsette på et prosjekt hele tiden. Dette gjorde vi for å raskt øke størrelsen på kunnskapsbasen vår, ettersom målet med masteroppgaven også var å lære praktiske ferdigheter rundt spillutvikling



Overordnet prosessløkke

Vi har utviklet spill, for å tilegne oss erfaring til å utvikle nye og bedre spill, samt generere prototyper vi kan jobbe videre med i fremtiden.

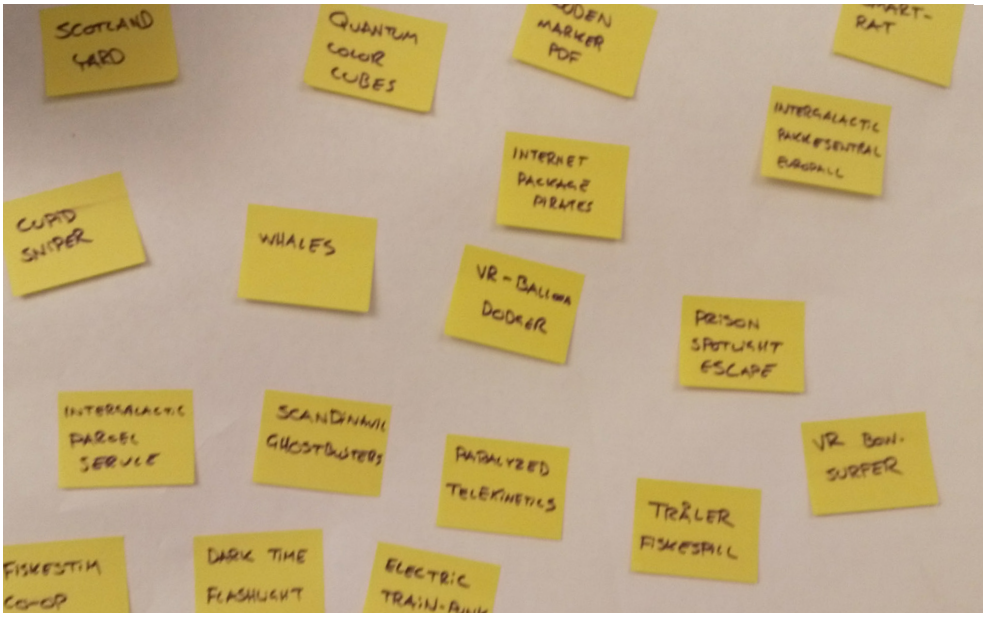
DESIGN

INNSIKTSFASE

Innledningsvis i dette masterprosjektet så vi til andre aktører i spillbransjen og hvordan de har brukt konsepter som *open production* (Kidwell, 2018), og rask prototyping (Kyle et.al., 2005) for å komme frem til kommersielt gyldige spillkonsepter.

Open production går ut på å inkludere brukere i utviklingsprosessen ved å dele tidlige prototyper og designforslag, og få feedback på disse gjennom åpen dialog. På denne måten bygges samtidig en fanbase av potensielle kjøpere. Kyle et.al. hadde en prosess hvor de løpet av enukerssykluser valgte ut tema før de delte seg opp og lagde hver sin prototype. Mot slutten av hver uke samlet de seg og kombinerte de beste delene fra alle konseptene. På den måten kom de opp med varierte konsepter, alle fikk utløp for sin kreativitet og godt eierskap til det endelige konseptet.

Ryan Clark, grunnlegger av og spilldesigner i *Brace Yourself Games*, anbefaler at man fra begynnelsen designer spillet med tanken om



Idégenerering

Et knippe stikkordsformulerte spillideer fra en av våre første idéworkshops

at det må ha *hooks* (Clark, 2018). Hooks i denne sammenhengen er elementer i spillet som har potensiale til å kjapt fange interessen til publikum idet det presenteres i en trailer, i et screenshot eller i en beskrivelse av spillet. Det kan være alt fra en morsom karakter eller en særegen visuell stil til en original spillmekanikk eller å ha en kjent skuespiller er fortellerstemme.

IDÉGENERERING

For vår del innledet vi hver toukerssyklus med å velge ut en idé til et spill. Vi valgte ideer delvis ut fra egendefinerte læringsmål relevante for spillutvikling. F.eks. ønsket vi på ulike tidspunkt å lære oss publisering til Android, nettverkskommunikasjon og "*Jump in-jump out*"-funksjonalitet for flerspillerspill på samme skjerm. Vi valgte også ideer ut fra hva vi syntes hørtes gøy ut og hvorvidt vi trodde dette var teknisk gjennomførbart for oss og spillmotoren vi hadde valgt å bruke, Unity3D. Det ble bevisst ikke brukt mye tid på å velge ut ideer. Vi anerkjente at et konsept uansett måtte testes for å valideres, og med bare to uker til rådighet hadde vi ikke tid til å lete lenge etter en tilsynelatende genial idé som vi risikerte å måtte kaste så snart vi fikk testet den. Da antok vi heller at iterativ utvikling kunne redde en i utgangspunktet middelmådig idé. Denne tankegangen støttes i episoden Fail Faster på YouTube-kanalen Extra Credits av James Portnow (Portnow 2014).

SPILLDESIGN

Spilldesign er prosessen å avgjøre hva et spill skal være (Schell, 2014). Det er en kontinuerlig prosess og i bunnen ligger et mål om å gi spilleren en gitt opplevelse. Spilldesign innebærer alt fra valg av estetisk uttrykk og hvilke spillmekanikker som best underbygger opplevelsen man ønsker å fasilitere for, til valg av målgruppe og plattform spillet skal spilles på.

Designprosessen vår har vært av en utforskende og iterativ karakter. Gjennom idégenering og skissering av spillets grunnmekanikker har vi fått et utgangspunkt for implementering og testing. Konseptet har på dette stadiet kunnet blitt oppfattet som løst og uferdig, men vi sa oss fornøyd med denne løse retningen fordi vi antok at konseptet uansett ville endre seg etter hvert som vi fikk opp testbare prototyper av det vi har tenkt.

I alle prosjektene våre har spilldesign og implementasjon gått ganske parallelt. Vi har kommet opp med ideer på hva vi tror er gode mekanikker og aktiviteter, for så å implementere disse, teste og validere dem. Dermed har avgjørelser blitt tatt fortløpende etter hvert som spillet har utviklet seg.

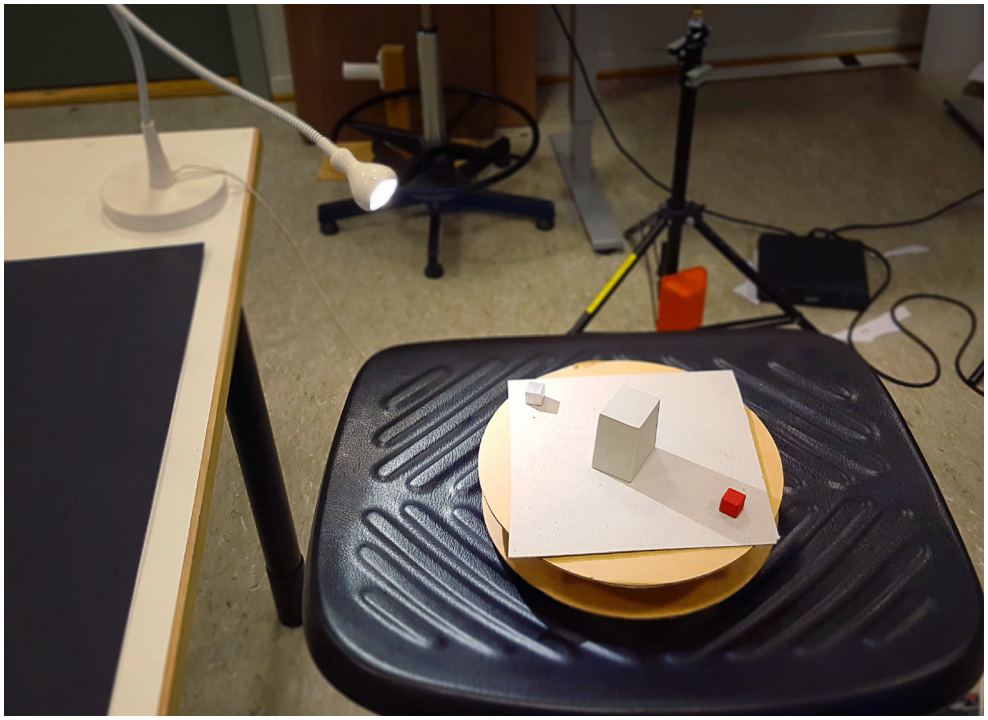
PROOF-OF-PRINCIPLE PROTOTYPING

I løpet av den tidlige utviklingsprosessen har enkelte ideer blitt skissert og testet før en digital *minimum viable product*-prototype (MVP) i Unity3D (heretter Unity) ble påbegynt. Forutenom skisser av skjermbilder og logikk ble det f.eks. testet hvordan lys og skygge oppførte seg på en roterende rigg, i forbindelse med utviklingen av spillet You Have To Tell Me If You’re Human (Y.H.T.T.M.I.Y.H.), og på et tidspunkt lekte vi stein-saks-papir-sisten i gangene på instituttet hvor vi sitter for å sjekke om grunnpremisset for spillet Super Rock Paper Scissors var gøy. Dette gikk fort, og ga oss en rask pekepinn på hvorvidt det vi hadde tenkt ut var verdt å gå videre med.

DIGITAL PROTOTYPING

I hvert prosjekt fikk vi opp raske MVP-er av gameplayet vi så for oss, for så å fortløpende vurdere hvorvidt aktivitetene i spillet kjentes riktige ut ved å teste spillet selv. Dermed kunne vi vurdere hvorvidt vi ønsket å videreutvikle eller trengte å ta konseptet i en annen retning.

Vi benyttet spillmotoren Unity både for protoyping og utvikling. Gjennom Unitys samarbeidsfunksjon hadde vi anledning til å jobbe samtidig på prosjektet. Vi fordelte arbeid mellom oss slik at vi ikke jobbet på samme del av spillet samtidig, da det å jobbe samtidig på ting lett førte til tekniske konflikter.



Proof of Principle-prototype i forbindelse med utviklingen av spillet Y.H.T.T.M.I.Y.H.

En test for å se om det oppstår noe gøy når lysv står i ro mens banen kan roteres.

IMPLEMENTERING

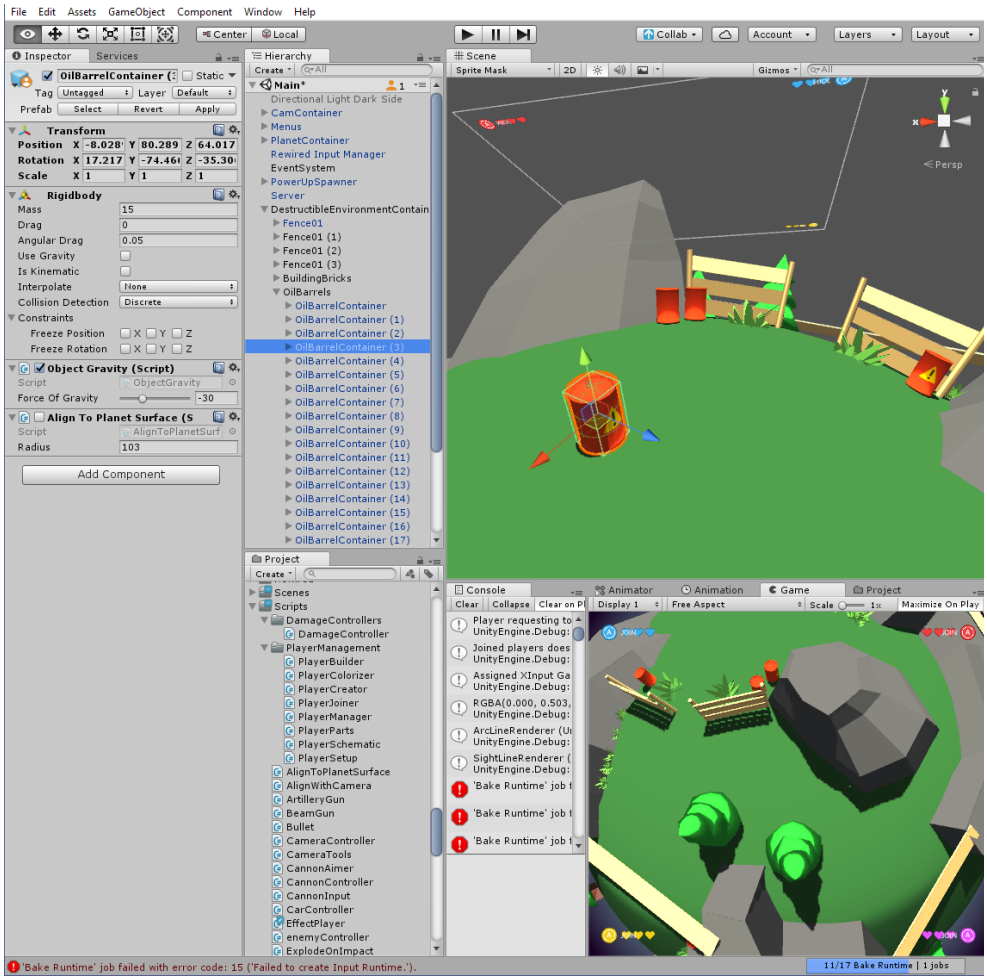
UNITY3D

Spillutvikling er et tverrfaglig prosjekt. Selve implementeringen har for vår del foregått i Unity3D (heretter Unity), et program hvor alt av *assets*, det vil si alt av 2D- og 3D-modeller, teksturer, animasjoner, visuelle effekter, musikk, lydeffekter, skript og så videre sys sammen til en fungerende og helhetlig spillopplevelse.

Unity er en spillmotor som er gratis å ta i bruk. Spill produsert i Unity kan publiseres kommersielt, og Unity krever ikke annet enn gratislisens før omsetningen i spillforetaket overstiger 100 000 USD. Som utvikler får man mye gratis ved å bruke en spillmotor som Unity, sammenlignet med hvis man måtte ha implementert teknologien selv. F.eks. tar Unity seg av funksjoner som rendring av grafikk til skjerm, mange mekaniske fysikkberegninger, lyssetting, og har dessuten mulighet til å eksportere spillet til et stort utvalg plattformer, for å nevne noe. Unity har også en innebygd samarbeidsfunksjon, som gjør at utviklere på teamet kan jobbe på prosjekter samtidig, uten å måtte forholde seg til mer teknisk avanserte løsninger som f.eks. GitHub. Vi erfarte at Unity's samarbeidsfunksjon gjorde akkurat det vi hadde behov for, så vi valgte å ikke å bruke GitHub. Alt dette gjør en spillmotor som Unity svært attraktivt for uavhengige spillstudio som vil bruke mer energi på kreativt arbeid enn teknisk tung implementering av helt grunnleggende funksjoner.

Det finnes alternativer til Unity, som f.eks. Unreal Engine og GameMaker, men siden vi allerede hadde erfaring med Unity valgte vi å bruke denne.

Unity er bare ett av mange verktøy som ble brukt under utviklingen. Unity er brettet hvor alle puslespillbitene settes sammen, mens programmer som f.eks. Adobe Illustrator, Blender, og LMMS brukes til å lage nødvendige assets (her henholdsvis 2D-grafikk, 3D-modeller og musikk). For et team på to stykker krever dette at vi setter oss inn en rekke ulike programmer, for så å produsere innhold som passer til spillet vi jobber på.



Skjermdump fra Unity

Arbeidsområde i Unity. Her under utvikling av spillkonseptet Tank Planet.

Mye tid går med til disse jobbene. Animasjoner, 3D-modellering, fleksibel kode og fine lys- og partikkeleffekter er typiske tidstyver. Disse typene assets tar ofte tid å få på et nivå som ser og føles bra ut (og erfaringene våre tilsier at det også er lett å *ta* seg tid til dette). Her bør man altså være sikker på at den jobben man legger ned ikke forkastes ved neste iterasjon.

Siden spillene har blitt utviklet iterativt betyr det at den siste versjonen av alle spill er bygd på toppen av gamle versjoner. Nødvendige spesifikasjoner har blitt avdekket fortløpende etter hvert som spillet har tatt form, fremfor å bli spesifisert i detalj på forhånd.

PROGRAMMERING

All logikken i spillene er programmert i C#. Dette er et kodespråk Unity legger opp til at utviklere skal bruke, og det finnes store mengder dokumentasjon, hjelp og veiledninger på internett. Ressurser som Unity Answers [1], Unitys kodedokumentasjon[2] og guider fra ulike YouTube-kanaler har vært uvurderlige i arbeidet. Kort sagt: Løsningene er sjelden mer enn et websøk unna.

Ved produksjon av en MVP er det anbefalt å skrive koden så raskt og enkelt som mulig. Skulle MVPen vise seg å være en suksess bør en være forsiktig med å fortsette utviklingen uten å enten revidere gammel kode eller rett og slett starte fra scratch. Vi har gjentatte ganger måttet bruke tid på å rydde opp i gamle skript eller organiseringen av spillobjekter fordi de har vært bygd opp på en lite hensiktsmessig måte.

Etterhvert som spillene tok form økte også kompleksiteten på programmeringen. Veldig mye mer tid har gått med til å kode spillmekanikkene vi designer sammenlignet med tiden det tok å faktisk komme opp med mekanikkene. Det er også en nivåforskjell mellom kode som er god nok, og kode som skaper en god brukeropplevelse, der sistnevnte ofte krever en del ekstra arbeid. Dette har gjort seg særlig gjeldende i programmering av tilstandsmaskiner (klasser som kan være i ulike tilstander, og skal gjøre ulike ting basert på hvilken tilstand de er i, f.eks. en menykontroller eller rundekontroller). For eksempel krever det ganske mye mer mer å gå fra en skjerm til en annen med en behagelig animasjon og lydeffekt, enn å bare skifte direkte til den.

---

1 answers.unity.com

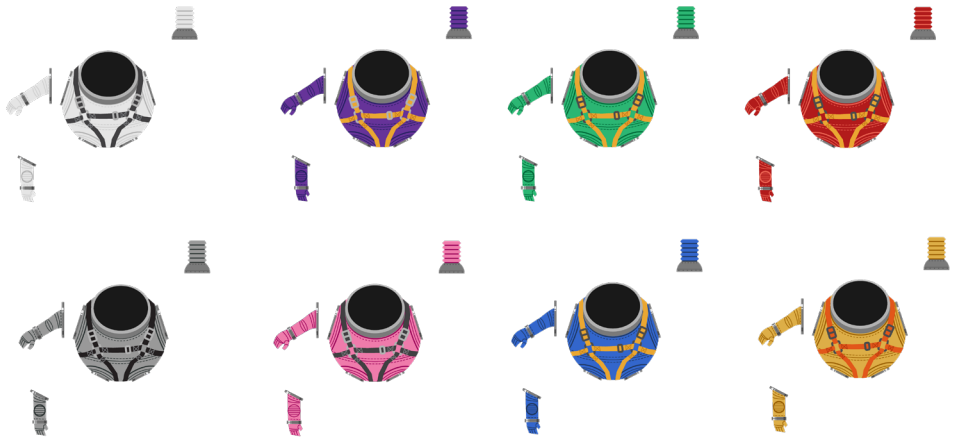
2 docs.unity3d.com

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using Rewired;

public class CannonAimer : MonoBehaviour {

    [Header("Aim Behaviour")]
    public float aimSpeed;
    public float stickThreshold;
    private float stickWalk;
    private Vector2 stickDirection;

    private float targetAngle = 0f;

    [Header("Moving parts")]
    public Transform carTransform;
    public Transform cannonAimContainer;
    public Transform carCorrectionContainer;
    public Transform CannonTower;

    [Header("Input")]
    public PlayerInput playerInput;


    // Update is called once per frame
    void Update () {
        stickDirection = GetStickRotation ();
        RotateTarget ();
        CarDeltaCorrection ();
    }

    private Vector2 GetStickRotation(){
        float h = playerInput.horizontalAim;
        float v = playerInput.verticalAim;

        if (Mathf.Abs(h) > stickThreshold || Mathf.Abs(v) > stickThreshold) {
            Vector2 vector = new Vector2 (-h, v);
            stickWalk = vector.magnitude;
            return vector;
        }
        else
            return Vector2.zero;
    }

    private void RotateTarget(){
        if (stickDirection != Vector2.zero) {
            //Get aim angle on joystick
            float aimAngle = Vector2.SignedAngle (Vector2.up, stickDirection);

            //Calulate target angle relative to local transform
            Vector3 newRot = cannonAimContainer.localRotation.eulerAngles;
            float deltaAngle = Mathf.DeltaAngle (newRot.y, aimAngle);

            //Save target angle
            targetAngle = newRot.y + deltaAngle;
```

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class LevelBlueprint : MonoBehaviour {

    public string name;
    public GameObject levelKit;
    public Vector2 timeLimit;
    public Vector3 scoreTargets;
    public int levelsToUnlock = 1;
    public Sprite solutionImage;

    [Header("Optional")]
    public GameObject[] gameObjects;
    //public Hazards[] hazards;

}
```

Skjermdumer fra MonoDevelop

Kodesnutter fra spillene Tank Planet (øverst) og Bonkies.



Draktfarger i Bonkies
Sprite-sheet med draktdeler i ikoniske farger

BRUKBARHET

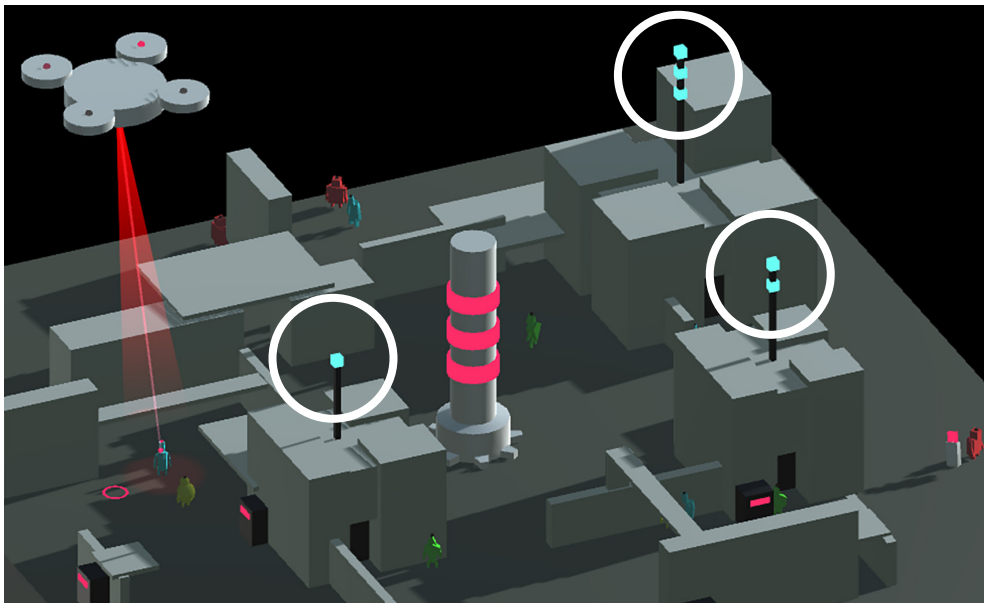
I design av brukergrensesnitt for spill forsøker en gjerne å treffe en balanse mellom tydelig visualisering av informasjon og et estetisk uttrykk som støtter opp under den overordnede spillopplevelsen. Spill trenger ikke ta like store hensyn til brukbar- og lesbarhet som ved design av grensesnitt i ikke-spillapplikasjoner, og kan med god samvittighet ha en lav data/blekk-rate[1], dersom man kan argumentere for at blekket er med på å underbygge en gitt opplevelse. For eksempel vil tallinformasjon tegnet i gullbokstaver på en mørk trebakgrunn kunne passe bedre i et spill med pirat-tema, enn en enkel sort font på en lys bakgrunn.

Flere av spillene vi har laget er ment å bli spilt av flere spillere i samme rom. Når det i tillegg blir snakk om samarbeid må spillerne kunne skilles fra hverandre, og spillere må kunne adressere hverandre. Måten vi valgte å løse dette på i spillet Bonkies var ved fargekoding av spillerne. Draktfargene i Bonkies (se bildet over) følger anbefalt

1 Vanligvis er det ideelt å strebe etter en høy data/blekk-rate, altså å kommunisere så mye kvantitativ data med så lite grafikk (blekk) som mulig (Tufte, 2001)

fargekoding for å effektiv kunne navngis og skilles fra hverandre (Ware, 2013). Ved å få anledning til å velge farge selv i forkant av spillet vet spillerne lettere hvem de kontrollerer i det spillet begynner. Fargekodingen gjør at spillere forholdsvis enkelt kan adressere og kommandere hverandre ved å rope ut fargen på spilleren de ser på skjermen.

I Y.H.T.T.M.I.Y.H. måtte vi ta hensyn til at alle spillerne så den samme skjermen, samtidig som de andre spillerne skulle gjemme seg for den ene spilleren. Dronespilleren flyr over brettet, godt synlig (se bildet under), så alle vet hvor han er, men spillerne på bakken må vite hvor de er plassert ut ved spillets start uten at dronespilleren får vite det. Løsningen ble å kommunisere posisjonen gjennom vibrasjon i kontrollene. I den nåværende prototypen er det plassert ut tre nummererte hus, og spillere plasseres i et vilkårlig valgt hus ved start, og kontrollen til spilleren vibrerer forsiktig like mange ganger som nummeret på huset. Denne måten å overføre hemmelig informasjon på fungerte effektivt, selv om hus-mekanikken i seg selv trenger seg en runde innom tegnebrettet.



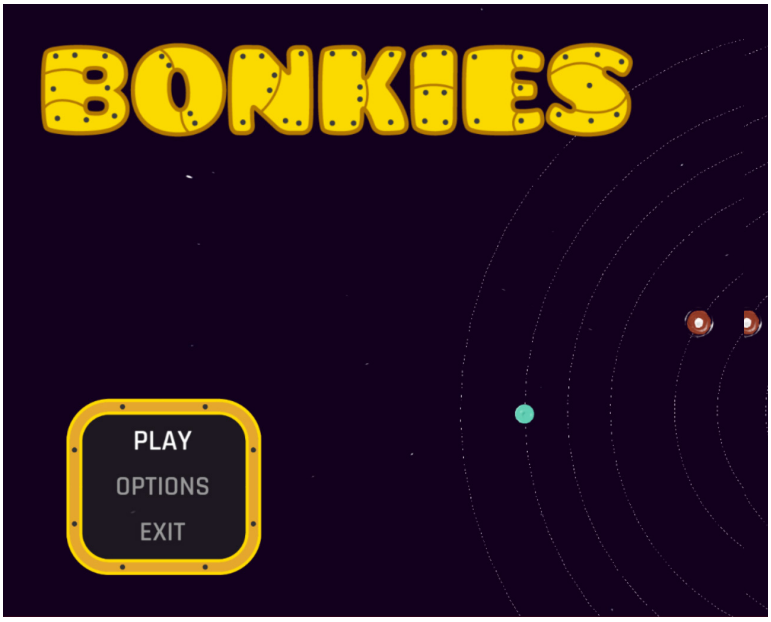
Skjermbilde fra Y.H.T.T.M.I.Y.H.

Spillerne kan starte i ett av de tre husene på brettet. Kontrollen vibrerer like mange ganger som det er antall lys på antennen på huset.

UI-DESIGN

Brukergrensesnittet i spill skal bidra både til å sikre gode tilbakemeldinger på spillerens handlinger i spillet og til å underbygge den overordnede opplevelsen spillet ønsker å fremkalle i spilleren. Brukergrensesnitt innbefatter her både menysystemer, infoskjermer og grafikk som vises mens man spiller.

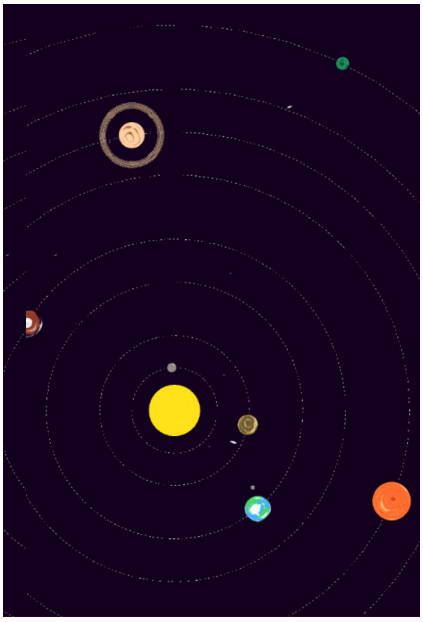
Utformingen av menyene samt hvor og når de dukker opp er en viktig del av å forsikre en god game loop. Av spillene vi har laget har kun BangShot og Bonkies et godt fungerende menysystem til å pakke inn spillet. I BangShot gikk vi for moderne knapper med avrundede kanter, kombinert med en pikselert font. Denne kontrasten fungerte bra, med tanke på at spillet var en remiks av det gamle spillet Missile Command (Atari, 1980).



Skjermbilder fra Bangshot og Bonkies

Menyskjermer leder spilleren inn og ut av spillet, fremfor å hoppe rett inn i gameplay

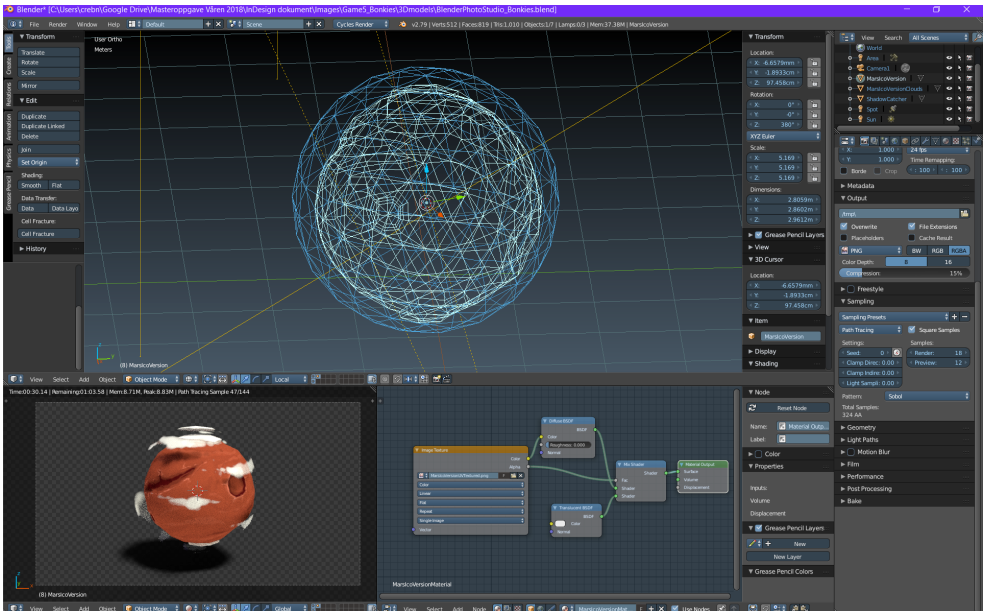
I Bonkies ledes spilleren inn i spillet via en hovedmeny som lar hen velge planet, bane og karakter. Deretter spilles selve spillet, og spilleren får informasjon om progresjon og tidsbruk gjennom grafikk på skjermen. Til slutt får spilleren en oppsummering av hvordan hen gjorde det i form av en infoskjerm, før spillet returnerer til skjermen for banevalg.



3D-MODELLERING OG ANIMERING

En del av spillkonseptene vi valgte å utforske involverte 3D modellering. Programvaren vi å brukte til dette var Blender[1] ettersom den er open-source og fullstendig gratis. 3D modellering er tidkrevende, spesielt hvis det skal animeres. Dette var noe vi erfarte i løpet av utviklingen av Super RPS, hvor vi animerte hendene. I Y.H.T.T.M.I.Y.H. brukte vi i stedet statiske 3D-modeller, med vesentlig lavere detaljnivå enn i Super RPS. Sistnevnte modeller fungerte veldig bra og fremstod i grunnen ikke særlig "prototypete" ut, i og med at kameraet var plassert mye lenger unna modellene enn det var i Super RPS.

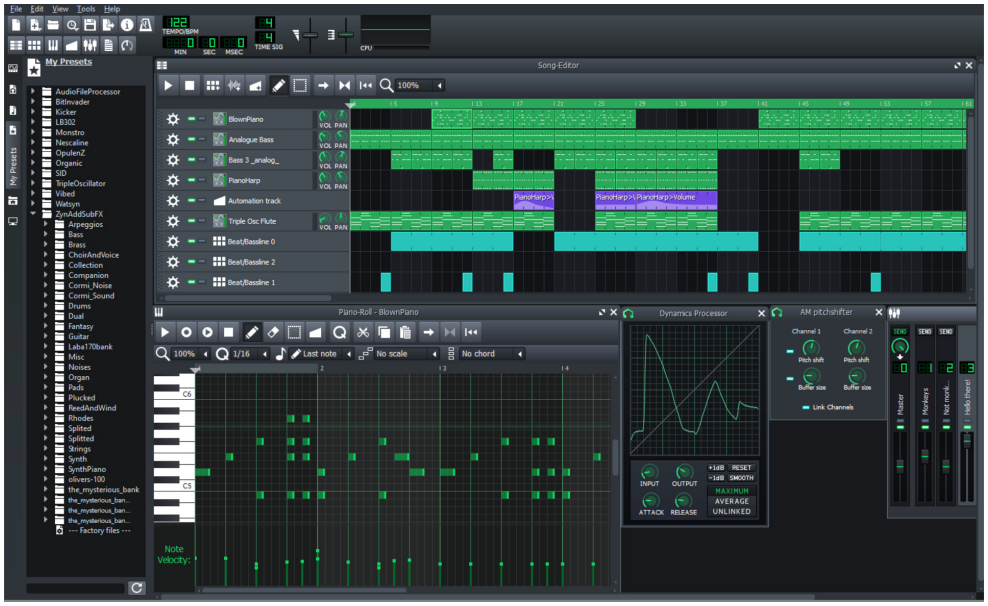
Dersom 3D-modellen skal tekstureres, som f.eks. planetene i menyene Bonkies, tar jobben med å ferdigstille 3D-modellen enda mer tid. Når jobben først er gjort kan gode 3D-modeller tilføre stor merverdi til spillet, gjøre verdenene mer levende og få spillet til å fremstå som mer velpolert.



Skjermdump fra Blender

Arbeidsområde i Blender, her under modellering av Mars for bruk i menyen i Bonkies

1 Blender.org



Skjermdump fra LMMS

Arbeidsområde i LMMS, her under komponering av musikk til Bonkies

LYD-DESIGN

Lyd er et viktig virkemiddel både for å skape stemning, kommunisere hvilken tilstand spillet er i, samt gi feedback på spillerens handlinger. Selv lyd på kun de mest sentrale aktivitetene har vist seg å vesentlig løfte spillopplevelsen, som i Tank Planet, hvor den eneste lyden man hører skudd og eksplosjoner (i tillegg til et lag av fuglekvitter)

I løpet av prosjektet har mye av testingen blitt gjort med lite eller ingen lyd. Vi ser på det som en styrke når testerne ser ut til å like konseptet selv uten lyd. Lyd har ikke vært sentralt for gameplaymekanikkene i spillene vi har laget. Vi har for eksempel ikke laget noen rytmebaserte spill, krevd timing av handlinger opp mot lydsignaler eller formidlet noen historie via lyd eller tale. Lydproduksjonen har dermed blitt prioritert lavt, ofte til fordel for bedre visuell feedback.

Lydeffekter har vi generert enten ved å laste dem ned fra freesound.org, hente dem fra Soniss sine gratis lydpakker gitt ut i

forbindelse med GDC[1] eller generert selv med verktøy som LMMS[2] eller ChipTone[3].

Til Bonkies og Bangshot produserte vi egen musikk til bruk i menyer og i under gameplay. Dette er jobb som krever musikalsk tekke, kjennskap til programvare som kan generere musikk (vi benyttet LMMS) og ganske mye tid.

Skjermdump av ChipTone

Webapplikasjon for generering av lydeffekter i 8- og 16-bit-stil

BRUKERTESTING

Størsteparten av testingen har vi utført selv under utviklingen, men takket være lokalene vi har sittet i mens vi har gjennomført denne masteren har folk stadig gått forbi og spurt om å få prøve siste versjon. I løpet av toukersprosjektene ble testing stort sett utført på denne måten, ved at interesserte personer kom og etterspurte å få prøve, eller at vi ba personer i nærheten om å komme å prøve.

Brukertestere identifiserer lett hva som ikke funker, og kommer

1 sonniss.com/gameaudiogdc18/

2 LMMS.io

3 SFBgames.com/chiptone

gjerne med forslag til hvordan man kunne gjort det bedre. Disse forslagene må tas med forbehold om at brukertesteren ikke nødvendigvis kjenner til systemet i detalj, eller hvordan mekanikker og endringer gjort på dem vil virke inn på hverandre. Det gjør ikke nødvendigvis designerne fullt ut heller før endringene har blitt testet, men poenget er at det er lettere å identifisere feil enn å komme opp med gode, fungerende løsninger, og det er fort gjort å gjøre for mange endringer samtidig. Da kan det bli vanskelig å identifisere hvilke endringer som egentlig er gode.

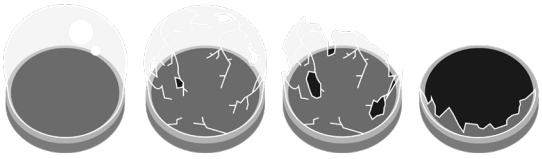
Bonkies har nå blitt vist frem flere ganger ved ulike arrangementer og sammenkomster, og vi har gjort flere endringer basert på observasjoner og feedback fra disse testene, som å endre timeren fra å telle uendelig oppover, til å telle nedover med en gitt tidsfrist. Dette gjorde vi basert på at folk så ut til å bli lei av å spille etter å ha brukt en 20-30 minutter på å løse et puzzle. Vi gjorde det også til et krav å bygge 100% ferdig, i stedet for å si at 80 % er godt nok og 100 % er supert. Alle testere prøvde å nå 100 % uansett. ”Monolittiske” puzzles uten så store krav til presisjonsmanøvre så ut til å fenge spillere mer

Brukertesting

Testing av Bonkies på utestedet Habitat i Trondheim

enn skjøre “stillas”-puzzles som hadde en tendens til å falle sammen frustrerende lett.

Vi gjorde det også mulig å knuse hverandres hjelmer, fordi alle forventet at det skulle gå an. Vi har lagt inn at jetpacken deaktiveres i det hjelmen knuser helt, mens robotarmen forblir operativ. Dette designgrepet var ment å øke risikoen for hasardiøs håndtering av byggeklosser (knust hjelm = ubrukelig spiller), og motivere til mer samarbeid og planlegging. Feedbacken så langt peker mot at det nok bør være mulig å reparere hjelmen igjen. Dette vil isåfall være en mekanikk som krever at en spiller hjelper en annen, hvilket vil være med på å bygge opp under følelsen av samhold (Jonassen, 2017).



MARKEDSFØRING

Markedsføring for å skape interesse for spillene var også noe vi måtte lære. Markedsføring av indiespill er ekstremt viktig for å få solgt enheter når det blir gitt ut, og vi har flere ganger blitt rådet til å starte markedsføring så tidlig som mulig.

Vi satset på grasrotmarkedsføring av spillene vi produserte, hvor vi henvendte oss til interesse-spesifikke publikum med spisset innhold som vi trodde de ville være interesserte i, i et håp om å maksimere effekten av den lille tiden vi kunne ta oss til markedsføringsinnsats. De sosiale mediene vi valgte å bruke var Instagram, Facebook, og Reddit. Facebook og Instagram ble benyttet 2-3 ganger i uka for å fortelle historien om Gauntlet Games, hva vi drev på med og hvilken progresjon vi hadde i spillene vi laget. Reddit ble stedet vi brukte for å skape interesse for de faktiske spillene vi laget ved å poste deler av konseptene på relevante subreddits. Vi opprettet også en nettside hvor vi publiserte informasjon om hvert prosjekt og som henviste videre til Facebook- og Instagramkontoen vår.

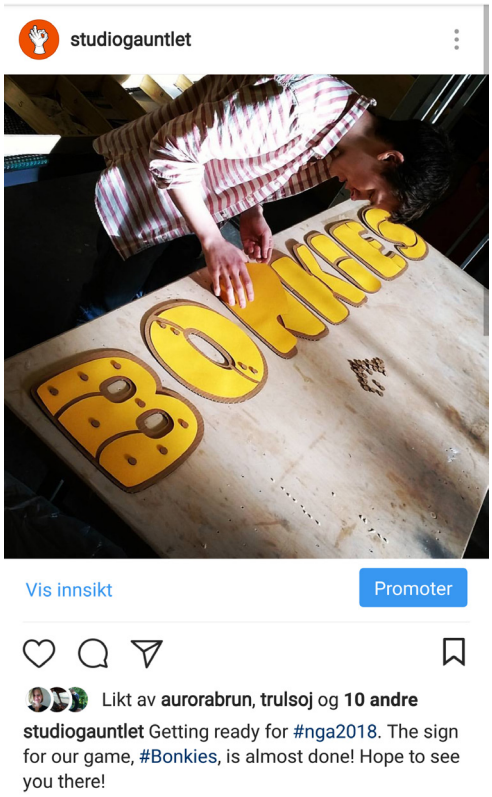
Vi har innsett at dette er en kreativ prosess helt uten fasit, men har erfart at dersom man vil tenne interessen til folk så kan det lønne seg å plukke ut en liten bit innhold fra spillet og vise den frem på en måte som ikke avslører for mye av det en ønsker å si. Dersom man deler et kort videoklipp med en tittel som ikke forklarer alt som foregår, så har brukere en tendens til å spørre etter mer informasjon. Bruk av Reddit viste seg å være effektivt for å nå et stort antall potensielle spillere over hele verden. Det er viktig å følge god skikk for Reddit-foraene, da andre brukere er raske med å fortelle en om man har begått en faux pas. Brukere later til å være vare for selvpromotering, men villige til å gå i dialog dersom man ikke prøver for hardt å selge dem et produkt.

Vi har også deltatt på en rekke sammenkomster i regi av Spillmakerlauget Midt og Spark NTNU som et ledd i prosessen for å ta del i det lokale spillutvikler- og oppstartsmiljøet.

MARKEDSFØRING I FORBINDELSE MED NGA 2018

Bonkies var et spillkonsept vi hadde jobbet på i forkant av masteren, og for gjøre det mer kjent samt prøve å karre til oss noen priser som kunne brukes videre i markedsføringssammenheng utsatte vi starten på masteren med tre uker. Disse ukene brukte vi på å få Bonkies i presentabel stand før Norwegian Game Awards (NGA) 2018. NGA er en prisutdeling i regi av Start NTNU, hvor man i 2018 kunne vinne både pengepremier samt standplass på SpillExpo 2018. På grunn av promoteringsverdien disse standplassene representerte var vi veldig motivert for å vinne dem. Plassene var en del av publikumsprisen (Gamer's Choice), en pris alle besøkende hadde anledning til å stemme på på selve arrangementsdagen. Allerede forrige semester under testing av Bonkies erfarte vi at spillet ikke bare hadde evnen til å fenge de som faktisk spilte, men også tilskuere.

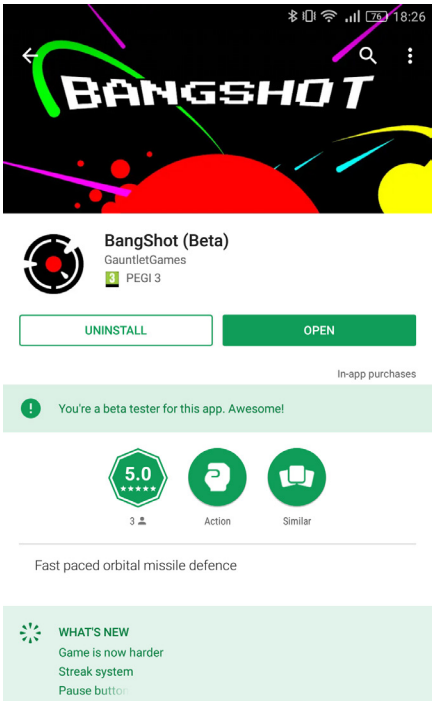
I et forsøk på å vinne publikumsprisen sørget vi for å ha en publikumsvennlig stand som skilte seg ut fra standard-standene. Vi sørget for å ha en større TV-skjerm enn alle andre, tydelige bokstaver som stavet ut navnet på spillet, høyt plassert for å synes over hodene på folk i lokalet, samt lave stoler foran TV-skjermen slik at publikum bak spillerne hadde god utsikt til det som foregikk på TV-en. I tillegg inviterte vi mange venner på arrangementet, da vi anså det som sannsynlig at venner av oss ville stemme på oss. Det endte med at vi vant både Best Design, og Gamer's Choice, så de tre ukene med ekstra utvikling samt jobben vi la ned i designet av standplassen så ut til å betale seg.



Norwegian Game Awards 2018

PUBLISERING

Publisering av spill kan foregå på flere måter. Enten kan man publisere selv, eller så kan man få hjelp av en profesjonell utgiver. De ulike spillplattformene tilbyr sine egne distribusjonskanaler hvor man kan publisere spillene sine. Playstation har Playstation Store, Xbox har Xbox Live Marketplace og Nintendo har Nintendo eShop. For mobil er de to største markedsplassene Google Play og App Store, og på PC finnes det en del alternativer, som Microsoft Store, itch.io og GoG, men Steam er den desidert største.



Skjermdump fra Google Play

Ett av målene våre var at alle spillene skulle publiseres på relevante plattformer etter hvert som de ble ferdige. For å samle alle spillene planla vi å publisere dem alle på Itch.io, en webside for publisering av spill, i tillegg til eventuelle andre mer egnede plattformer. Dette for å enkelt kunne henvise folk til Itch en oversikt over hele spillbiblioteket vårt. På Itch-siden vår lagde vi til å begynne med dummy-prosjekter med lovnad om at nye spill skulle slippes ved gitte datoer. Hensikten var å oppmuntre besøkende til å følge med på prosjektet og sjekke tilbake senere.

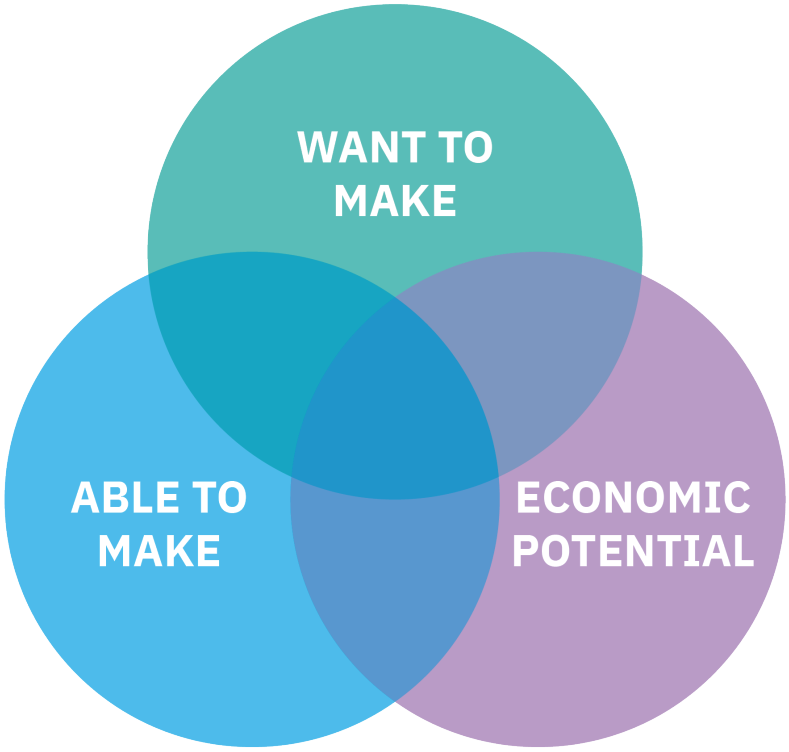
I løpet av prosjektet har vi publisert to spill, begge på egen hånd: BangShot på Google Play, og Super R.P.S. på itch.io. Bonkies er i en prosess hvor det venter på bli publisert på Steam Early Access. Vi har erfart at publisering krever mye marketing-materialer, skal det nå ut til folk, og at publiseringsprosessen i seg selv kan ta tid. Mens det å skaffe tilgang til publiseringsverktøyene på Google Play var en rask og enkel affære, krever publisering på Steam per i dag (25.05.18) at man betaler inn et depositum og så venter i 30 dager før spillet

kan publiseres. I tillegg må spillet ha hatt en “kommer snart”-side i minst 14 dager før spillet kan publiseres. Vi antar at prosessen med å publisere på en konsoll som f.eks. Nintendo Switch vil ta enda lenger tid.

MARKEDSUNDERSØKELSE

I forbindelse med at vi gjenopptok jobben med Bonkies, utførte vi en markedsundersøkelse hvor vi kartla populariteten av samarbeidsspill og hvilket prisnivå disse lå på, for å kunne gjøre et overslag på potensiell inntjening fra dette konseptet. Vi brukte SteamSpy[1] for å finne antall solgte enheter av samarbeidsspill, og bruke dette som en pekepinn for populariteten deres. I tillegg undersøkte vi hvordan denne sjangeren ble omtalt i medier og i ulike fora på nett. Steam, Playstation Store, Xbox Live Marketplace og Nintendo eShop ble brukt som referanser for prising av spillet.

1 steamspy.com, et nettsted for bruksstatistikk på spill kjøpt på Steam



Fasability-analyse

Modell for vurdering av spillideer (Clark, 2018)

ANALYSE AV SPILLKONSEPT

På et tidspunkt måtte vi velge ut et spill blant flere konsepter som vi ønsket å jobbe videre med. For å vurdere hvilket av disse vi skulle gå for brukte vi en metode presentert videoen “How to consistently make profitable indie games” (Clark, 2018). Vi vurderte alle spillene mot følgende kriterier: Om vi trodde spillet hadde kommersielt potensial, om vi hadde evnen til å ferdigstille spillet, og om vi i det hele tatt hadde lyst å jobbe med spillet. I tillegg forsøkte vi å identifisere potensielle hooks i enkelte av spillene (mer om dette om ca. 80 sider).

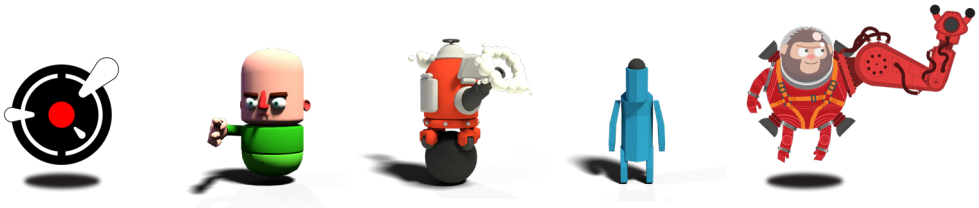
RESULTATER

RESULTATER

Resultatene fra dette masterprosjektet omfatter både de konkrete spillene som ble produsert, samt progresjon vi har gjort i forbindelse med bedriftsutvikling og markedsføringsinnsats.

Vi endte opp med å utvikle fem spill - BangShot, Super Rock Paper Scissors (Super RPS), Tank Planet, You Have To Tell Me If You're Human (Y.H.T.T.M.I.Y.H.), og Bonkies.

De to førstnevnte av disse spillene ble publisert, men samtidig anser vi disse for å ha svakest markedspotensial av de fem, BangShot fordi det er et mobilspill, og Super RPS fordi det krever mer jobb til å fungere bra enn vi har lyst eller anledning å legge ned i det. De tre siste



spillene ble ikke publisert. Vi erfarte at folk reagerte godt bare på små drypp av spillkonseptene, og når vi uansett ikke hadde anledning til å videreutvikle eller vedlikeholde noen av prototypene før masteren var overstått anså vi det som lite hensiktsmessig å legge dem ut. Likevel anser vi de tre siste spillene for å ha et reelt markedspotensiale dersom vi utvikler dem videre, da særlig på konsollen Nintendo Switch som har en profil og et design som fremmer sosial spilling. Både Tank Planet, Y.H.T.T.M.I.Y.H. og Bonkies er spill for for opptil fire spillere på samme skjerm.

Deler av alle konseptene har blitt delt på sosiale medier. Tilstedeværelsen vår på sosiale medier såvel som i det lokale oppstartsmiljøet har økt. Vi har 73 følgere på instagram, et domene og en nettside hvor vi kan poste informasjon og annonseringer, og vi har fått tilbud om å søke kontorplasser i Gründerbrakka på NTNU gjennom nettverket vi ble en del av gjennom Spark* NTNU. Vi har også fått forespørsel fra tiltspot.io (oppstartsbedrift ved NTNU som utvikler en web-basert spilleplattform) om å utvikle Y.H.T.T.M.I.Y.H. videre for utgivelse på deres plattform.


BANGSHOT
Angrepet fra alle kanter
1
2
3
4
5

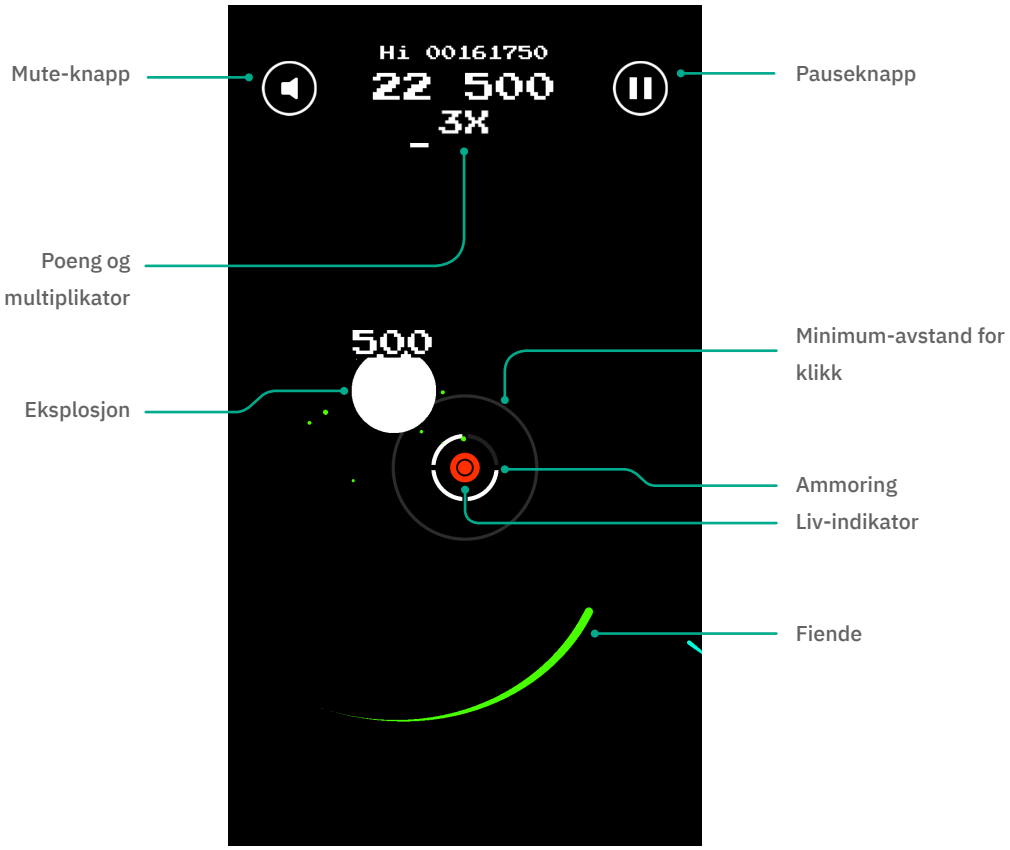
BANGSHOT

BESKRIVELSE

BangShot tar mye inspirasjon fra det klassiske arkadespillet Missile Command (Atari, 1980), hvor hovedmekanikken går ut på å fyre av raketter mot et punkt på skjermen man selv bestemmer. Rakettene detoneres ikke før de når punktet, så hvor man sikter må times riktig, skal man hindre de innkommende fiendtlige missilene fra å nå bakken. I BangShot er bakken representert av en sirkel midt på skjermen, og missilene kommer inn fra sidene og nærmer seg midten. Det finnes tre forskjellige fiender, og de har ulike bevegelsesmønstre. Fiendene kommer i bølger, og intensiteten i fart og kompleksiteten på bevegelsesmønstrene deres øker etter hvert som bølgene passerer.

Spilleren har tre liv, representert som tre konsentriske sirkler i midten av skjermen. Spilleren har et gitt antall skudd tilgjengelig, representert ved hvite ringsegmenter rundt de røde sirklene. Spilleren mister ett liv for hver fiende som når midten. Spilleren kan klikke på skjermen, hvorpå det sendes ut et prosjektil fra midten av skjermen til punktet hvor spilleren tappet og detonerer. Fiender som befinner seg i missilets sprengradius bekjempes, og man får poeng for dem. Dersom samme eksplosjon tar ut flere fiender, multipliseres summen av poengene med antall fiender som ble bekjempet. For hver fiende som tas ut, får man tilbake ett skudd. Blir man fri for skudd begynner ammunisjonsringen å fylles opp igjen, men dette tar tid, og man kan ikke skyte igjen før ringen er helt full.

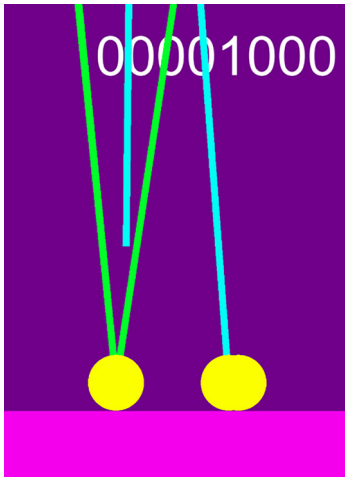
For hver bølge av fiender får spilleren valget mellom å øke antallet skudd hen har tilgjengelig, eller regenerere et liv (maksimalt tre liv).


Mute-knapp
Hi 00161750
22 500
3X
Pauseknapp
Poeng og multiplikator
500
Minimum-avstand for klikk
Eksplosjon
Ammoring
Liv-indikator
Fiende

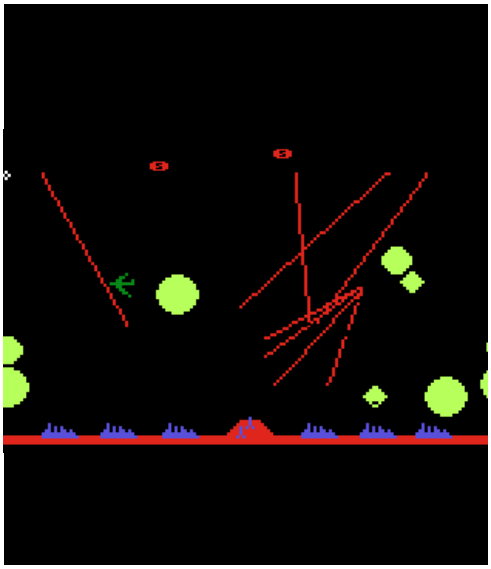

Hi 00161750
0
1X
BANGSHOT
PLAY !
GauntletGames © 2018
Hi 00161750
46 950
4X
WAVE 7 CLEARED
choose 1
Hi 00161750
114 150
1X
GAME OVER
TRY AGAIN
Like the game?
Buy us a coffee
or follow us @studiogauntlet

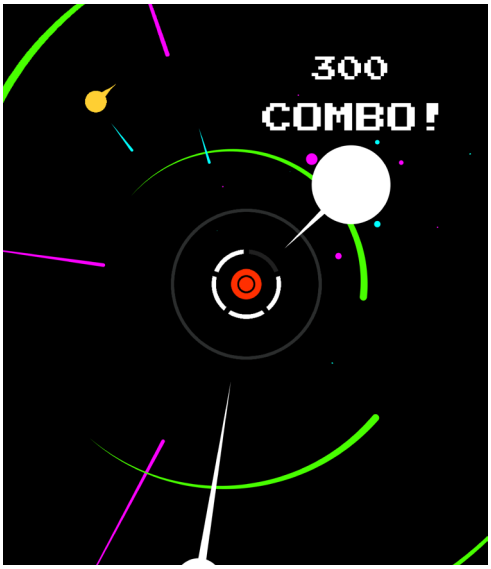
PROSESS

1. Det startet med en idé om et spill hvor vi benyttet mobilens akselerometer for å trille en kule rundt og prøve å fange å beskytte bakken mot stråler som kom ned fra toppen av skjermen (se bilde til høyre).
2. Endret konseptet til det som skulle bli endelig versjon av BangShot etter at første versjon ikke var særlig gøy.
3. Implementerte en valgfri in-app-purchase funksjon for å lære hvordan ta betalt i en app.
4. Dedikerte en hel dag kun til promotering av spillet. Vi lagde en trailer som vi postet på imgur, og delte på facebook og instagram.



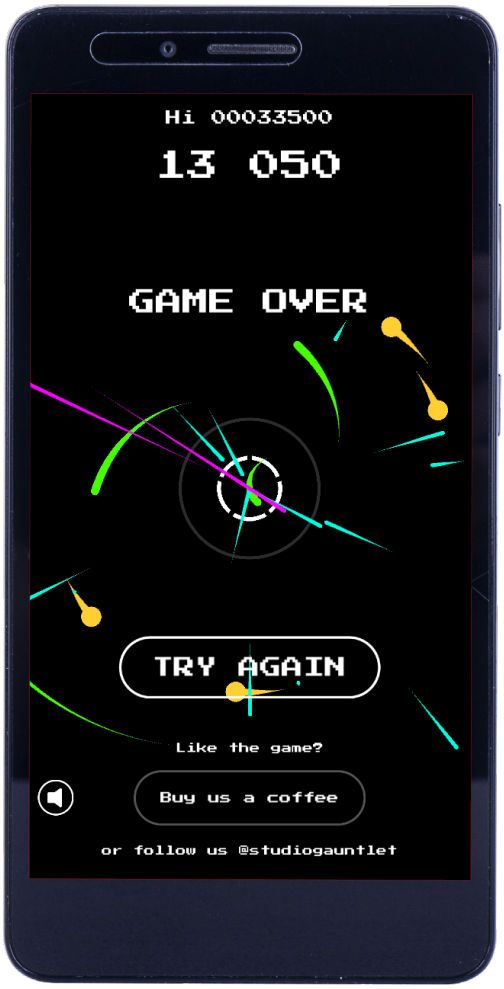
Tidlig versjon av BangShot





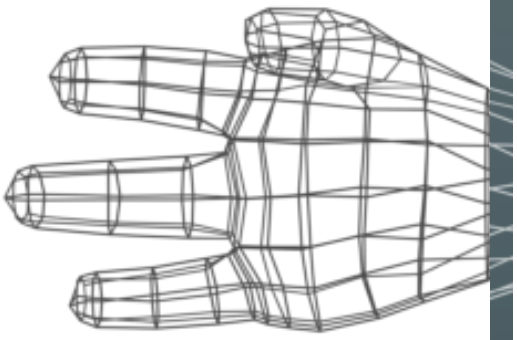
Missile Command og BangShot

Missile Command-skjermbilde hentet fra Wikipedia.org/wiki/Missile_Command



SLUTTRESULTAT

Bangshot endte som et ferdig mobilspill publisert på Google Play, for Androidtelefoner. I skrivende stund har spillet 19 aktive installeringer, 31 installeringer totalt samt 54 kroner i inntjening, og noen av disse kronene kom fra personer som ikke er venner eller familie. Vi anser spillet som en god betaversjon av et ferdig spill.



SUPER R.P.S.

Førstepersons stein-saks-papir-sisten

1 2 3 4 5

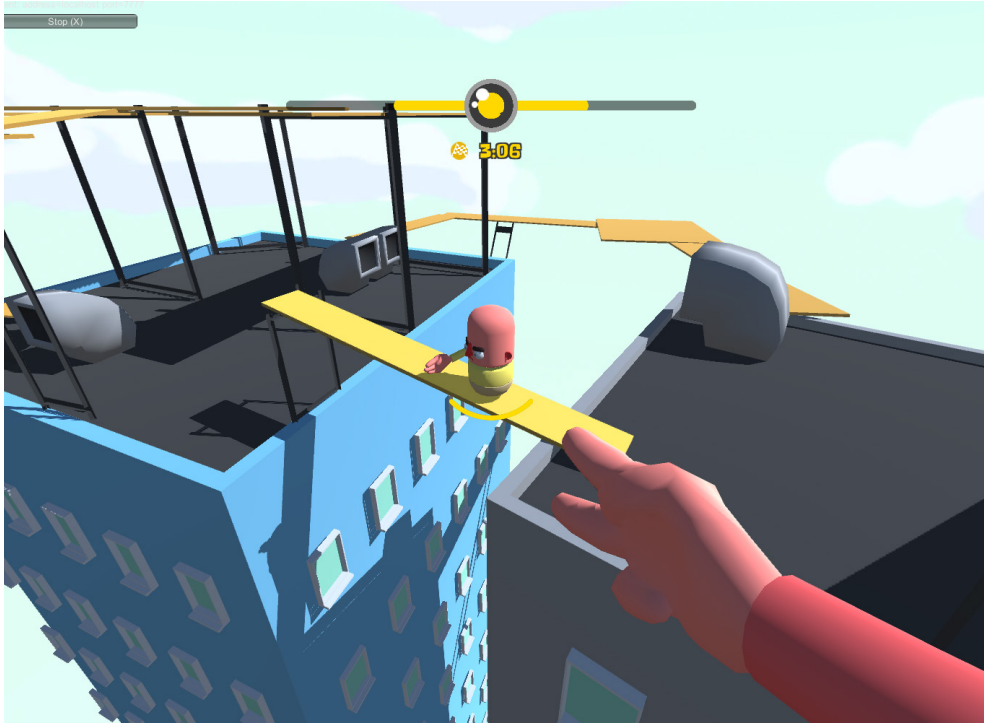
SUPER R.P.S.

BESKRIVELSE

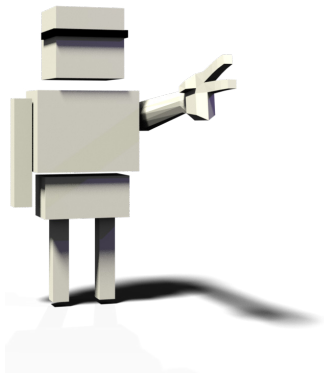
Super RPS (Rock Paper Scissors) kan best beskrives som et klassisk førstepersons skytespill hvor alle våpnene er nærkampvåpen, representert ved spillerens hånd i stein-, saks- eller papirpositur. Som i stein-saks-papir kan man kun skade/ta skade dersom man angriper/blir angrepet av en korrekt hånd. På skjermen er det en teller som teller ned til neste våpenbytte. I det telleren når null stopper alle spillerne opp, og alle tvinges til å velge stein saks eller papir på nytt. På denne måten vil rollen som jeger og jaget kunne skifte veldig fort.

Spillerne har evnen til å løpe, sprinte, hoppe og angripe. Sprinting bruker opp energi, representert ved en bar på midten av skjermen. Energi fylles opp igjen så lenge man ikke sprinter, med en liten forsinkelse etter at man har sluttet å sprinte. Sprintefunksjonen og forsinkelsen før den begynner å fylles opp igjen bidrar til å gjøre jagesekvensene mer intense.

Runden er over etter fem minutter, selv om runden i nåværende versjon varer i en time og er mer som en test-lekegrind å regne.



Skjermbilder fra Super RPS

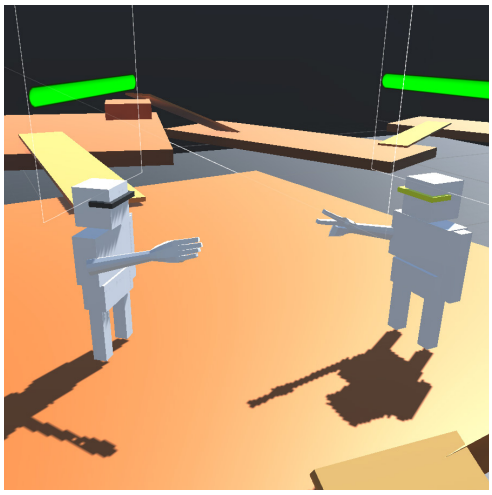


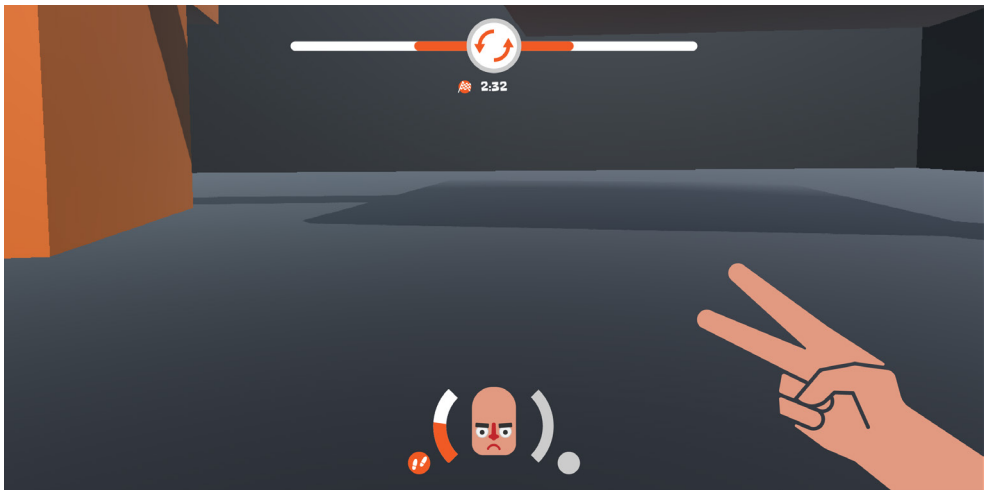
PROSESS

1. Vi hadde en idé om at vi ville lære oss å lage spill som kunne spilles over nett. Vi antok at networking ville bli teknisk krevende, så vi satset på et enkelt spillkonsept, nemlig stein saks papir. Ideen for nettopp stein saks papir fant vi ved å be brukere på Reddit om ideer for vårt neste spill.
2. Vi valgte å gjøre det i 3D grunnet en mistanke om at blandingen mellom det veletablerte og seriøse first person-uttrykket og den banale stein-saks-papir-mekanikken kunne bli veldig gøy med tanke på promotering.
3. Implementering, modellering og animasjon tok lenger tid enn vi forventet, og spillet ble hastig ferdigstilt og publisert, uten særlig markedsføring.

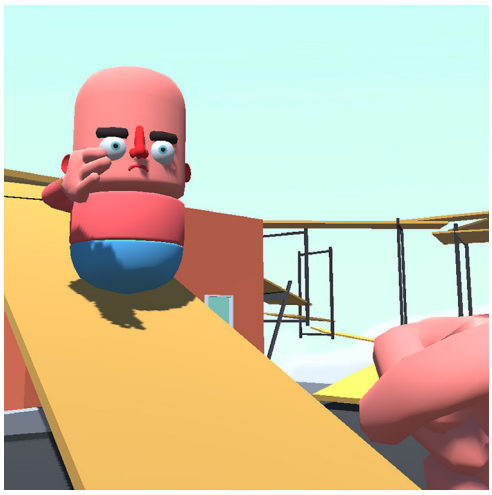
SLUTTRESULTAT

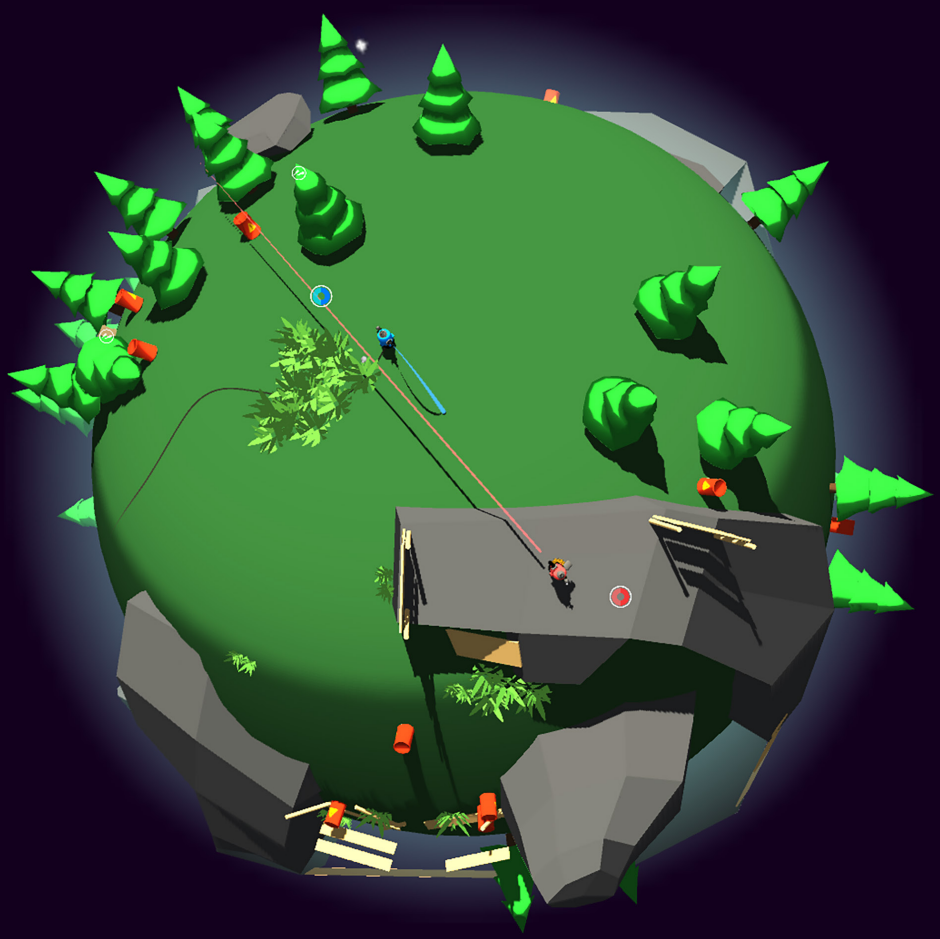
Super RPS endte som en spillbar prototype gitt ut på vår Itch.io side, med 6 nedlastinger totalt.




2:32


0:19
1
2
3
0 of 2
ready





TANK PLANET

Kanonkrig på en klode

1 2 3 4 5

TANK PLANET

BESKRIVELSE

Tank planet er et lokalt flerspillerspill hvor målet er å bekjempe sine motspillere. Opp til fire spillere kontrollerer hver sin avatar som kan bevege seg rundt på en sfærisk bane. Avataren er utstyrt med en kanon, og hver spiller har tre liv. Spillere kan når som helst hoppe inn og ut av spillet, og alt foregår på samme skjerm.

For hvert liv har spillerne en gitt mengde helsepoeng representert ved en livmåler og en begrenset mengde energi som kan brukes til unnvikende manøvre, representert ved en energimåler. Spilleren har også enten en begrenset mengde skudd tilgjengelig eller en tidsforsinkelse for når han får lov å fyre av neste skudd. Dette er representert ved en ring som kan krympes radielt. Informasjon om helse, energi og ammunisjon er pakket inn i en og samme glyf.

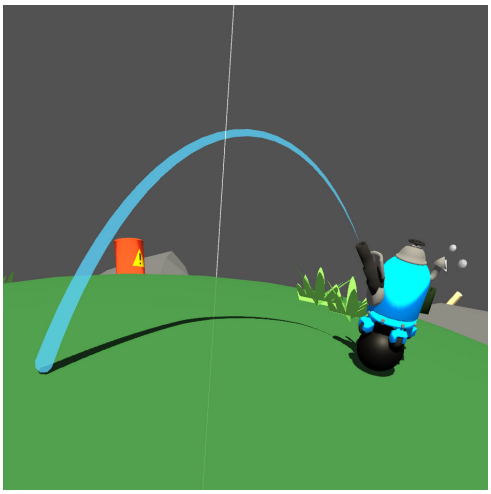
Spillernes standardvåpen er et skudd som fyres av i en bue oppover før det lander et stykke foran spilleren. Treffet lager en eksplosjon hvor alt innenfor eksplosjonsradiusen tar skade. Standardvåpenet fyrer av missiler som beveger seg ganske sakte, og når alle spillerne har dette våpenet er tempoet i spillet forholdsvis rolig. Når en spiller får tak i en våpenoppgradering, som f.eks. et maskingevær, går tempoet vesentlig opp, og rollene går fort fra å være alle mot alle til å ligne mer på en alle mot én-situasjon.

Skjermbilder fra Tank Planet

Spillere kan få tak i våpenoppgraderinger og ekstra liv fra kasser som faller ned mot planeten (se bilde over). Kassene begynner alltid å falle slik at de vil lande langs planetens horisont, for å motivere spillerne til å bevege seg rundt på hele planeten.

PROSESS

1. Den første ideen var å ha biler med våpen som kjørte rundt i et post-apokalyptisk ørkenlandskap. Vi ville gjenskape følelsen av å kunne kjøre i en evig ørken mens man kjempet mot hverandre, inspirert av Mad Max Fury Road. Vi gikk for en sfærisk bane, slik at man kunne kjøre rundt og rundt uten at vi trengte å tenke på å fortløpende måtte generere nytt landskap.

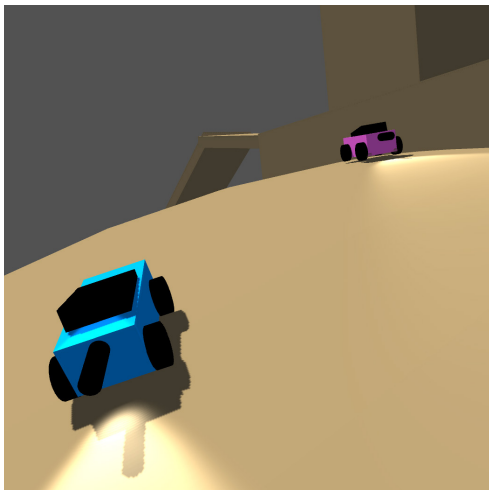


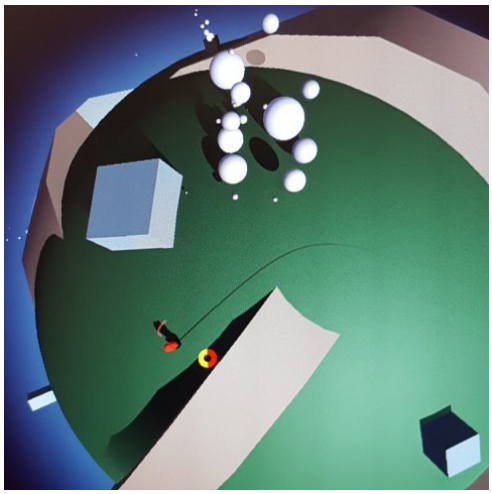
Skjermdump fra Unity

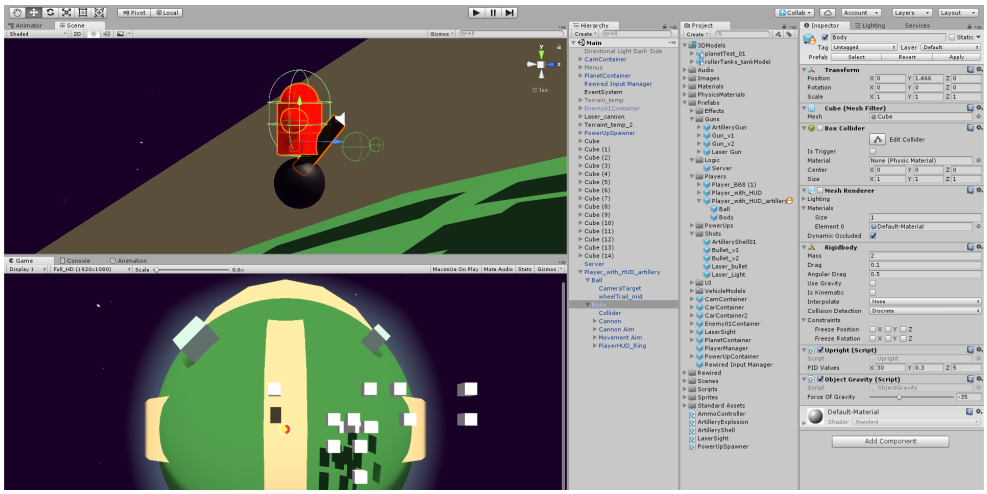
Forhåndsvisning av prosjektilbane

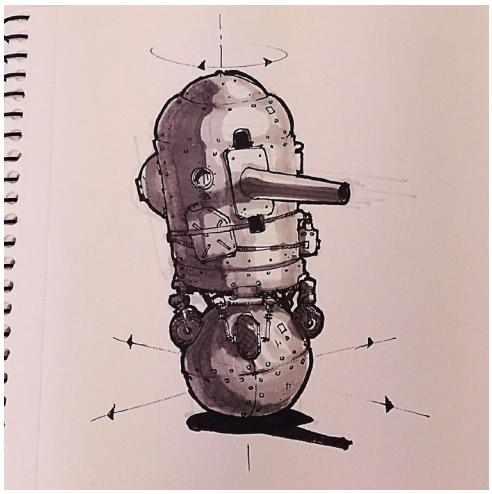
2. Vi slet med å få til en komfortabel bilfysikk, så for å forenkle utviklingen gjorde vi kjøretøyene om til enkle kuler, og så monterte vi et kanontårn oppå ballen slik at avataren nå minnet mer om en selvbalanserende robot.

3. Kanontårnet justerte seg i forhold til overflaten på planeten ved hjelp av en PID-kontroll, og beveget seg derfor ganske komisk under akselerasjon og støt fra omgivelser og andre spillere. Derfor ble tonen i hele spillet gjort mer lystig, noe som gjenspeiles bl.a. i fargebruk, karakter- og i banedesign.










JOIN

SLUTTRESULTAT

Tankplanet endte opp som en fullt spillbar demo av kjerne-gameplayet. Vi valgte å ikke publisere denne på internett men satset heller på å slippe markedsføringsmateriale. Bildet vi postet på Reddit endte opp med 6.6k visninger og titalls kommentarer fra brukere som ønsket å få vite mer informasjon om prosjektet, noe vi anser som en god pekepinn på at dette prosjektet har potensiale.

Tre iterasjoner av karakteren i Tank Planet

Modellen øverst til venstre er bygget av primitive former tilgjengelige i Unity. De to andre er modellert i Blender.

[–] badadvice4all 49 poeng 2 måneder siden
Can we get some screenshots of in-game stuff? Looks good here, but I'd like to see how it's rendered in-game compared to here. Good post OP.
permalenke embed lagre rapporter gi gull svar

[–] Atomic_Command [S] 40 poeng 2 måneder siden
Thank you for the feedback!
And yes, things look different in-game compared to the render, we haven't had time to actually put the latest iteration into the game (still needs UV unwrapping/baking and the 2 weeks we alloted to working on this game just ran out) but you can see how version 2 looks in-game, in this video.
permalenke embed lagre foreldre rapporter gi gull svar

[–] badadvice4all 11 poeng 2 måneder siden
Very cool. Thanks for the reply.
permalenke embed lagre foreldre rapporter gi gull svar

[–] Troxicale 10 poeng 2 måneder siden
This looks sick! Where can i find out more about the game?
permalenke embed lagre foreldre rapporter gi gull svar

[–] Atomic_Command [S] 3 poeng 2 måneder siden
Thank you so much! For further updates on this, the best place to check out is probably our website.
permalenke embed lagre foreldre rapporter gi gull svar

[–] Troxicale 2 poeng 2 måneder siden
Awesome!
permalenke embed lagre foreldre rapporter gi gull svar

[–] stb1Cluny 1 poeng 2 måneder siden
For real this game looks super fun!
permalenke embed lagre foreldre rapporter gi gull svar

[–] Mricypaw1 2 poeng 2 måneder siden
Newbie here. You mentioned, you still need to UV unwrap the model. Is there a way to apply textures to a model like you did here without UV unwrapping the mesh? Thx
permalenke embed lagre foreldre rapporter gi gull svar

[–] sticklebackridge 2 poeng 2 måneder siden
I believe this is the high poly mesh, so they will bake these textures onto a lower poly mesh, this doesn't look textured to me. You can also use generated coordinates instead of UVs
permalenke embed lagre foreldre rapporter gi gull svar

[–] Atomic_Command [S] 2 poeng 2 måneder siden
Yup, anything is possible with nodes! This is what the node tree looked like for this model: link. I used the texture coordinate node set to "Object" to get the paint chips to apply uniformly, and all the little details, like the bolts, are actual meshes.
permalenke embed lagre foreldre rapporter gi gull svar

[–] tj2708 14 poeng 2 måneder siden
Thats really cute I wanna hug it
permalenke embed lagre rapporter gi gull svar

[–] -Rivox- 3 poeng 2 måneder siden
as you can see, he's very happy to be hugged. Maybe too happy?
permalenke embed lagre foreldre rapporter gi gull svar

[–] tj2708 6 poeng 2 måneder siden
Is that an artillery gun in your pocket or are you just happy to be hugged
permalenke embed lagre foreldre rapporter gi gull svar

[–] Olive-Snake 17 poeng 2 måneder siden
Does anyone else think the first iteration looks like it has a boner?
permalenke embed lagre rapporter gi gull svar

[–] Atomic_Command [S] 2 poeng 2 måneder siden
(͡° ͜ʖ ͡°)
permalenke embed lagre foreldre rapporter gi gull svar

[–] Mida2010 1 poeng 2 måneder siden
r/mildlypenis
permalenke embed lagre foreldre rapporter gi gull svar

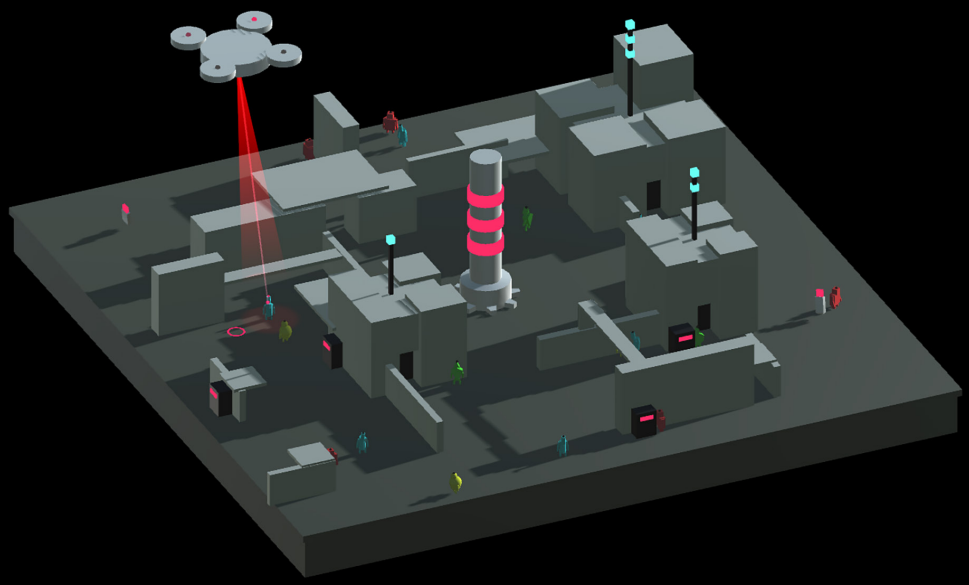
[–] TheGreatB3 8 poeng 2 måneder siden
Nicely done! This looks like it was inspired by the boss Megaleg from Super Mario Galaxy.
permalenke embed lagre rapporter gi gull svar

[–] MawoDuffer 5 poeng 2 måneder siden
Did you make a sketch and go with that or go along the way as you made it?
permalenke embed lagre rapporter gi gull svar

[–] Atomic_Command [S] 9 poeng 2 måneder siden
I always make at least 1 sketch before I attempt to model it, it helps me to figure out the silhouette and where the major shapes go. However the real answer is I do both, I sketch a bit, then model a bit, then do some sketches based on the model, and vice versa, always trying to make it a little bit better each time.
permalenke embed lagre foreldre rapporter gi gull svar

Skjermdump fra Reddit

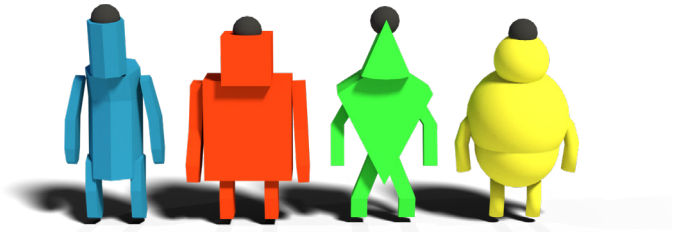
Kommentarfeltet på reddit/r/blender etter å ha postet bildet til venstre



Y.H.T.T.M.I.Y.H.

Forsvinn i mengden

1 2 3 4 5



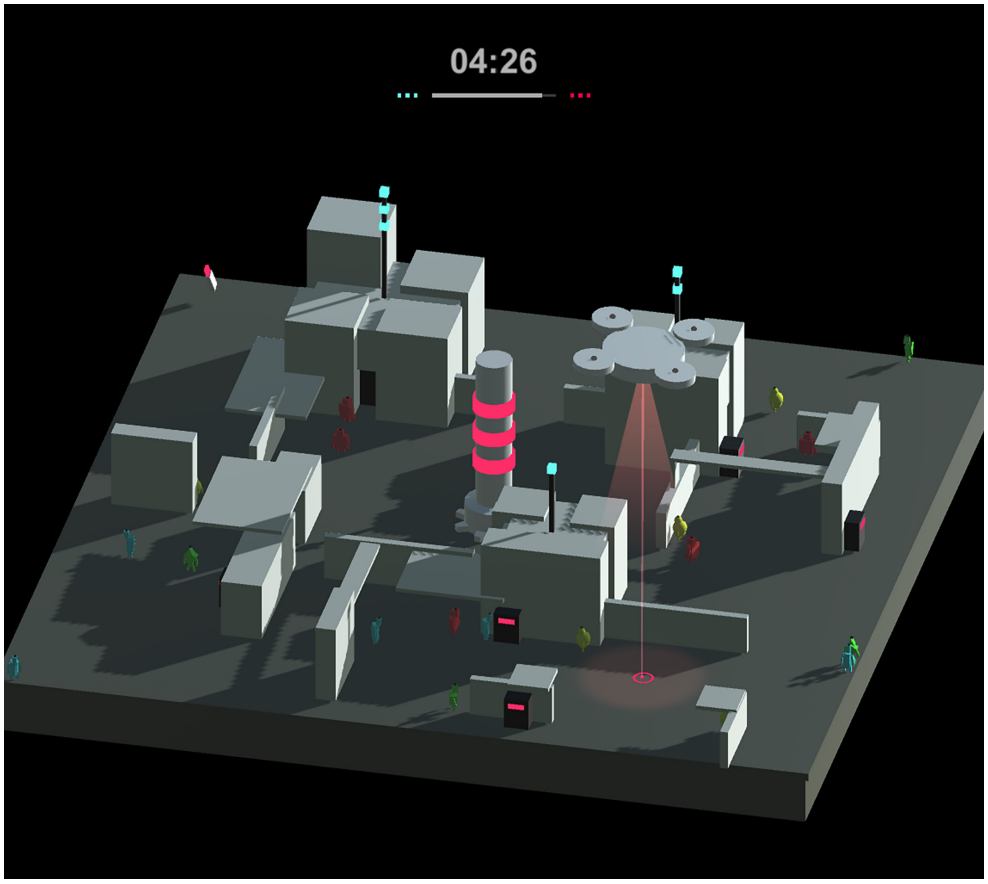
Y.H.T.T.M.I.Y.H.

BESKRIVELSE

You Have To Tell Me If You're Humaner et asymmetrisk flerspiller-spill som foregår på samme skjerm. Opp til tre spillere styrer sin avatar (menneske) på et 3D-plan, og går rundt blant en mengde AI-styrte avatarer (androider) mens de prøver å samle objekter som er spredt rundt på kartet for så å levere dem til gitte punkter. Én av spillerne styrer en drone som svever over alle avatarene, og dens oppgave er å identifisere og tilintetgjøre de menneskelige spillerne uten å ta ut androider.

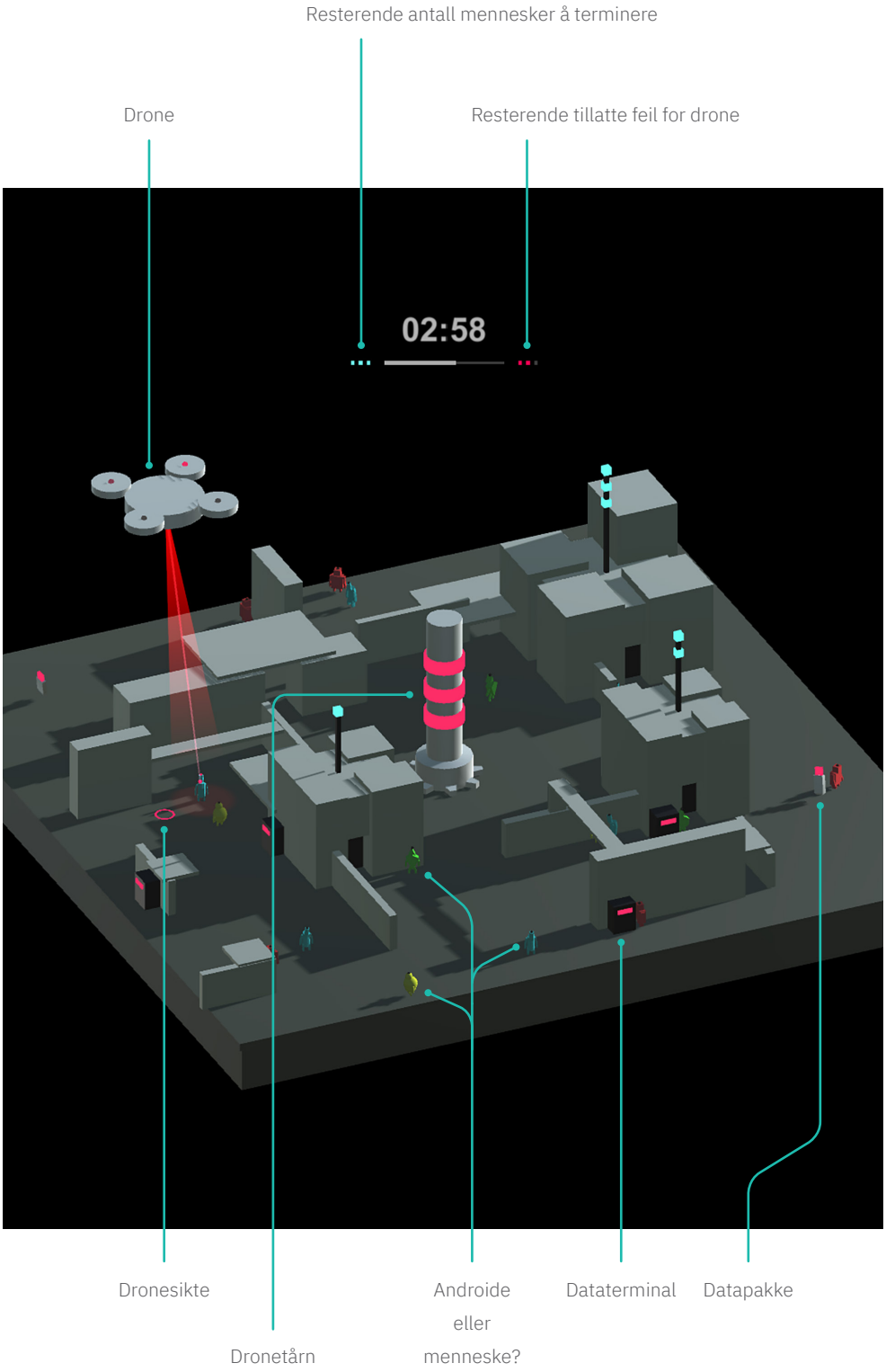
Dronen har et lasersikte som peker rett ned, men som låser seg fast til en avatar dersom den kommer i kontakt med strålen. I det lasersiktet har låst seg på en avatar har dronen noen verktøy til disposisjon for kunne skille ut og dømme mennesker fra androider:

1. Strålen aktiverer automatisk et lys på hodet til androider. Mennesker må tenne lyset manuelt med et knappetrykk.
2. Dronespilleren kan velge å markere avatarer av interesse med en indikator som holder seg svevende over hodet på en avatar. Mennesker har anledning til å fjerne indikatoren med et knappetrykk, ideelt sett mens de og indikatoren befinner seg bak en hindring så de ikke avslører seg selv.
3. Dronespilleren kan skyte og eliminere avataren.



Skjermbilde fra Y.HTTMIYH

Én spiller styrer dronen, opp til tre spillere styrer hver sin avatar nede på bakken blant AI-kontrollerte avatarer.


Resterende antall mennesker å terminere
Drone
Resterende tillatte feil for drone
02:58
Dronesikte
Dronetårn
Androide eller menneske?
Dataterminal
Datapakke

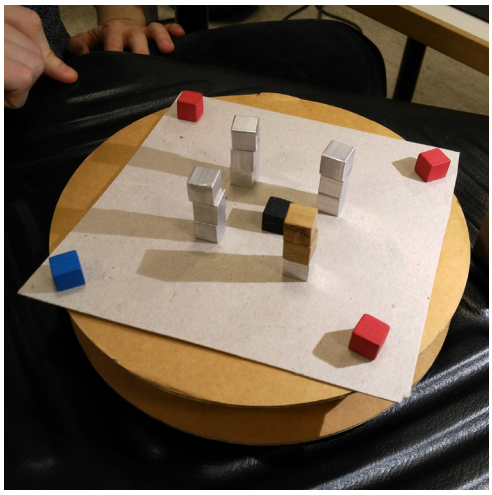
For å hindre dronespilleren fra å skyte vilt rundt seg kan den bare treffe tre androider før den taper. Disse forsøkene fylles opp igjen så snart dronen eliminerer et menneske.

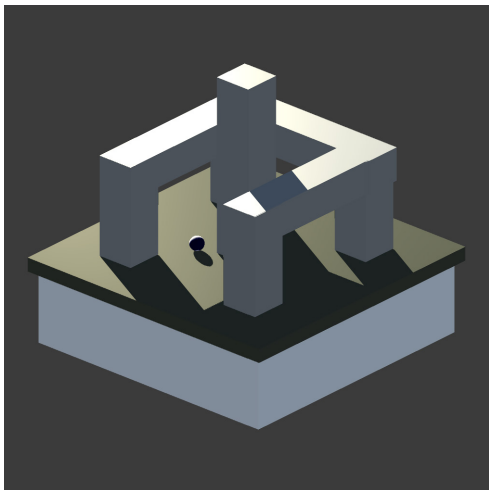
Menneskenes mål er å deaktivere dronen ved å plukke opp datapakker som dukker opp på kartet ved faste intervaller, på et tilfeldig valgt sted av et sett forhåndsbestemte steder. Disse pakkene kan plukkes opp av alle avatarer. Dersom en androide kommer nær en datapakke, forsvinner den og er den i praksis tapt. Dersom et menneske kommer nær en datapakke forsvinner den, men spilleren bærer den nå på seg, og må prøve å komme seg til en av dataterminalene som står på brettet. I det en spiller med en datapakke kommer nær en terminal, blinker terminalen slik at alle kan se hvor det nettopp ble levert en datapakke, og terminalen er inaktiv i 30 sekunder før den kan brukes igjen. Når like mange datapakker er levert som det er rosa lys på dronetårnet har dronen tapt.

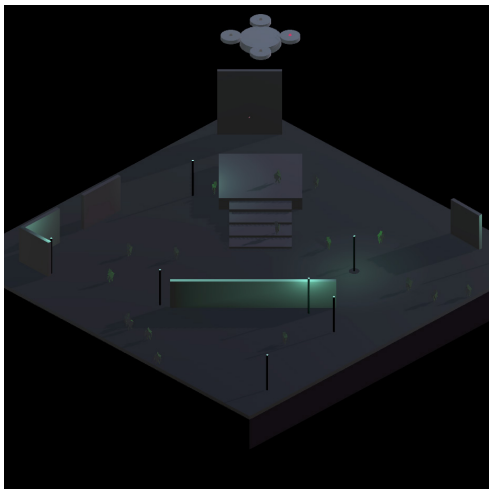
Spillerne kan bytte forkledning. Dette bytter ut karaktermodellen deres, lager en liten røyksky, og fjerner markøren på spilleren, dersom hen er markert.

PROSESS

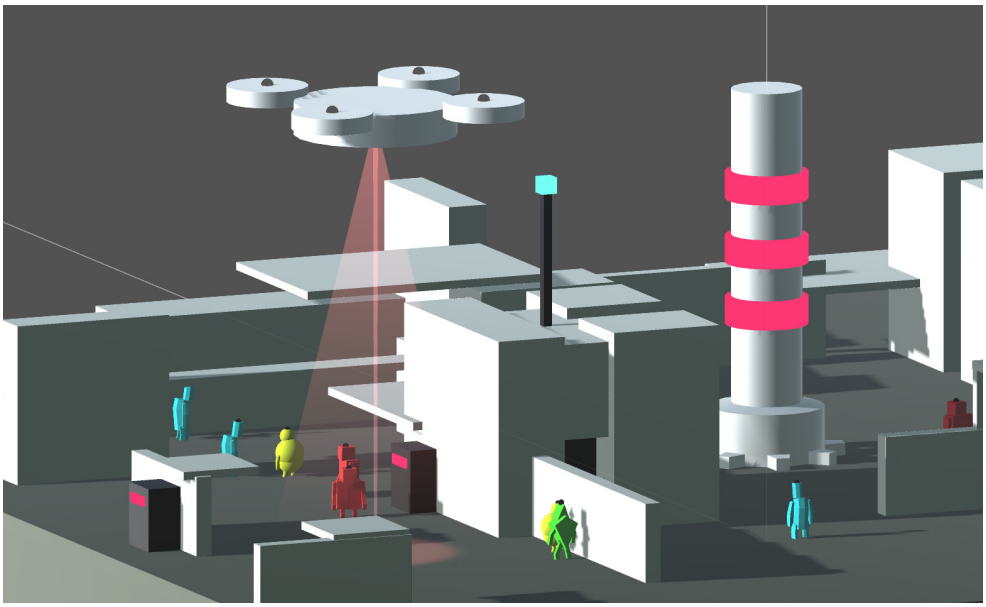
1. Den første ideen vi hadde var å lage et spill som lekte med lys og skygge. Tanken var å vise et lite diorama som kunne vris, lyssatt av et lys fra en fast vinkel. På den måten ville skygger falle på bakken avhengig av hvordan man vridde dioramaet, og tanken var at spilleren måtte manøvrere seg mellom to punkter uten å bli truffet eller sett av av dette lyset.
2. Det viste seg problematisk å designe morsomme puzzles, og i tillegg var dette et prosjekt vi trodde kom til å kreve mer tid til produksjon av gode puzzles enn det vi hadde. Men vi likte diorama-looken, og gjenopptok en gammel idé om et spill hvor flere spillere spiller på samme brett og jakter på hverandre uten å vite hvem de andre er.
3. Vi lagde en kjapp prototype av menneske- og AI-kontrollerte spillere på et enkelt brett, samt en funksjon for å tagge en valgfri avatar, og merket at grunnkonseptet fungerte veldig bra til tross for (eller kanskje rettere sagt takket være) superenkel AI.
4. Det visuelle uttrykket justerte vi til å passe til AI-folkemengden og gjemsel-mekanikken. Måten å velge ut avatarer på ble gjort om fra en peker på skjermen til en truende overvåkingsdrone, og miljøet og stemningen ble gjort mørkere og mer Blade Runner-esque.
5. Etter mye brukertesting ble det gjort mange eksperimenter med evner som dronen kunne ha for å avdekke avatarenes identitet.
6. Vi delte en forenklet versjon av gameplayet på Reddit i form av en GIF, og fikk god respons på konseptet







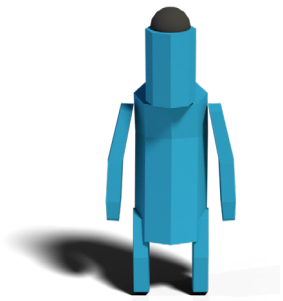

01:17



Skjermdump fra Unity
Gameplay fra Y.H.T.T.M.I.Y.H.

SLUTTRESULTAT

Y.H.T.T.M.I.Y.H. endte som en fullt spillbar demo av kjerne-gameplayet. Igjen valgte vi å ikke slippe noe spillbart til internettet og heller satse på markedsføringsmateriale. Vi hadde nok en gang god suksess med denne formelen. En GIF av gameplayet postet på Reddit fikk 10.1k visninger, og igjen mange kommentarer fra engasjerte brukere som ønsket å vite mer.



mine subredditer · hjem · popular · alle · tilfeldig · users | askreddit · worldnews · todayilearned · gaming · gifs · mildlyinteresting · aww · television · jokes · prequelmemes · iama · crappydesign

unity3D **Kommentarer**

482 Prototype of a same screen hide-and-seek game. 3 players hide in the crowd. 1 player has to spot and kill them.

sendt inn 2 måneder siden av Joelspeanuts

69 kommentarer del lagre gjem slett nsfw spoiler FLAIR crosspost

alle 69 kommentarer - sorter etter: beste Deaktiver innbokssvar (?) pin to profile

save innholdspolicy

[–] Madnessbox 40 poeng 2 måneder siden

Reminds me of http://store.steampowered.com/app/303590/Hidden_in_Plain_Sight/

Looks good though! :)

permalenke embed lagre rapporter gi gull svar

[–] wstdsgn Hobbyist 21 poeng 2 måneder siden

or this: http://www.spyparty.com/

permalenke embed lagre foreldre rapporter gi gull svar

[–] Roborobob 3 poeng 2 måneder siden

I remember seeibg that game back in like 2011 and getting hype about it..... The hype has sort of worn off by now :(

permalenke embed lagre foreldre rapporter gi gull svar

[–] rootyb 7 poeng 2 måneder siden

Yeah, it's taken a long-ass time, but you should re-warm your hype. It's an incredible game, and has spent the intervening years getting polished to a mirror finish (well, maybe a semi-gloss, so far but the progress has been remarkably steady).

It's gearing up for a Steam Early Access launch in the (actually) near future, and, IMO, is already well worth the $15 price.

permalenke embed lagre foreldre rapporter gi gull svar

[+] *[slettet] 2 måneder siden (2 barn)*

[–] streaker03 3 poeng 2 måneder siden

this game is fantastic btw I hope someone out there (possibly OP) release a successful one

permalenke embed lagre foreldre rapporter gi gull svar

[–] Roborobob 2 poeng 2 måneder siden

That game is amazing

permalenke embed lagre foreldre rapporter gi gull svar

[–] Joelspeanuts [S] 4 poeng 2 måneder siden

Yeah we're definitely playing around with similar mechanics. But I think hope pleasedontsueus our aesthetics, theme and the way players interact will set them sufficiently apart.

permalenke embed lagre foreldre rediger Deaktiver innbokssvar slett svar

[–] Roborobob 2 poeng 2 måneder siden

If they did sue you and win on THAT, the ramifications would be felt across the video game world? "Oh you shoot people in Battlefield 1? Well you also shoot at people in Star Fox 64 soooooooooooo....."

permalenke embed lagre foreldre rapporter gi gull svar

[–] iams3b @hellosebby 1 poeng 2 måneder siden

Hey man, don't even worry about that. It's a genre, and I'm sure you can bring something new to the table (: It's kind of like how there are so many "connect-4-block" games - just because they all share the same/similar mechanics doesn't make them the same game!

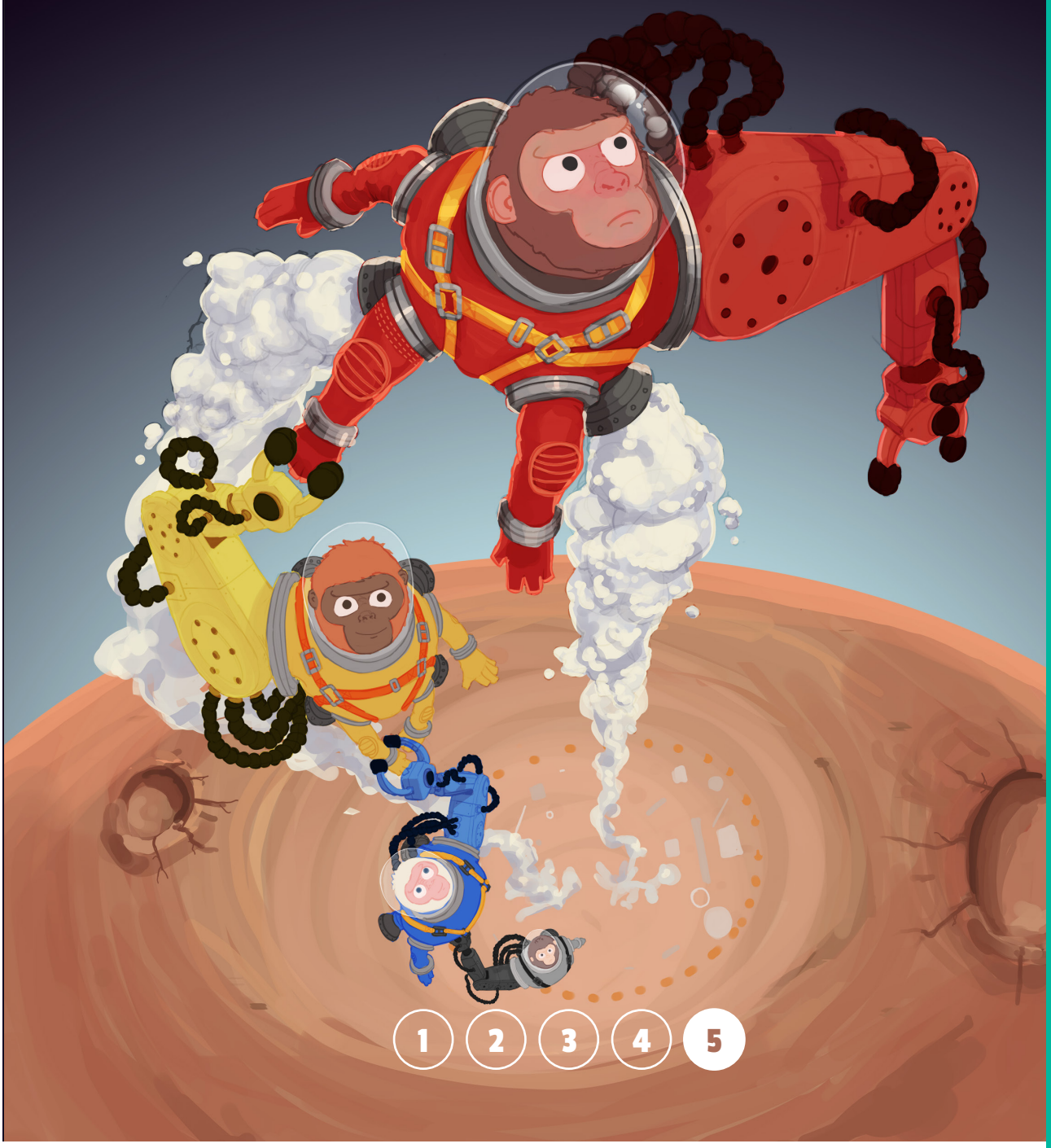
permalenke embed lagre foreldre rapporter gi gull svar

Skjermdump fra Reddit

Kommentarfeltet under posten vår med GIF af Y.H.T.T.M.I.Y.H.-gameplay

BONKIES

Kløvete konstruksjonsarbeid



1 2 3 4 5

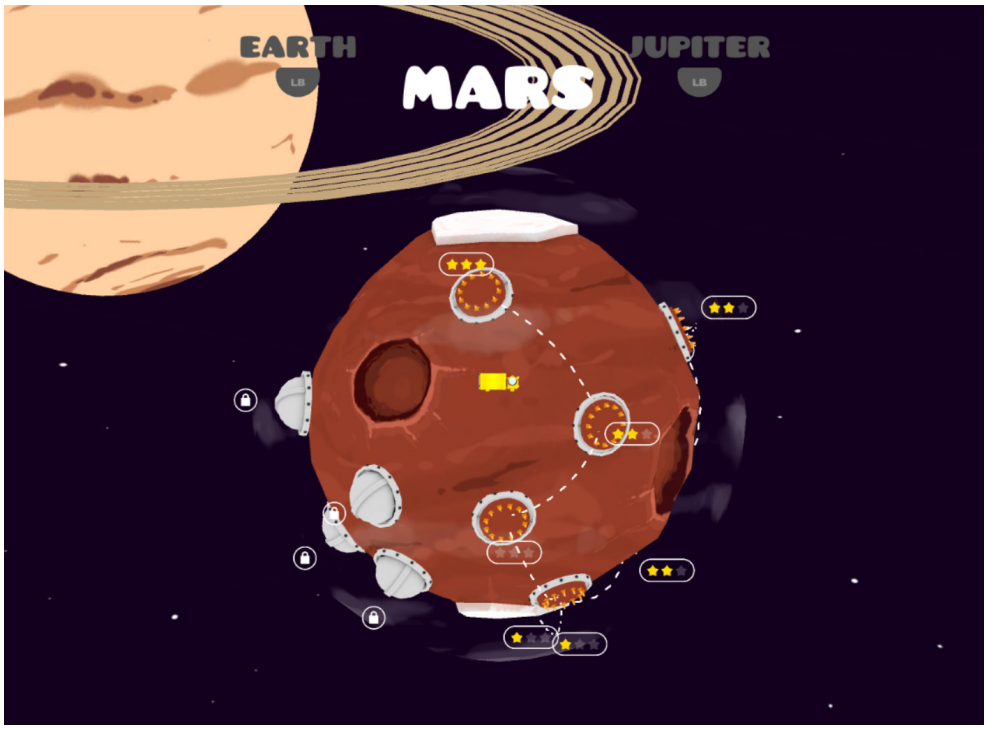
BONKIES

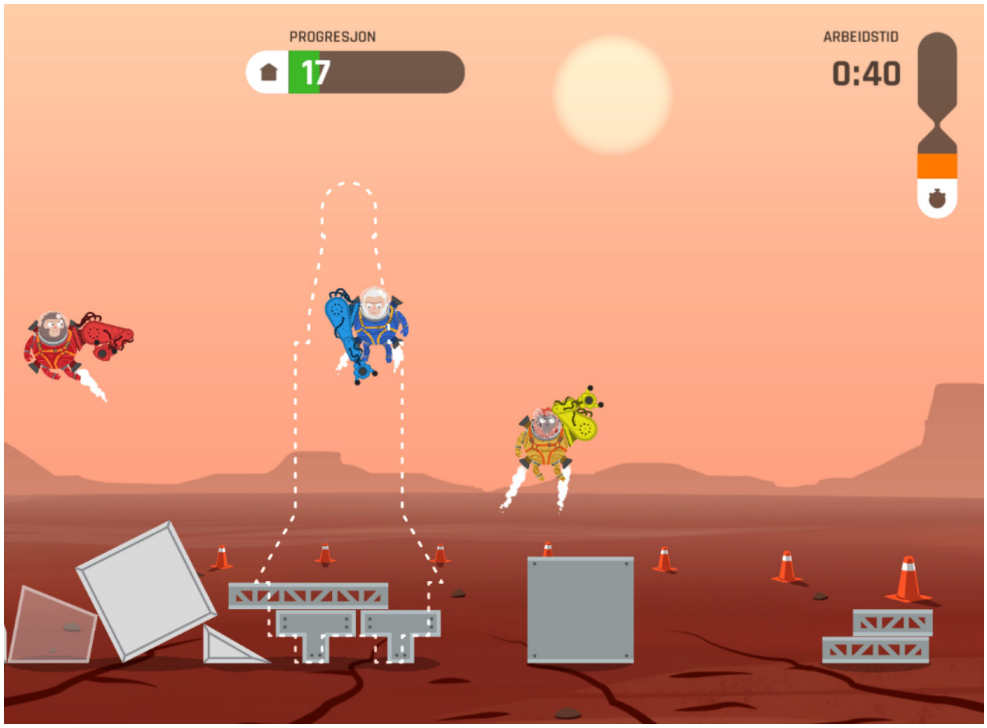
BESKRIVELSE

Bonkies er et spill for opp til fire spillere, hvor målet er å bygge en figur ved hjelp av tilgjengelige byggeklosser før tiden går ut. Omrisset av figuren som skal bygges er tegnet opp på skjermen, og de tilgjengelige brikkene som tilsammen utgjør figuren er plassert på bakken ved omrisset.

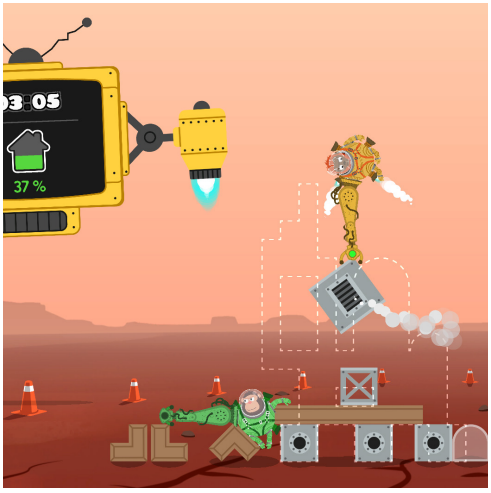
Hver spiller styrer en avatar utstyrt med jetpack og en robotarm. Robotarmen kan manøvreres til å peke i alle retninger rundt spilleren, samt gripe tak i klosser og andre spillere. Jetpacken kan flytte rundt på spilleren med en viss kraft. Jetpacken har også en boost-funksjon som gjør at spillerne kan fly raskere og løfte tyngre.

Spillet er fysikkbasert, så idet en spiller griper tak i noe annet, vil jetpacken måtte jobbe hardere. Spillere kan bli satt ut av spill dersom hjelmen deres blir truffet av en byggekloss med for mye bevegelsesenergi. Når tiden er ute får alle spillerne felles score basert på hvor presist, og hvor raskt de har bygget.

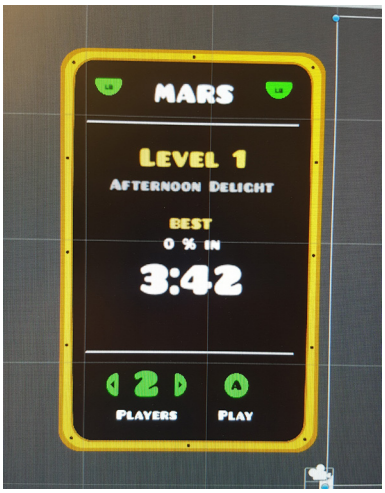




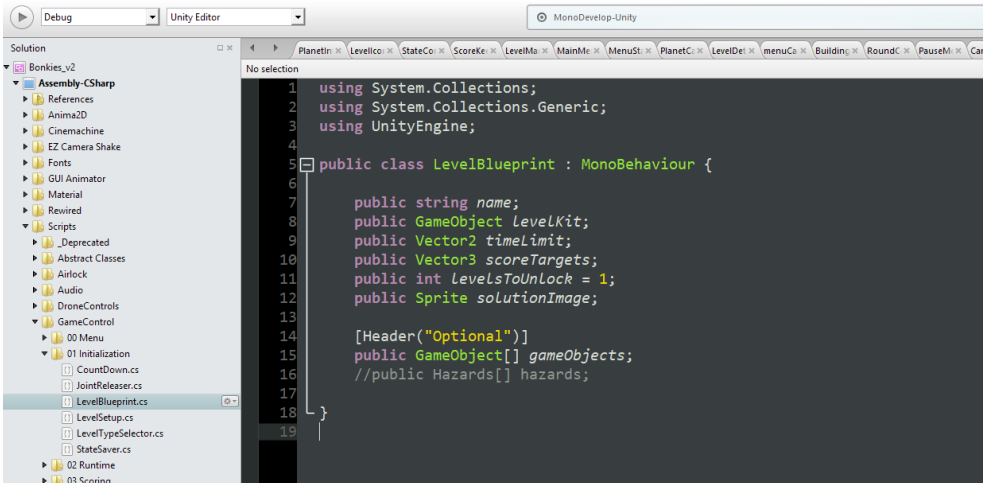
Skjermbilder fra Bonkies


3:05
37 %


PROGRESJON
17

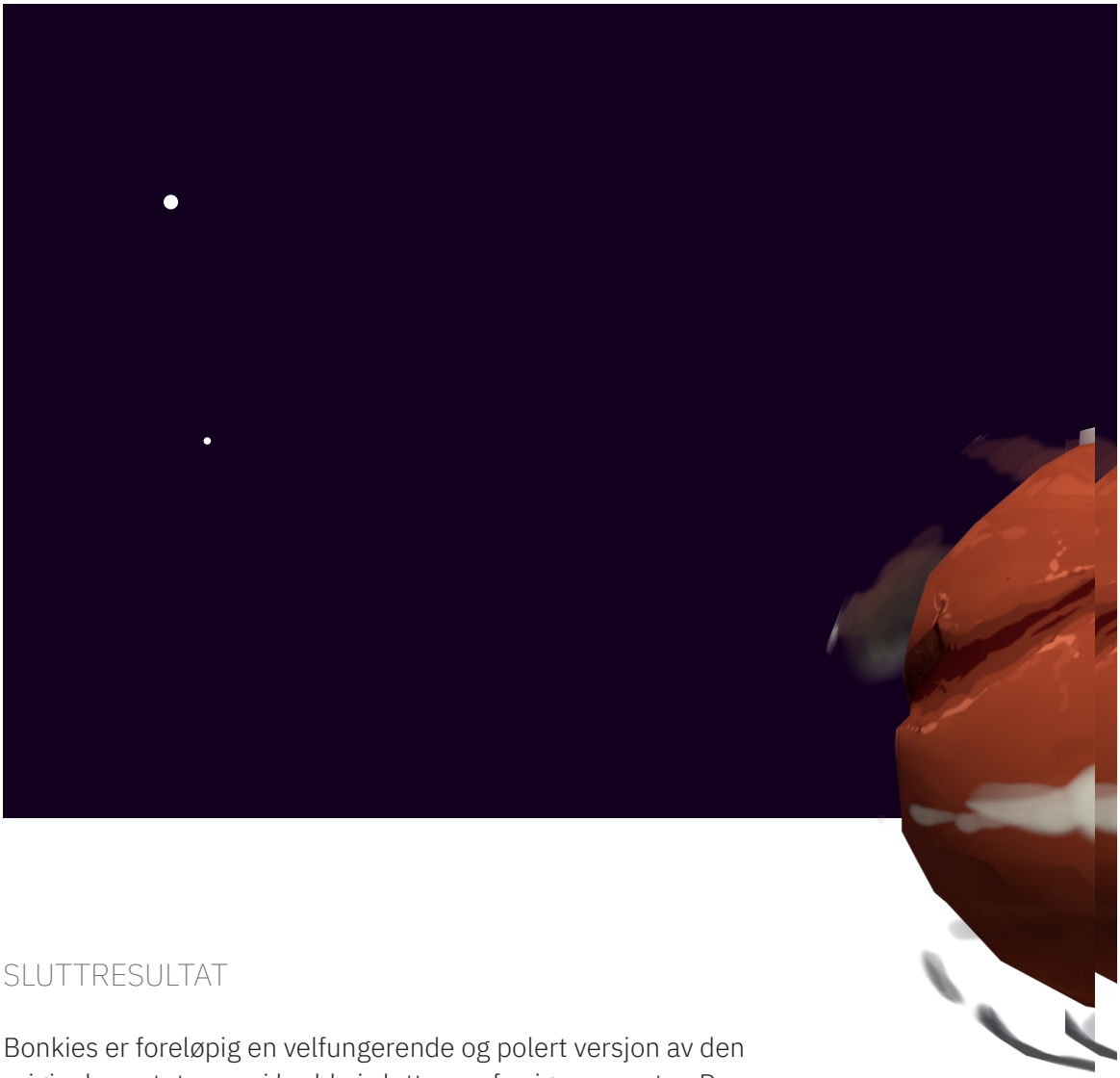

MARS
LEVEL 1
AFTERNOON DELIGHT
BEST
0 % IN
3:42
2
PLAYERS
PLAY


EARTH
MARS
3-1
THE CUBE
BEST 410
A
SIGN


Debug
Unity Editor
MonoDevelop-Unity
Solution
Bonkies_v2
Assembly-CSharp
References
Anima2D
Cinemachine
EZ Camera Shake
Fonts
GUI Animator
Material
Rewired
Scripts
_Deprecated
Abstract Classes
Airlock
Audio
DroneControls
GameControl
00 Menu
01 Initialization
CountDown.cs
JointReleaser.cs
LevelBlueprint.cs
LevelSetup.cs
LevelTypeSelector.cs
StateSaver.cs
02 Runtime
No selection
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
public class LevelBlueprint : MonoBehaviour {
public string name;
public GameObject levelKit;
public Vector2 timeLimit;
public Vector3 scoreTargets;
public int levelsToUnlock = 1;
public Sprite solutionImage;
[Header("Optional")]
public GameObject[] gameObjects;
//public Hazards[] hazards;
}

PROSESS

1. Vi hadde allerede jobbet med dette i et halvt år, og vunnet to priser for spillet.
2. Brukeropplevelsen ble vesentlig forbedret ved å fornye brukergrensesnittet og designe en helhetlig gameloop. Spillet fikk et underholdende menysystem, en lobby der spillerne fikk anledning til å personalisere avataren sin, pluss en oppsummeringsskjerm som viser en poengoppsummering etter at en bane er fullført. Baneprogresjon ble implementert og prestasjonen til spillerne på en gitt bane skulle bedømmes fra 0 til 3 stjerner.
3. Vi justerte kontrolloppsettet til å kunne fungere med Nintendo Switch sine Joy-Cons. En Joy-Con kan skilte med én joystick, fire knapper, to skulderknapper og et akselerometer.
4. Dessuten effektiviserte vi måten vi som utviklere kunne produsere innhold på. Vi lagde en level-builder og en player-builder, som kunne ta inn henholdsvis en level- eller player-blueprint, for så å bygge denne og putte den inn i spillet. På denne måten trengte vi bare én scene for selve spillet, i tillegg til hovedmeny-scenen. Tidligere lagde vi en scene per level, hvilket viste seg å være særdeles lite effektivt, og veldig kaotisk når det begynte å bli mengder av baner.
5. Vi søkte om tilgang til Steam workshop for å kunne legge ut spillet på Steam Early Access. Denne prosessen tok lengre tid enn antatt.



SLUTTRESULTAT

Bonkies er foreløpig en velfungerende og polert versjon av den originale prototypen vi hadde i slutten av forrige semester. Denne versjonen har en fungerende gameloop, med et underholdende menysystem, et baneprogresjonssystem og vi har lagt opp til at det skal være enkelt for oss å legge til mer innhold. Vi har sluppet en rekke bilder av utviklingen på instagramkontoen vår, og en GIF som viser apekatt-karakteren på Reddit. Spillet er nær å bli utgitt på Steam Early Access..

BEDRIFTSUTVIKLING

FORRETNINGSMODELL

Forretningsmodellen baserer seg på at vi produserer våre egne spill og distribuerer dem selv gjennom kanalene som de store plattformaktørene tilbyr, altså Steam, Playstation Store, Xbox Live Marketplace, Nintendo eShop, Google Play eller App Store. Kanalene vil avhenge av spillet. Målgruppen vil også variere fra spill til spill, men vi tenderer mot å lage familievennlige spill for sosiale grupper. Vi promoterer spillene våre via sosiale medier, og ønsker med dette å opprette en dialog mellom oss og kundene. Vi ønsker å være en åpen og ærlig utvikler som søker kontakt med brukerne sine. Foreløpig har vi ingen partnere utover Spark* NTNU som tilbyr veiledning i forbindelse med bedriftsutvikling. Prisstrategi vil også kunne variere fra spill til spill, men med tanke på den lille størrelsen på teamet vårt er det nok foreløpig mest hensiktsmessig at vi satser på å gi ut komplette spill for en engangssum, for å slippe å binde oss til vesentlig vedlikehold og videreutvikling av utgitte spill.

TEAMET

Vi er et lite team på to personer som verdsetter kreativ frihet og autonomi, samt deler et ideal om å være en ærlig og etisk spillutvikler. Vi har bred kompetanse, blant annet på felter som spilldesign, grensesnittdesign, UX, 2D-art, 3D-modellering, animasjon, visuelle effekter, programmering, lyddesign og musikkproduksjon.

Per nå er våre svakere sider i hovedsak, dømt etter tendensen vi har til å prioritere de ulike arbeidsoppgavene, lyddesign og musikk, markedsføring, publisering og strategisk bedriftsutvikling. Dersom vi utvider teamet burde vi nok vurdere personer som kunne hjulpet til på disse feltene, kanskje særlig business-biten.

Åpningsfest ved Gründerbrakka, NTNU

Alexander Jonassen (t.v.) og Christer Rebni utgjør Gauntlet Games

SPARK-MENTOR

Vi søkte Spark NTNU om å få en mentor for oppstarten vår. Under første møte med mentor ble vi oppfordret til fortgang i utviklingen, med mål om en så tidlig release som mulig for å begynne å samle brukerfeedback, samt bevise at folk er villige til å betale for produktet vårt. Han foreslo også å finne ut av hvorvidt vi ønsket å skaffe markedsførings-/økonomiansvarlig/CEO.

Gjennom mentoren i Spark fikk vi et tilbud om å søke om kontorplasser i Gründerbrakka, NTNU's nye kontortilbud for unge entreprenører ved NTNU. Hvorvidt vi får disse plassene er i skrivende stund uavklart.

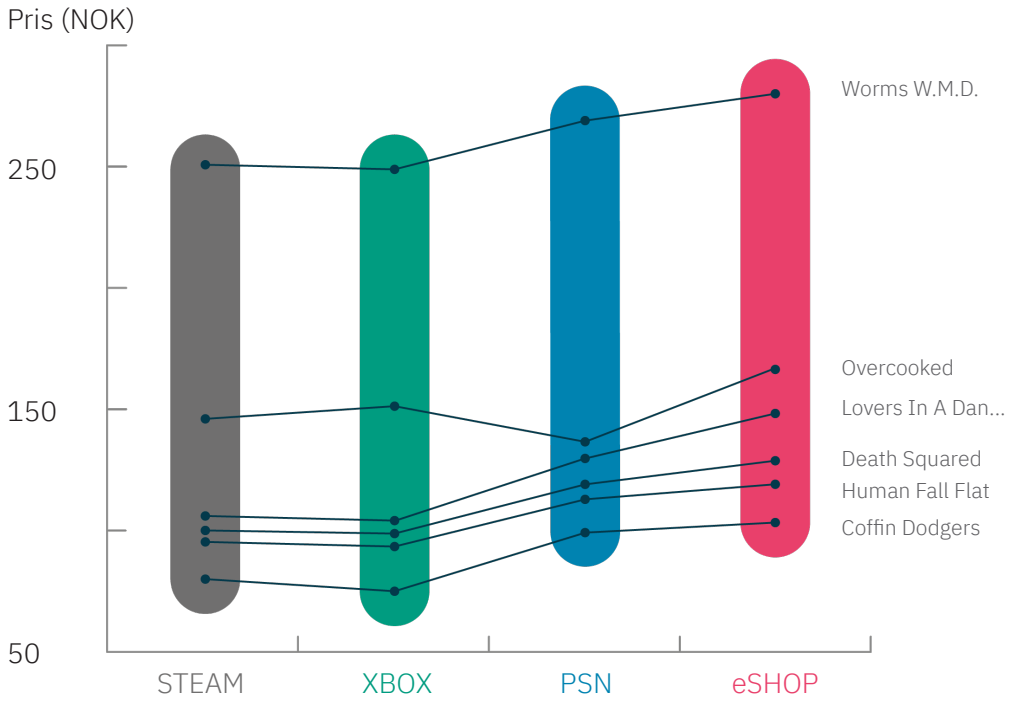
MARKEDSUNDERSØKELSE

Samarbeidsspill er en økende trend akkurat nå, og spillere er villige til å betale for disse opplevelsene (Clark, 2018). Vi er dog avhengige av at kjøper er interessert i og har anledning til å spille sammen med venner, og i så måte sikter vi oss inn på en smalere nisje av spillere. Dette er likevel spillere som etter alt å dømme vil være på utkikk etter noe nytt, da tilhengere av co-op-spill har en tendens til å hoppe raskt fra ett spill til det neste (Clark, 2018).

Prisnivået på ni spill med lokale flerspillermuligheter som vi har sjekket[1] ligger stort sett mellom mellom 80 og 140 NOK, med et snitt på 115 NOK. Prisene varierer også noe fra plattform til plattform, men tendensen er at gitt en pris x på Steam, så ligger prisen omtrent på x på Xbox Live Marketplace og x+25 NOK på Nintendo eShop og Playstation Store.

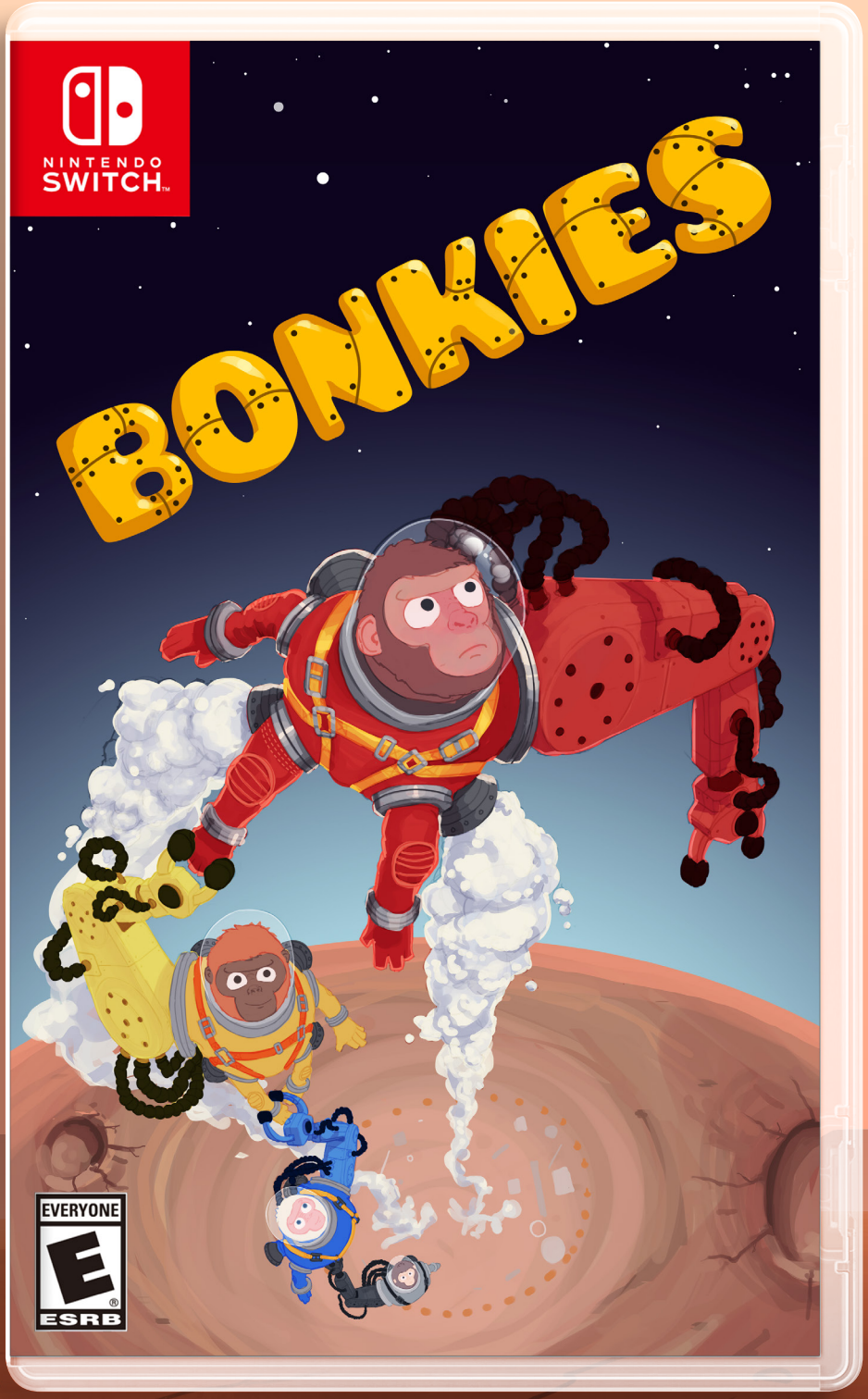
---

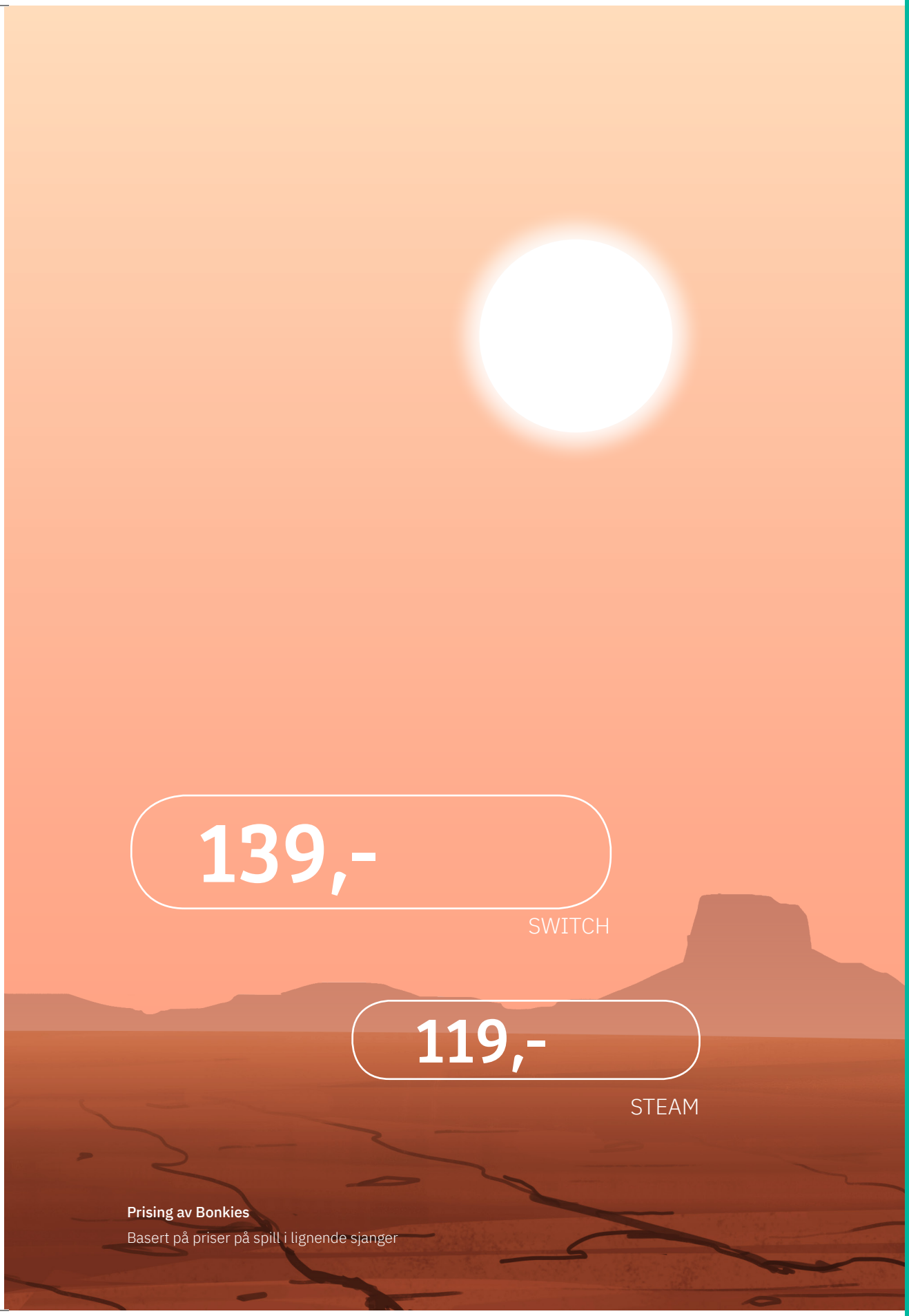
1 Lovers In A Dangerous Space Time (Asteroid Base, 2015), Overcooked (Ghost Town Games, 2016), Gang Beasts (Boneloaf, 2014), Human Fall Flat (No Breaks Games, 2016), Stickfight (Landfall West, 2017), Death Squared (SMG Studios, 2017), Tricky Towers (Weirdbeard, 2016), Worms W.M.D. (Team 17, 2016) og Coffin Dodgers (Milky Tea Studios, 2015)



Pristrend på tvers av spillplattformer

Priser på spill i lignende sjanger som Bonkies


NINTENDO
SWITCH
BONKIES
EVERYONE
E
ESRB



Prising av Bonkies

Basert på priser på spill i lignende sjanger

VISJON FOR BEDRIFTEN

Ambisjonen er å starte en levedyktig spillbedrift, hvilket krever at vi genererer nok inntekter til å kunne lønne oss selv i tillegg til andre personer vi tar med på teamet. Siden vi er ferske ut fra studiet har vi foreløpig ikke så store krav til årlig inntekt. Vi har beregnet at et overskudd på 400 000 vil være mer enn tilstrekkelig for å dekke teamets personlige utgifter for et år.

Visjonen for Gauntlet Games er å produsere spillopplevelser av høy kunstnerisk kvalitet, med nye vrier og som skaper glede hos spillere. Selskapet skal designe med et mål om å utvise sosialt ansvar, både med tanke på innhold som presenteres, budskap som kommuniseres, og kontoer som faktureres. Gauntlet Games skal være blant “the good-guys” i bransjen, som setter godt design over høy profitt, har full kreativ frihet innenfor sosialt ansvarlige rammer, og som er et veldig fett sted å jobbe.

Gauntlet Games

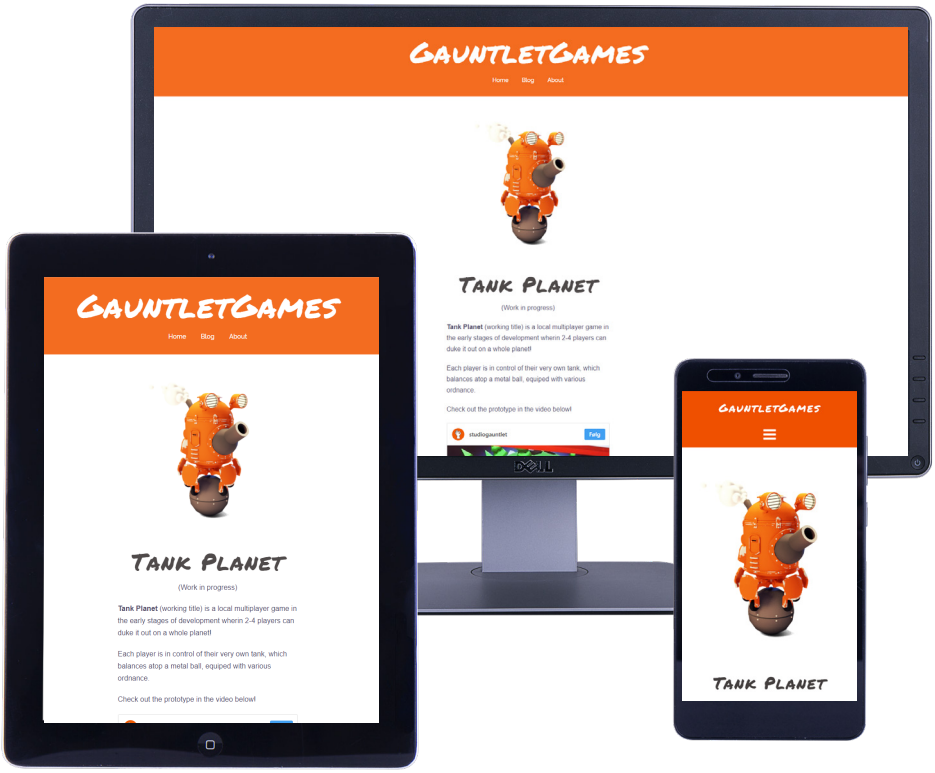
Gauntlet Games Logo og font

”Helmax, Egon!”, eventuelt ”It’s a goode game!” Det kan du velge selv.

MERKEVAREBYGGING

Både som et ledd i markedsføringen av spillene, og som en del av prosessen med å starte en bedrift har vi også jobbet med å bygget opp en merkevare rundt bedriften Gauntlet Games. Dette innebar at vi lagde en logo, fargepalett, nettside og opprettet kontoer på sosiale medier som Facebook og Instagram.

Logo og fargepalett ble designet med bedriftsvisjonen i tankene: Paletten er lys og leken, fonten som er valgt framstår som håndtegnet og logoen er humoristisk og litt useriøs i tillegg til at posituren til hånden lett gir assosiasjoner til ting som eksisterer i popkulturen fra før



Skjermdumper fra GauntletGames.net

Dynamisk nettsidelayout for bruk på alle enheter

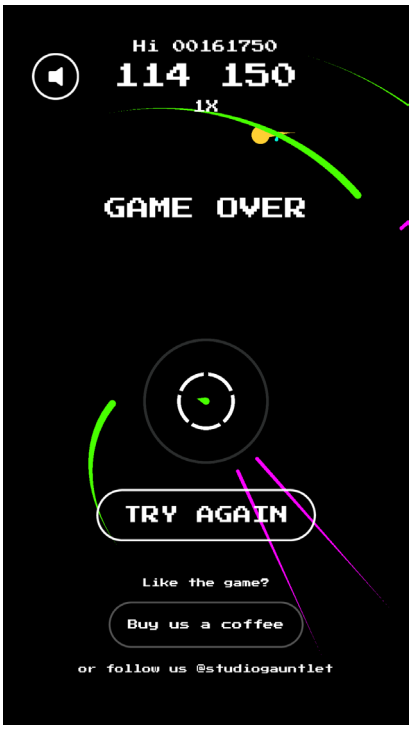
av. Alt er valgt for å bygge et image som et energisk, oppadstormende indie-spilldesignkontor.

Nettsiden ble laget hovedsakelig for å kunne sende nysgjerrige brukere fra Reddit til et sted de kunne vite mer om spillet på en enkel og konsis måte. Vi visste også at mange av disse brukerne leser Reddit på mobiltelefon, så nettsiden ble designet med visning på mobiltelefon i tankene.

REFLEKSJON

REFLEKSJON

I denne delen gjennomgås spillene nok en gang, men denne gangen med mer fokus på hva vi lærte fra dem og utviklingsprosessen. I tillegg går vi gjennom argumentasjonenen vår for hvorfor vi i løpet av masteroppgaven endret kurs, samt våre tanker om hva bedriften har blitt og skal bli.



BANGSHOT

Bangshot var første gang vi laget et spill med tanken om å faktisk publisere og kommersialisere det. Vi hadde bare én uke på oss (på grunn av at poleringen av Bonkies til NGA 2018 spiste opp en uke mer av masteren enn vi planla for) og fokuserte på å heller få et produkt som framstår ferdig, enn å lage alt helt perfekt. Mantraet var "If it works, it works (ingen skal gjennomgå koden vår uansett)".

Dette resulterte i et prosjekt som ble litt kaotisk å holde orden i mot slutten. Objektorganiseringen og skriptene var rotete, og det gjorde det utfordrende å f.eks. designe menyinteraksjoner og balansere vanskelighetsgrader. Det ble en del "hot-fixes", der vi løste problemer som oppstod nøyaktig der de oppstod. I praktsis betød det at vi skrev kodelinjer inn i skripts hvor de egentlig ikke hørte hjemme. Dette gjorde det vanskelig å holde oversikt over hvilke skript som egentlig kontrollerte hva. Plutselig kunne et menyskript påvirke livene til spilleren direkte, for eksempel.

Men spillet fungerte til slutt, det ble ganske underholdende og det så ganske polert ut. Dersom vi skulle videreutviklet det hadde vi nok gjort

oss selv en tjeneste ved å starte helt på nytt og planlegge bedre hvilken funksjonalitet vi vil ha inn før vi implementerte.

Publisering via Google Play sine verktøy var veldig enkelt, og man merket at Google har brukt mye tid på å strømlinjeforme prosessen for utviklere.

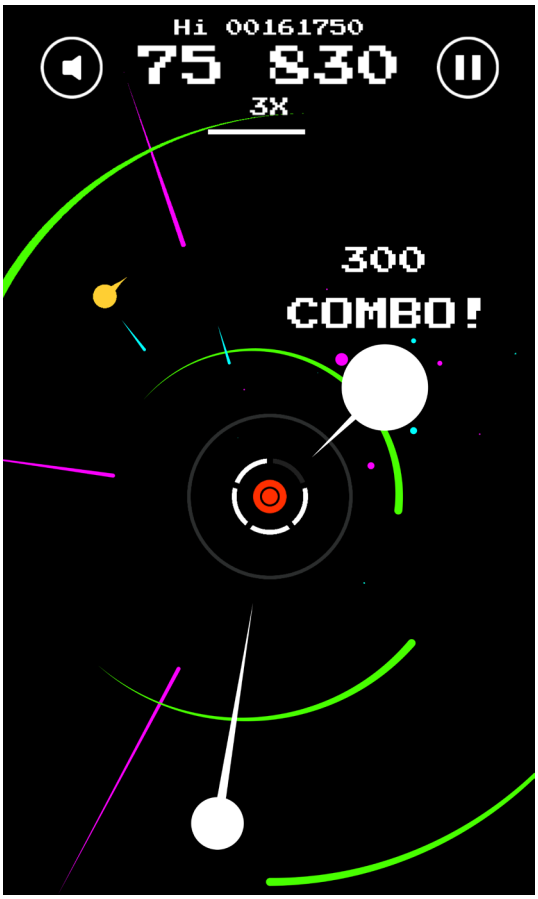
En utfordring vi ga oss selv var å kunne ta betalt for innhold i appen, såkalt in-app-purchases, rett og slett for å lære oss hvordan dette fungerte Dette viste seg også å være enkelt å implementere.

MARKEDSFØRING

Vi erfarte hvor mye tid det tar å produsere godt markedsføringsmateriale for et spill, og vi satte av en hel dag til dette, men kunne nok hatt enda mer tid. Vi valgte å lage diverse skjermbilder, en kort filmsnutt og en kort GIF, sistnevnte fordi vi har observert at det som oftest blir stemt opp på Reddit er korte, intense klipp som underholder brukeren i 5-10 sekunder.

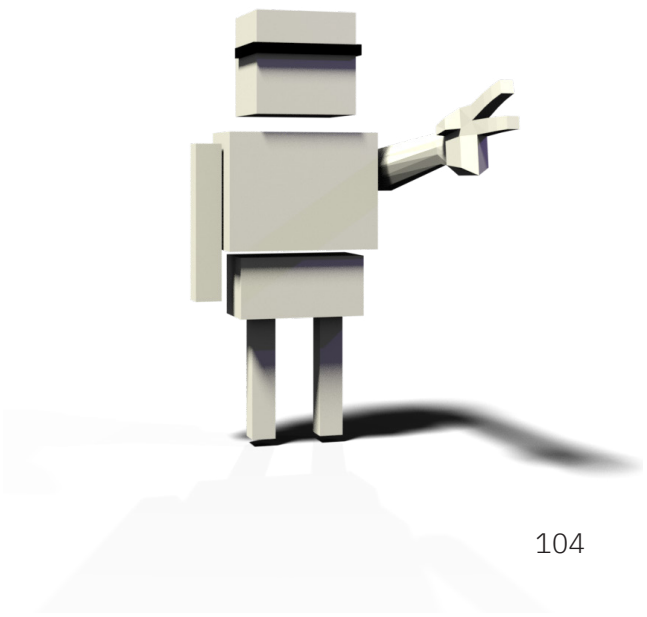
Skjermbilder fra trailer for BangShot

Vi valgte å demonstrere hvordan gameplayet ble mer og mer intenst



NESTE STEG

Etter tilbakemelding fra testere innser vi at spillet ikke insentiverer gjenspilling utover spillerens eget ønske om å slå sin egen high score. Med et progresjonssystem som f.eks. inkluderte opplåsing av nye egenskaper eller skins, hadde spillerne kunne blitt mer motiverte til å besøke spillet igjen. Dessuten har vi fått delte tilbakemeldinger på vanskelighetsgraden. Denne burde balanseres bedre, sannsynligvis gjøres dynamisk for å sørge for at nye spillere ikke mister motivasjonen av for høy vanskelighetsgrad, samtidig som veteraner ikke kjeder seg mens de spiller seg gjennom de første bølgene.

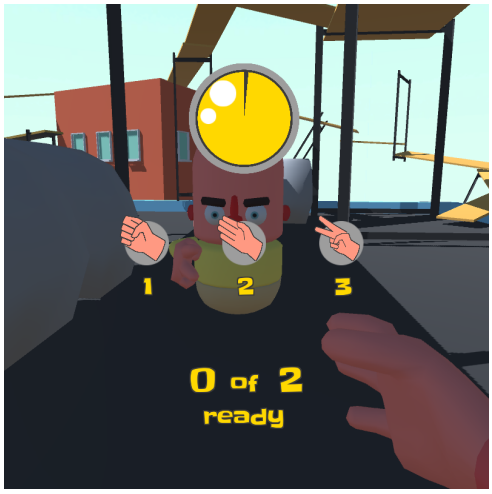


SUPER RPS

Av Super RPS lærte vi ettertykkelig at vi må sette realistiske mål for prosjektene, spesielt når vi bare har to personer og to uker til rådighet. Super RPS endte opp med å bli overambisiøst, med mange nye tekniske utfordringer noe som førte til et spill som ikke var like polert som BangShot.

For dette prosjektet hadde vi fortsatt en tanke om å utfordre oss selv på alle dimensjoner, og vi valgte å prøve oss på flerspiller over nettverk. Dette viste seg å by på mer hodebry enn i utgangspunktet forventet. Arbeidet ble komplisert ytterligere av at vi la inn detaljerte 3D-modeller med animasjoner. Disse animasjonene måtte synkroniseres over nettverket, i tillegg til spillernes posisjon og spillets tilstand.

Nettverksfunksjonaliteten er hentet rett ut av Unity. For å implementere de grunnleggende funksjonenen av denne trengte vi ikke gjøre stort annet enn å følge et par læringsvideoer på nett. I det vi skulle begynne å synkronisere spiller- og spilltilstander ble ting mer



To iterasjoner av Super R.P.S.

UI-et ble designet med en tidligere prototype som estetisk utgangspunkt

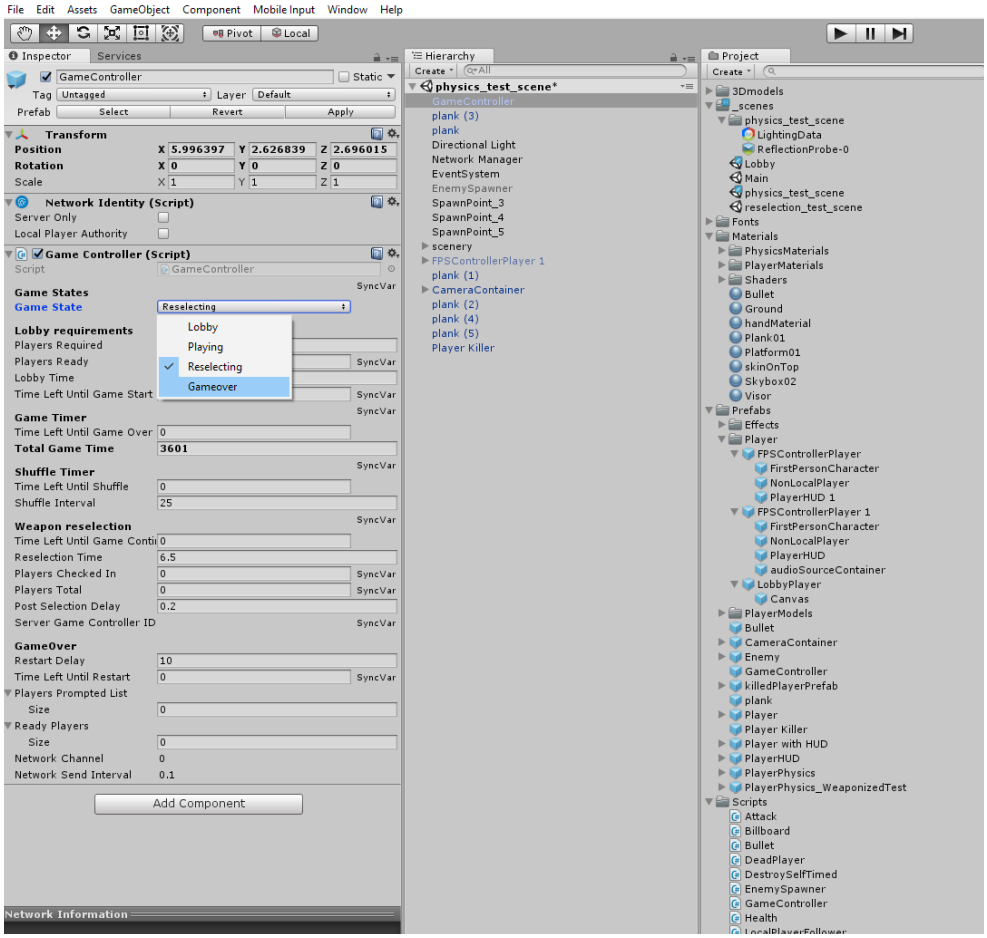
kompliserte, og server-klient-logikken måtte modnes ganske lenge før vi fikk alt til å fungere noenlunde slik vi ville. Dermed ble det lite tid til å f.eks. sprite opp det trauste lobby-grensesnittet som Unity leverer som standard, samt lite tid til å designe en game loop som oppsummerer spillet etter en viss tid og lar spillere få spille om igjen, dersom de ønsker, eller returnere til hovedmenyen.

UI-designet ble gjort mens spillet var i et lavoppløst prototype-stadie, og matchet estetikken i denne noe bedre enn den gjør i den endelige versjonen (se bilder på forrige side). Den endelige versjonen av spillet er mye mer lystig og tegneserieaktig. Denne diskrepansen i visuelt uttrykk bidrar også til at spillet fremstår som uferdig.

MARKEDSFØRING

Da de to ukene med utvikling var over var vi ikke særlig begeistret over resultatet. Det var hastig sydd sammen for i det hele tatt å få det spillbart, men spillet manglet en god game loop, og en helhetlig visuell stil. Vi delte noen skjermbilder fra spillet på facebook, og publiserte spillet på itch.io. Responsen var laber.

Vi satte et spørsmålstegn ved konseptet om å stadig lage helt nye spill, som ikke hadde noen form for sammenheng med foregående spill, ettersom det gjorde at vi måtte starte markedsføringen fra ingenting hver gang. Vi begynte å vurdere muligheten for å skape et univers som kunne være felles for de resterende spillene vi skulle lage, slik at markedsføringen fikk mer av en rød tråd gående gjennom seg, og dermed kunne skape mer interesse hos potensielle følgere. Eventuelt kunne vi markere oss som et studio som spesialiserer oss innenfor en sjanger spill. Sistnevnte tanke er den vi etterhvert har peilet oss inn på.



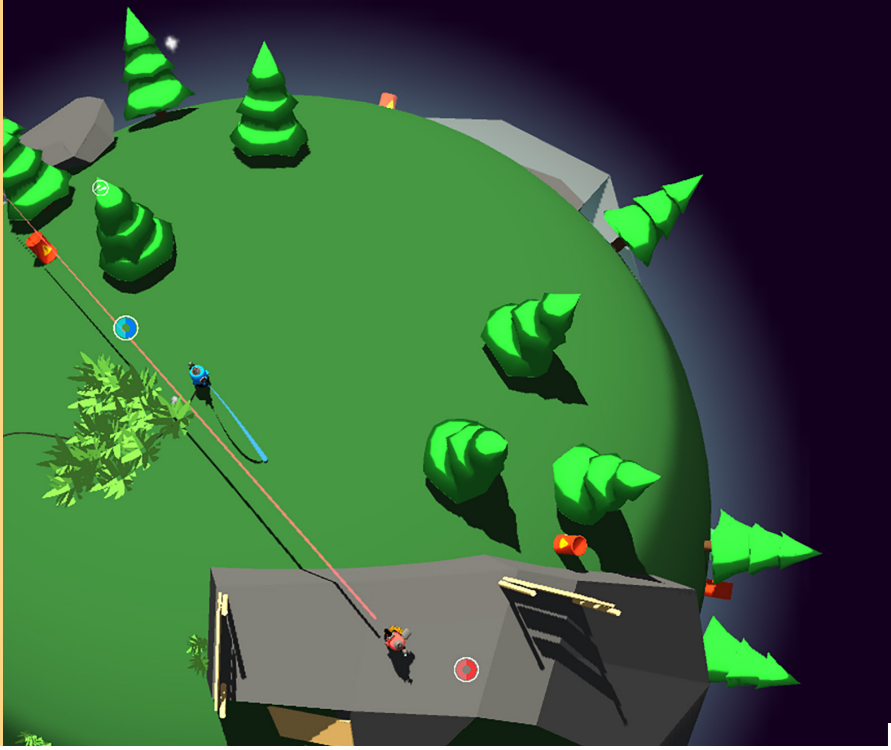
Skjermdump fra Unity

Spillkontroll-objektet ligger på serveren, og forer alle klientene med informasjon om spillets tilstand

NESTE STEG

Super RPS mangler innpakning. Hovedgameplayet fungerer ok, men spillet starter med en knotete og lite innbydende lobby, og avsluttes brått ved at tiden går ut og man blir sendt tilbake til lobbyen. Spillet mangler en god game loop.

Mye kunne vært gjort for å tilby enda mer variasjon og dybde, som f.eks. power-ups, spillerstatistikk og bedre arenaer. For oss var dette primært en øvelse i networking, og det blir nok med det.



TANK PLANET

For Tank Planet begynte vi å føle at vi måtte endre strategi, både når det gjaldt utviklingen av spillene, samt markedsføringen. Vi hadde tidligere utviklet samarbeidsspill for spillere på samme skjerm, og tenkte vi kunne utnytte den erfaringen ved å spesialisere oss på denne sjangeren. Vi prøvde å dele prosessen vår på sosiale medier litt hyppigere slik at folk faktisk fikk anledning til å følge progresjonen.

En av de mest nyttige tekniske lærdommene var hvordan man bygger et robust lokalt multiplayer kontroller system som tillater at folk dropper inn og ut dynamisk. Dette var lærdom vi fikk god nytte av senere, både i neste prosjekt og når vi tok opp tråden på Bonkies helt til slutt. Det var nok lurt å lære oss dette i et prosjekt som var helt ferskt, i stedet for å prøve å implementere direkte inn i Bonkies.

At spillerne bevegde seg på en sfære bød på en del utfordringer. For det første måtte vi simulere gravitasjon mot et punkt for alle fysikkobjekter. Unitys innebygde fysikksystem støttet kun gravitasjon i en gitt retning. For å få spillerne til å alltid stå oppreist normalt på planetens overflate brukte vi et PID-regulatorskript for å justere orienteringen deres. Ved å bruke de rette PID-verdiene fikk karakteren en veldig “wobbly” og sjarmerende mimikk. Det var veldig gøy å bruke teknologiske løsninger brukt i den virkelige verden for å løse problemer i en spillkontekst, og vi har blitt store tilhengere av logikken bak PID-regulatorer. Vi bruker også PID-regulatorlogikk for styring av robotarmene i Bonkies.

En annen stor utfordring i Tank Planet var å konvertere joystick-retning til en fornuftig sikteretning. Retningen karakteren skal sikte i må ta høyde for både spillerens nåværedne rotasjon, hvor på planeten spilleren står, og kameraets rotasjon. Det ble mye kvalifisert gjetning og eksperimentering med vektortransfoamsjoner, og på et tidspunkt så

```
1   using System.Collections;
2   using System.Collections.Generic;
3   using UnityEngine;
4
5   public class Upright : MonoBehaviour {
6
7       public bool separateAxes;
8
9       private Vector3 gravityVector;
10      private Rigidbody rb;
11      private float selfMass;
12
13      public Vector3 PIDValues;
14      private Vector3 PIDForce;
15
16      private float lastErrorAngle;
17      private float lastXErrorAngle, lastZErrorAngle;
18
19      private float angle;
20      private Vector3 axis;
21
22      void Start () {
23          rb = GetComponent<Rigidbody> ();
24          selfMass = rb.mass;
25      }
26
27      void FixedUpdate () {
28          KeepUpright ();
29      }
30
31      void KeepUpright () {
32          gravityVector = (transform.position - Vector3.zero).normalized;
33          if (!separateAxes) {
34              Quaternion deltaRotation = Quaternion.FromToRotation (transform.up, gravityVector);
35
36              deltaRotation.ToAngleAxis (out angle, out axis);
37              rb.AddTorque (axis * PID (angle) * selfMass);
38              lastErrorAngle = angle;
39          } else {
40
41              float xErrorAngle = Vector3.SignedAngle (transform.up, gravityVector, rb.transform.right);
42              float zErrorAngle = Vector3.SignedAngle (transform.up, gravityVector, rb.transform.forward);
43
44              float xTorque = PID (xErrorAngle);
45              float zTorque = PID (zErrorAngle);
46
47              lastXErrorAngle = xErrorAngle;
48              lastZErrorAngle = zErrorAngle;
49
50              rb.AddTorque (transform.right * xTorque * selfMass);
51              rb.AddTorque (transform.forward * zTorque * selfMass);
52          }
53
54
55      }
56
57      float PID (float errorAngle) {
58          PIDForce.x = errorAngle;
59          PIDForce.y = PIDForce.y + errorAngle* Time.deltaTime;
60          PIDForce.z = (errorAngle - lastErrorAngle) / Time.deltaTime;
61
62          return PIDForce.x * PIDValues.x + PIDForce.y * PIDValues.y + PIDForce.z * PIDValues.z;
63      }
64  }
65
```

Skjermdump fra MonoDevelop

Eksempel på skript som benytter PID-reguleringsprinsippet (brukt i Tank Planet)

det ut til å fungere godt nok. Da sa vi oss foreløpig fornøyde. Fortsatt oppfører siktet seg litt hakkete siden det konstant kompenserer for kamerarotasjon, men det kan også ha noe med at PID-verdiene som holder spilleren oppreist er dårlig kalibrert, og at spilleren egentlig står og rister og dermed tukler med siktelinjen.

Siden alle spillerne var på samme skjerm måtte kameraet kunne holde oversikt over alle spillerne og justere seg selv slik at alle til enhver tid var synlige. Å få denne trackingen til fungere uten at kameraet av og til spant ut av kontroll var en utfordring, og fortsatt kan kameraet oppfører seg utilsiktet dersom en spiller forsvinner bak horisonten på planeten.

I Tank Planet gikk vi for å knytte UI-elementene tett opp mot spillerne, med en glyf som kombinerte all informasjon vi mente var relevant for spilleren under gameplay, inspirert av staminaringen i Zelda: Breath of the Wild. Glyfen fulgte etter spilleren på skjermen, men plasserte seg alltid mot utkanten av skjermen, for ikke å komme i veien for spillere eller gameplayet. Antall resterende liv anså vi ikke som like kritisk informasjon å ha tilgjengelig ved spilleren, så den plasserte

Skjermbilde fra Tank Planet

Hver spiller har en glys som følger etter dem, men som prøver å holde seg ute av veien.

vi i hjørnene av skjermen. Det viste seg at glyfen var lite lesbar, en kombinasjon av at den var for liten, hadde for dårlig kontrast med bakgrunnen og, sine egne bestanddeler, samt en veldig abstrahert symbolikk.

UI-element fra Tank Planet

Glyf som viser gjenværende ammunisjon (ytre ring), stamina (sektor t.v.) og helse (sektor t.h.)

MARKEDSFØRING

Markedsføringen ble denne gangen gjort annerledes: I stedet for å vente med markedsføring før helt mot slutten når spillet nærmer seg ferdig la vi en plan om å dele ting fra utviklingen som vi trodde kunne slå ann på sosiale medier tidlig. I tillegg slapp vi ikke noen form for spillbar demo til slutt, i stedet satset vi på å legge ut skjermbilder og en video av gameplay som viste fram spillet på en spennende måte.

Resultatet av den nye taktikken var at en render av en karakter i spillet oppnådde 6.6k visninger samt 852 upvotes på /r/Blender subredditen, noe som vi anså som en bekreftelse på at folk responderer godt på små biter av markedsføring, hvertfall på et sosialt medie som Reddit. Denne suksessen gjorde også at vi fortløpende laget en nettside for å ha et sted å henvise personer som var nysgjerrige på å vite mer, samt ønsket å følge prosjektet videre.

NESTE STEG

Også Tank Planet mangler en ordentlig game loop. Det er for eksempel ingen meny som leder inn til gameplayet, og det er ingen tydelig vunnet tilstand.

Under testing så vi at folk slet med å treffe hverandre, særlig med standardvåpnet, og at det frustrerte dem. Det er mulig spillet hadde føltes bedre dersom man innførte at ett skudd alltid var nok til å ta livet av spillerne, og balansere våpnene opp mot dette premisset. Det ville også fjernet behovet for den lille og lite lesbare livmåleren som følger etter hver spiller. Denne kunne da heller kun vise energi- og ammunisjonstilstand.

Deretter burde det legges til innhold som flere baner og flere våpenvarianter for mer variert gameplay, samt flere karakterklasser, både for å tilføre spillet mer dybde, men også som et markedsføringsgrep.

Vi tror også at det hadde vært en gøy mekanikk at hele planeten er ødeleggbar. For eksempel kunne planeten vært delt inn i heksagonale felter (som en slags fotball) som kunne ødelegges og avdekke lava, vann eller lignende farer. Dette ville gjort bevegelsesareal til en begrenset ressurs, og det ville plutselig gitt mer mening at spillet foregår på en kule med begrenset areal.

Y.H.T.T.M.I.Y.H.

Med lærdommen fra forrige prosjekt begynte vi arbeidet på konseptet som skulle ende opp som Y.H.T.T.M.I.Y.H. I løpet av dette prosjektet begynte vi å føle veldig på ønsket om å jobbe mot en virkelig utgivelse av et spill, i stedet for å konstant lage prototyper. Den generelle lærdommen fra dette var nok at vi det holder ikke å komme opp med et godt core-gameplay, så lenge innpakningen ikke er til stede.

Vi opplevde også det som popullært kalles “feature creep”. Etter at grunnmekanikkene var implementert begynte vi fort å legge til fler og fler features. Mye av dette kom av at det var et reellt behov for å gi dronespilleren mulighet til å kunne skille mennesker fra androider, samt motivere spillere til å ikke bare gjemme seg bak en vegg og bli der. Å balansere asymmetrisk gameplay er komplisert.

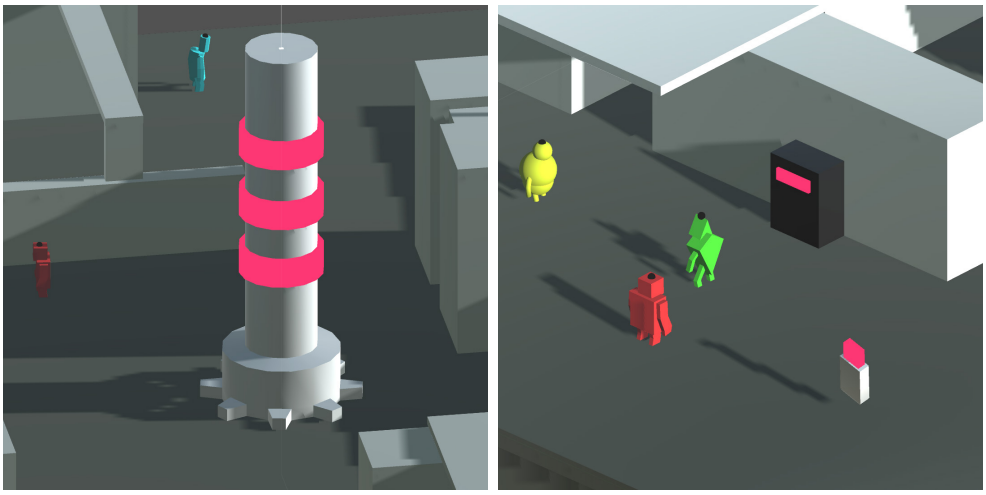
Vi eksperimenterte med plasseringen av objekter som spillerne måtte plukke opp, samt hvor disse objektene måtte leveres. Først måtte de leveres ved tårnet, men da ble det for lett for dronen å bare patruljere akkurat rundt tårnet. Derfor plasserete vi heller ut åtte terminaler slik at minst to to alltid ville vende mot kamera. Mens dette gjorde det lettere for spillerne å levere databrikker usett, ble det nok litt for vanskelig for dronespilleren å beholde en god oversikt over hva som foregikk på brettet. Vi prøvde så med seks terminaler, men antallet bør nok reduseres ytterligere.

Dronen har anledning til å markere spillere, en mekanikk som enten må fjernes eller implementeres på en bedre måte, i og med at mennesker spawner i skjul, mens androider er synlige på kartet fra

starten av. Dronen kan altså i nåværende versjon utelukke alle synlige spillere i det spillet starter som kandidater for menneske.

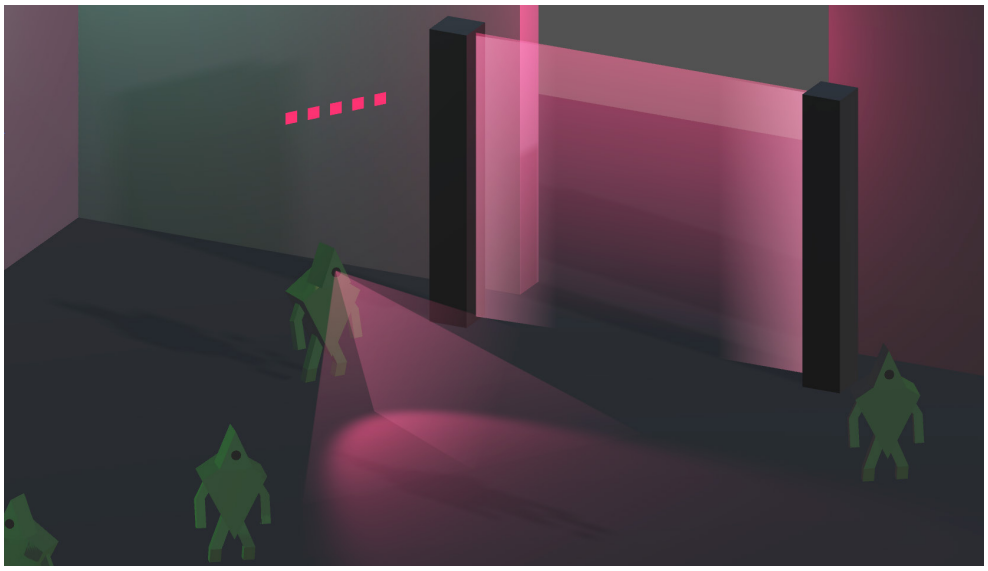
På et tidspunkt kunne dronen plassere ut portaler som kunne detektere androider (grønt blink) og menneskelige spillere (rødt blink). Androiden kunne i tillegg kommandere avatarer til å gå til et punkt (f.eks. gå gjennom en portal). Dronen kunne også på et tidspunkt konvertere en avatar til en vandrende scanner for en viss periode. Den blinket rødt dersom en spiller kom innenfor synsfeltet dens. Dersom en menneskelig spiller ble konvertert til en scanner, kunne spilleren selv velge å blinke rødt når hen selv ville.

Dette ble mange og forholdsvis kompliserte mekanikker, og under testing virket det som at spillere ble litt overveldet. Vi bestemte oss derfor for å holde mer fokus på kjerne-gameplayet, men tiden begynte å bli knapp, og spillet ble aldri balansert godt nok. Det er fortsatt for vanskelig å være drone.



Skjermbilder fra Y.H.T.T.M.I.Y.H.

For å bekjempe dronen må spillere plukke opp minnepinner, og levere dem ved terminaler.for å tappe dronetårnet for liv



Bio-Scannere fra en iterasjon av Y.H.T.T.M.I.Y.H.

Veggen/inspektøren blinker rødt eller grønt dersom en spiller eller androide passer gjennom/foran den.

I begynnelsen brukte vi en god del punktlys i dette spillet, for å simulere lyktestolper, men vi lærte at å ha mange piksellys som virker på samme objekt (og vi hadde mange objekter i form av androider som gikk rundt omkring) ble krevende for Unitys innebygde grafikkmotor. Derfor reduserte vi antall lys i scenen, og satset mer på å oppnå den korrekte stemningen med ambient lighting.

MARKEDSFØRING

Markedsføringen av dette spillet ble til slutt gjort i tråd med det vi hadde lært fra foregående spill, og vi endte opp med å legge ut en GIF som presenterte essensen av gameplayet på en forenklet og lettfattelig måte på /r/Unity3D subredditen. Dette er hovedsubredditen for Unity-utviklere og entusiaster på Reddit. Dette fikk god respons, og valget med å holde det vi viste fram på prototypingsnivå gjorde at folk lettere kunne bidra med tilbakemeldinger og kritikk. Den egentlige versjonen av spillet på dette tidspunktet hadde veldig mange ekstra features som

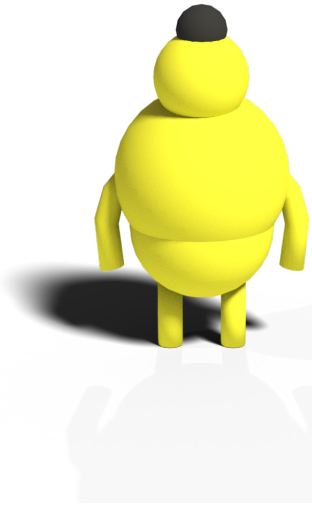
gjorde det vanskelig å formidle hva grunnkonseptet egentlig var, men en videosnutt av noe som ligner på det endelige gameplayet ble postet på instagram, og også der ble folk nysgjerrige på konseptet.

NESTE STEG

Selv om testere og publikum ser ut til å like grunnkonseptet godt, synes de fortsatt det er vanskelig å avsløre hvem som er mennesker når de spiller som dronen. Dronen må få gode, men rettferdige verktøy til disposisjon for å kunne luke ut mennesker.

Testerne liker å snike seg bak vegger, men for at dronen skal kunne ha sjans å avsløre dem bør det være en balanse mellom insentivene for å snike seg bak vegger og å bevege seg åpenlyst. For eksempel kunne dronen aktivert en type midlertidig røntgensyn hvor mennesker bak vegger lyser opp, mens mennesker ute i åpenhet forblir anonyme, for å gjøre det å stå bak vegger mindre sikkert. Dersom det er offentlig informasjon hvorvidt dronen har røntgensynet tilgjengelig eller ikke ville det skapt to helt ulike dynamikker hos menneskespillerne. I det ene tilfellet vil spillerne trekke mot dekning av vegger. I det andre vil de måtte unngå vegger for alt det er verdt.

Y.H.T.T.M.I.Y.H. mangler også en god gameloop og startskjerm til å ramme inn spillopplevelsen.



ENDRING AV KURS

Etter at vi var ferdige med Super RPS vant vi to priser under NGA for spillet vi hadde utviklet semesteret før: Bonkies. Allerede da vurderte vi å endre formatet på masteren vår, og heller fokusere alle krefter på Bonkies. Etter samtale med veileder kom vi fram til at vi burde nok fortsette på den originale planen vår, men være mer bevisste på hvem målgruppen for spillene vi lager er og dermed hvordan vi kan markedsføre bedriften vår på best mulig måte. Vi bestemte oss for å spesialisere oss inn på et marked i stedet for å prøve oss på alt mulig. Vi liker begge lokale coop-spill og tenkte nå at det hadde vært interessant å prøve å bli et kjent studio innen denne nisjen.

Etter å ha brukt fire uker på å utvikle to nye prototyper (Tank Planet og Y.H.T.T.M.I.Y.H.) så vi oss nødt til å gjøre nok en vurdering av retningen på masteroppgaven og hvorvidt vi burde fortsette på samme kurs:

HVORFOR SATSE PÅ BARE ETT SPILL

Ved å fokusere på ett prosjekt over lengre tid ville vi kunne produsere innhold til spillet utover grunnleggende aktiviteter, samt designe gode game loops. I tillegg ville det bli enklere å markedsføre spillet mer kontinuerlig og svare på feedback fra brukere og gjøre endringer basert på denne feedbacken.

Spillkonseptene vi hadde til nå mente vi at hadde kommersielt potensiale. Tre av konseptene (Tank Planet, YHTTMIYH i tillegg til Bonkies) hadde greid å vekke interesse hos spillere.

I tillegg begynte vi kjenne oss utbrente på å konstant måtte starte fra ingenting på hvert prosjekt, og å legge konsepter som hadde begynt å få interesse fra brukere på Reddit og Instagram på is.

[–] **Luuno** 4 poeng 2 måneder siden
Would love to see the progress of this concept!
permalenke embed lagre rapporter gi gull svar

[–] **Joelspeanuts** [S] 5 poeng 2 måneder siden
And we would love to keep you posted!
You could check out our website or our Instagram if you're interested in seeing some behind the scenes stuff from this project :)
permalenke embed lagre foreldre rediger Deaktiver innbokssvar slett svar

Kommentar på Y.H.T.T.M.I.Y.H. fra Reddit

Eksempel på kommentar som gir lyst til videreutvikle et konsept, ikke starte fra scratch på et nytt et.

HVORFOR FORTSETTE Å UTVIKLE NYE SPILL

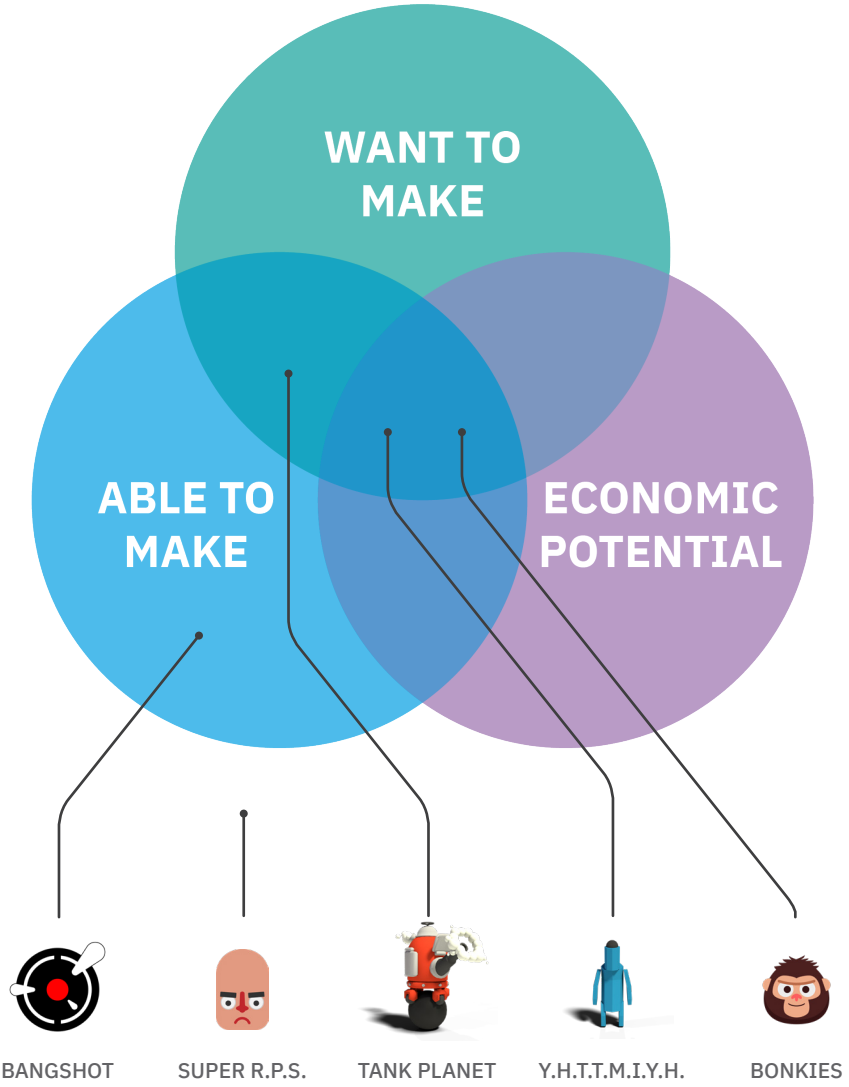
Ved å fortsette som planlagt kunne vi utforsket flere spillkonsepter, -mekanikker og -teknologier. F.eks. hadde vi lyst å utforske VR og dypere historieformidling. Masteroppgaven sa også ganske tydelig at det skulle produseres syv spill.

KONKLUSJON

Utbyttet fra å fortsette å lage nye spill var ikke lengre like tydelig. Målet var å lage ferdige spill som ble publisert, men vi endte bare opp med uferdige prototyper. Så vi fikk trening i å lage prototyper av spill, men hvis bedriften skal ha en framtid måtte vi lære å fullføre og tjene penger på dem.

Det ultimate målet var alltid å opprette et levedyktig spillstudio. Nå kunne det se ut som at det å produsere syv spillprototyper ikke var den mest hensiktsmessige måten å oppnå dette på, og derfor så vi på det som uproblematisk å gå bort fra antagelsene vi hadde gjort i oppgaveteksten.

Listen, kombinert med nye impulser i form av råd fra veileder på Spark* NTNU og Ryan Clark (2018) sin video om hooks, gjorde at vi bestemte



Fasability-analyse

Vurdering av spillkonseptene våre, for å hjelpe oss å finne ut hvilket vi skulle satse videre på.

oss for å endre på hva vi egentlig jobbet mot: Å lansere ett spill. Målet ble dermed etter å ha vurdert alle spillene (se neste side) å få Bonkies ferdig nok til å kunne slippes til et betalingsvillig publikum.

VALG AV SPILLKONSEPT

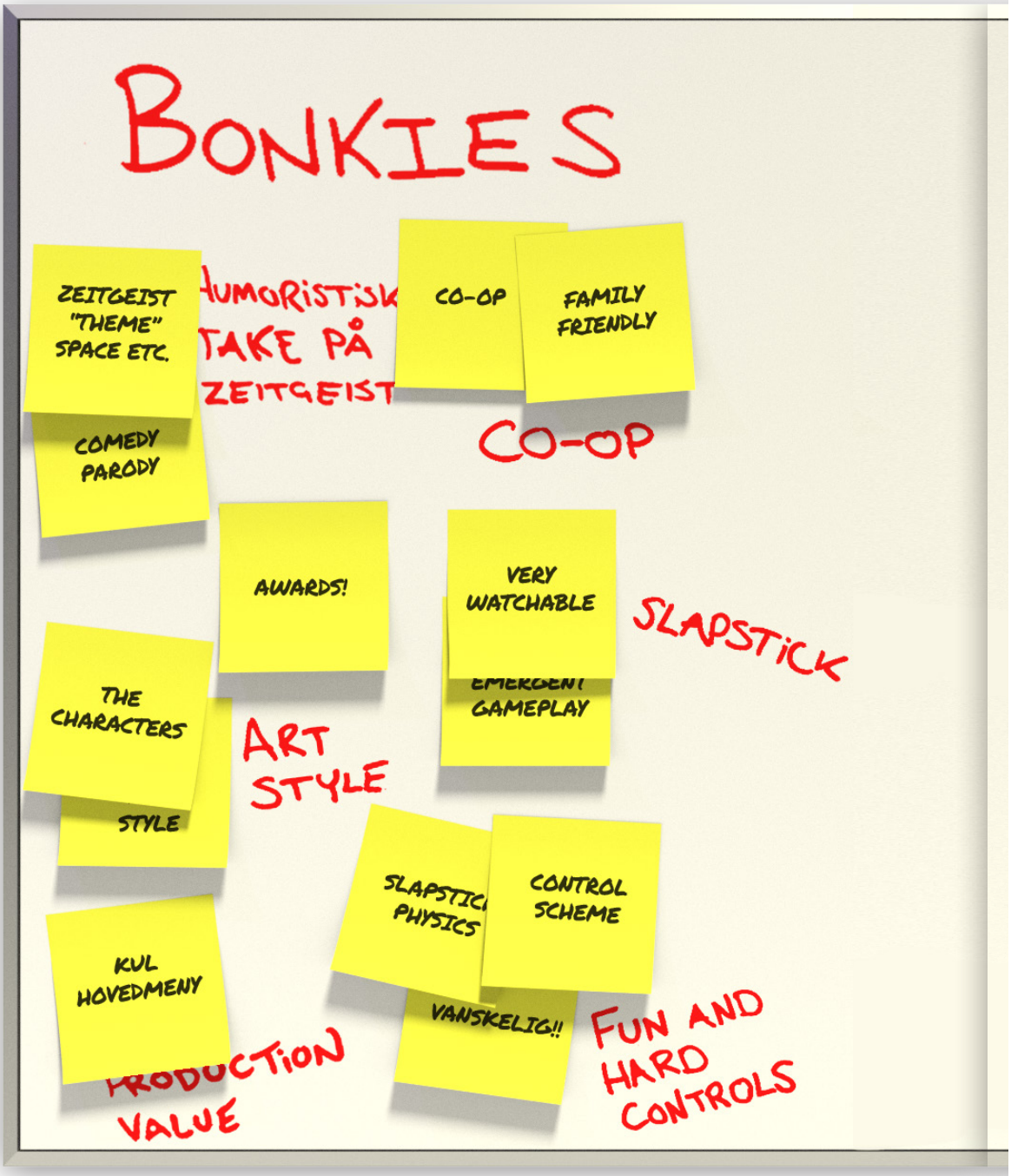
For å avgjøre hvilket spill vi skylle gå videre med gikk vi gjennom alle spillene og vurderte hvorvidt det var interessant å gå videre med det.

På dette tidspunktet vurderte vi BangShot som urealistisk å tjene penger på uten massiv markedsføringsinnsats i og med at det var et mobilspill. Super RPS ville kreve for mye teknisk krevende jobb til at vi kunne realisert det, og sett at vi hadde det, så ville det vært vanskelig å tjene penger på det uten en massiv markedsføringsinnsats, i og med at det krever en base av onlinespillere. Vi vurderte Tank Planet i sin daværende form som såpass enkel rent spillmekanisk, at markedet ikke ville tenke det var interessant nok, til tross for planet-gimmicken. I retrospekt anser vi denne gimmicken som en særdeles sterk hook og et utgangspunkt for flere gøye mekanikker.

Y.H.T.T.M.I.Y.H og Bonkies anså vi som gode kandidater, da de begge hadde mottatt god feedback fra spillere og hadde en del hooks vi anså som mulige å utnytte i fremtidig markedsføringsarbeid (se neste side).

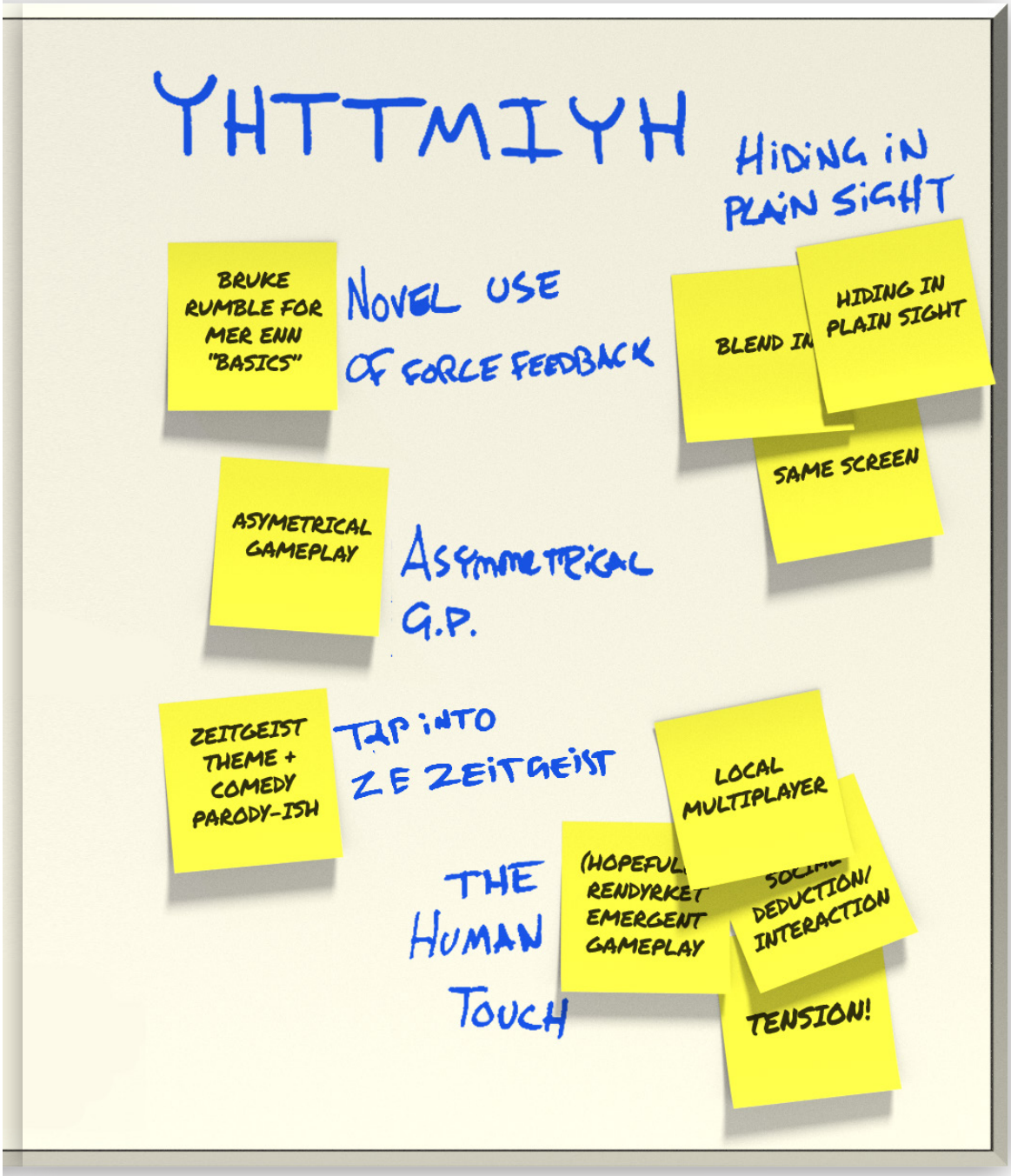
Etter en måned gjorde vi en vurdering av helomvendingen, og begge følte at det var riktig valg, både for følelsen av progresjon med selve foretaket Gauntlet Games, men også for å få erfaring med flere vitale aspekter ved utvikling av et spill, som implimentering av progresjonssystemer, dypere menysystemer, utgivelse på Steam og polering av gameplay .

Likevel har utviklingen av prototypene vært en meget verdifull erfaring, og kunnskapen vi opparbeidet oss gjennom det arbeidet ville vi ikke fått dersom vi hadde holdt oss til Bonkies-prosjektet fra starten av. Denne erfaringen har gjort at vi føler at vi selvsikkert kan begi oss ut på nye og mer ambisiøse prosjekter senere.



Identifisering av hooks

Clustring av styrker ved spillene, i et forsøk på å definere potensielle *hooks* (Clark, 2018).


YHTTMIYH
HIDING IN PLAIN SIGHT
BRUKE RUMBLE FOR MER ENN "BASICS"
NOVEL USE OF FORCE FEEDBACK
BLEND IN
HIDING IN PLAIN SIGHT
SAME SCREEN
ASYMETRICAL GAMEPLAY
ASYMMETRICAL G.P.
ZEITGEIST THEME + COMEDY PARODY-ISH
TAP INTO ZE ZEITGEIST
LOCAL MULTIPLAYER
THE HUMAN TOUCH
(HOPEFUL
RENDYRKET
EMERGENT GAMEPLAY
DEDUCTION/ INTERACTION
TENSION!

BONKIES

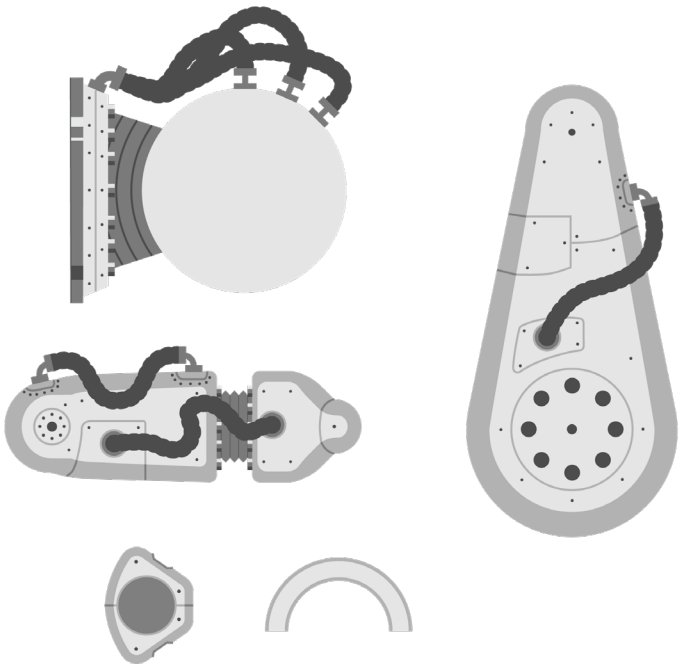
Under videreutviklingen av Bonkies innså vi nødvendigheten av å tilrettelegge for å kunne produsere relativt store mengder innhold uten at dette førte til merarbeid dersom en større, overdordnet endring måtte gjøres. Til nå hadde alle spillerkarakterene og baner vært bygd på duplikater av en basiskarakter og -bane. Dette hadde til nå betydd at dersom en skulle gjøre endringer på karakteren, måtte samme endring gjøres fire ganger (siden vi hadde fire spillere). La vi inn ti nye baner måtte eventuelle baneendringer utføres ti ganger til. Dette var ikke holdbart i lengden med tanke på at et spill som Bonkies er avhengig av å ha en del levels. Derfor lagde vi en karakterbygger som kunne fores med en oppskrift på en player samt en puzzle-bygger som kunne fores en oppskrift på en puzzle, og henholdsvis spytte ut en karakter og en puzzle. På den måten klarte vi oss med én karakter-master og én bane-master. Dette gjorde at vi mye lettere også kunne implementere en meny som lot spillerne personliggjøre karakterene sine med ulike apehoder og farger på draktene.

For å gradvis introdusere spilleren for mekanikkene i spillet samt gi en følelse av mestring og progresjon implementerte vi et baneprogresjonssystem. Dette innebar å sjekke om spillerne på slutten av en bane hadde oppnådd høy nok poengsum til å låse opp neste bane, og isåfall låse opp neste bane, og deretter lagre informasjonen om hvor mange baner og planeter som er låst opp, og høyeste poengsum på hver spilte planet i en ekstern lagringsfil.

For å gjøre menyopplevelsen i spillet enda bedre lagde vi et enkelt animasjonssystem som fungerte med Unitys system for grafiske grensesnitt (canvases). På denne måten kunne vi enklere sørge for at skjermer kom inn og ut av skjermen på en måte som både var

behagelig å se på, samt gav spilleren en bedre mental modell over hvordan spilltilstandene hang sammen, sammenlignet med om skjermer og kameravinkler bare skulle poppet frem.

Vi har gått for et moderne og minimalistisk formspråk på tekst og UI-elementer, rammet inn av en grov, tegneserieaktig industriell stil. Dette skal underbygge ideen om at det faktisk finnes en bedrift der ute som tar det de har av teknologi, ny eller gammel, og terraformer Mars så godt de kan. Tidligere viste vi grafikken på en flyvende skjerm bak spillerne. Hensikten med skjermen var å ta inn enda mer av denne gjenbrukspregede romfartsteknologien, men akkurat denne løsningen var med på å lage støy og ubalanse i bildet. Vi gikk derfor heller for å sette gameplayet i fokus, modernisere grafikken og vise den som et tradisjonelt skjerm-overlay.



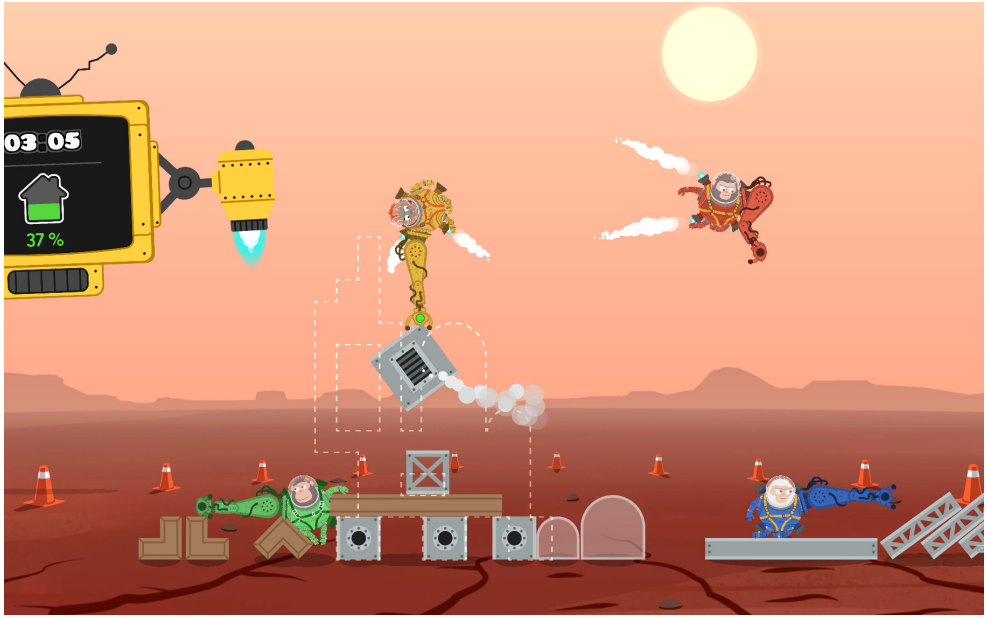
Bestanddelene til en Robotarm i Bonkies

Eksempel på en *sprite-sheet* med grafiske 2D-elementer. *Sprite-sheeten* splittes opp i Unity for å kunne bruke de individuelle elementene for seg selv.

I Bonkies er grafikken som viser hvor mye tid spillerne har igjen formet som et timeglass (se bildet nederst til høyre), for å understreke at det er tid som renner ut. I tillegg er innsnevringen på timeglasset plassert litt lenger ned enn midten. Dette gjør at idet sanden i timeglasset når innsnevringen får spillerne et inntrykk av at de fortsatt har halve tiden igjen, selv om de i realiteten har brukt nærmere to tredjedeler. Denne effekten håper vi er med på å redusere stressnivået hos spillerne.

Menyskjerm fra Bonkies

En industriell, tegneserieaktig look



To iterasjoner av UI i Bonkies

Bildet ble ryddigere, og gameplayet kom mer i fokus da vi gikk fra å vise informasjon på en drone som kjørte frem og tilbake til et tradisjonelt skjerm-overlay

MARKEDSFØRING

Progresjonen på Bonkies ble delt via poster på Instagram og Facebook. I tillegg tok vi opp en GIF av karakteren som flyr litt rundt og flytter på klosser. GIFen ble delt på subredditene /r/Unity2D og /r/Unity3D under tittelen “GIF of our physics based player character, animated using Anima2D”. Tilsammen fikk posten 170 upvotes, og 3.9k visninger. Dette var lavere respons enn vi håpet på. Vi mistenker det er fordi at apen ikke kommer inn i bildet før et sekund har gått. Dermed vises ikke apen på thumbnailen (bildet som vises før animasjonen starter) til GIFen, og kunne nok derfor fremstå som mindre interessant.

NESTE STEG

Neste steg for Bonkies er å få gjort spillet så ferdig at det kan lanseres på Steam Early access. Videre gjenstår det å markedsføre spillet, samt få kontakt med Nintendo, Sony og Microsoft for å kunne gi det ut på deres konsoller etterhvert. I første omgang ønsker vi å gi det ut på Nintendo Switch, ettersom spillet passer veldig godt inn i Nintendos produktkatalog, med sitt familievennlige og fargerike uttrykk, i tillegg til at Switch gjør det enkelt å skaffe to eller flere kontrollere.

Nintendo Switch

Nintendo Switch leveres med en kontroll som kan deles i to.

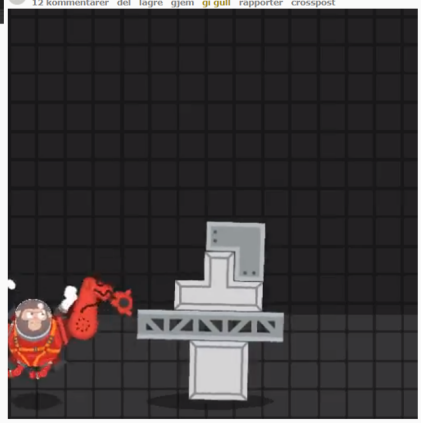
unity2D Kommentarer andre Diskusjoner (1)

135

GIF of our physics based player character, animated using Anima2D Show-off (i.imgur.com)

sendt inn 25 dager siden av Atomic_Command

12 kommentarer del lagre gjem gi gull rapporter crosspost



alle 12 kommentarer • sorter etter: beste

save innholdspolicy formateringshjelp

[–] deanbrown404 10 poeng 25 dager siden
This is awesome. But you should make another gif where he spells out "send nudes"
permalenke embed lagre rapporter gi gull svar

[–] Atomic_Command [S] 5 poeng 25 dager siden
Thanks! And that would require a lot of pieces, might need a team of these guys to do that...
permalenke embed lagre foreldre rapporter gi gull svar

[–] isaelsky21 3 poeng 25 dager siden
A.I. MECHANICS COMING SOON!
Edit: Mechanics.
permalenke embed lagre foreldre rapporter gi gull svar

[–] lorpo1994 5 poeng 25 dager siden
The moving eyes finishes this :D, nice one
permalenke embed lagre rapporter gi gull svar

[–] Atomic_Command [S] 3 poeng 25 dager siden
Thank you! And yup, we learned pretty early on how much life moving eyes give a character (and how unsetteling it can be when it is missing).
permalenke embed lagre foreldre rapporter gi gull svar

[–] Vitasa 2 poeng 25 dager siden
I think it would look better if the legs and arms were a bit heavier thus nit flailing as much. I love the eyes and facial expression.
permalenke embed lagre rapporter gi gull svar

[–] Atomic_Command [S] 1 poeng 25 dager siden
I agree, I'll be tweaking the drag and weight on the arms and legs to make them a bit less floppy. And thank you! :)
permalenke embed lagre foreldre rapporter gi gull svar

[–] _eka_ 2 poeng 25 dager siden
How is this Anima2d compared to Puppet2d?
permalenke embed lagre rapporter gi gull svar

[–] Atomic_Command [S] 1 poeng 25 dager siden
I haven't tried Puppet2D so I can't really say, I chose Anima2D because it is owned by Unity and it had all the tools I needed for this project.
permalenke embed lagre foreldre rapporter gi gull svar

[–] _eka_ 2 poeng 25 dager siden
Yeah. I can see that Anima2D hasn't been updated since Dec 2017, dunno if that's a good sign tho.
permalenke embed lagre foreldre rapporter gi gull svar

[–] crashAlgorithm Beginner 2 poeng 25 dager siden
This is so cool
permalenke embed lagre rapporter gi gull svar

Skjermdump fra Reddit

Responsen på posten med en ape fra Bonkies. Merk hvordan thumbnailen øverst til venstre ikke viser noe av apen.

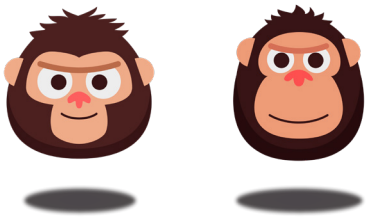
BEDRIFTSUTVIKLING

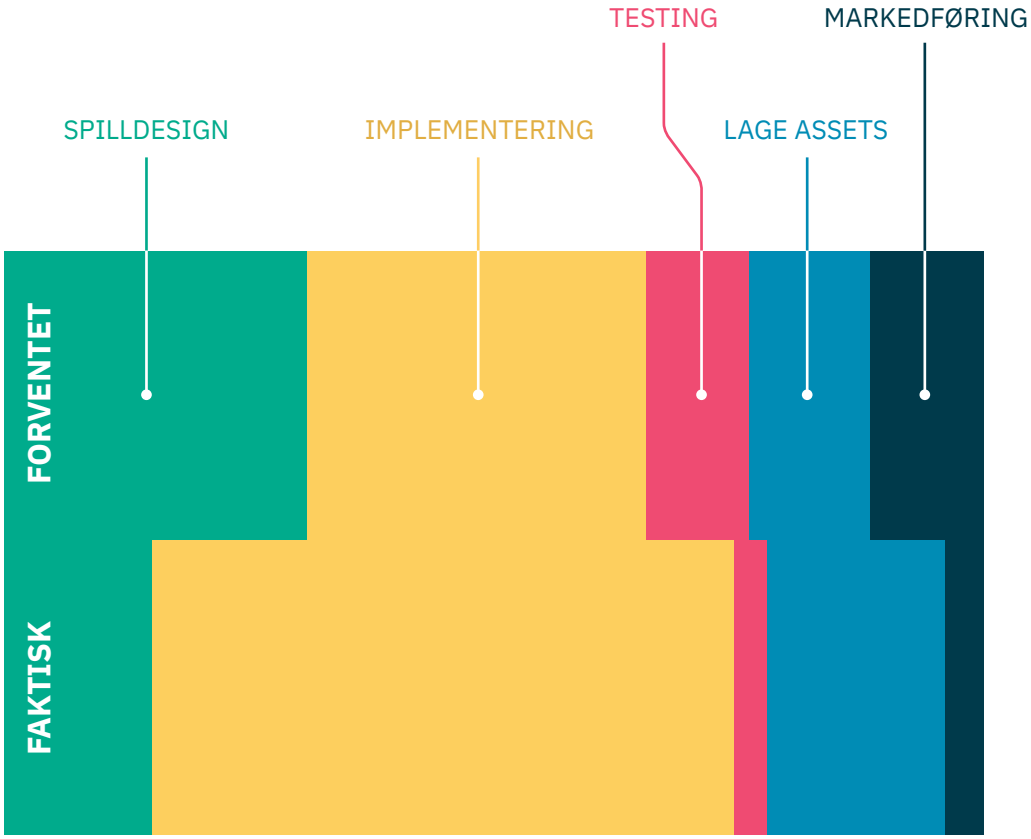
Å FORBLI ET LITE TEAM

Ettersom bedriften oppstod nokså organisk var det aldri et tidspunkt der vi satte oss ned og planla strukturen, men det har gjort seg mer og mer gjeldende, og spesielt i møte med veileder fra Spark* NTNU har vi måttet ta et standpunkt til dette. Vi er et lite team på to personer, og dette har fordeler og ulemper. Ulempene er at det er begrenset hvor mye to personer har tid til å gjøre. Det er også begrenset hvilken kompetanse vi innehar, har mulighet til, og interesse av å lære oss.

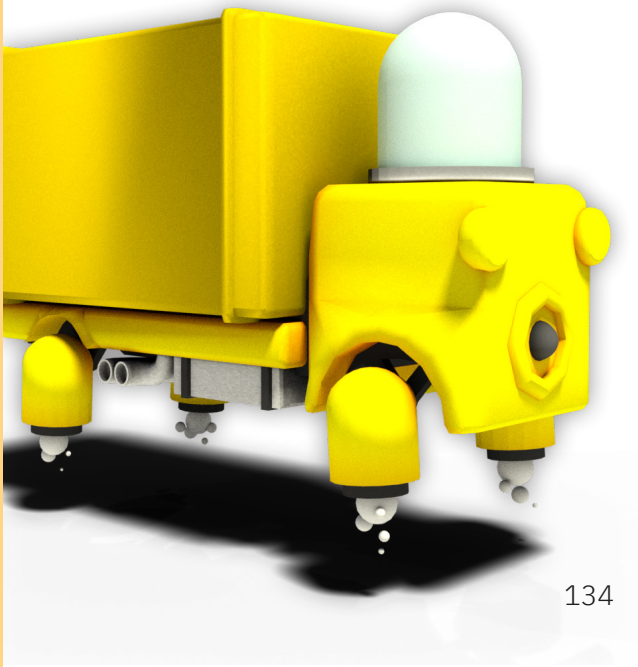
Disse begrensningene har vi vært klar over fra starten av, og vi har derfor valgt å tilpasse forretningsmodellen vår til dette, og satse på små prosjekter som to personer praktisk sett har mulighet for å gjennomføre. Dette bringer oss til fordelene vi har, nemlig at vi som et lite team kan jobbe meget effektivt, og vi har ikke like stort krav til bedriftsorganisering eller inntjening som større bedrifter og prosjekter har.

Hvis vi veier for og i mot ser vi spesielt at forretningssiden av bedriften kunne trengt en dedikert person som kunne sørget for at det hadde gått litt lettere. Når det er sagt så anser vi likevel det å lage et solid produkt som det viktigste kravet for å oppnå suksess, og der føler vi at vi har tilstrekkelig kompetanse.




TESTING
MARKEDFØRING
SPILLDESIGN
IMPLEMENTERING
LAGE ASSETS
FORVENTET
FAKTISK

Tidsbruk



Å FORBLI ET INDIE-STUDIO

I løpet av tiden vi har utviklet spill sammen, også i løpet av masteroppgaven, har vi måttet ta stilling til om vi ønsker å ta på oss oppdrag som spillutviklere for andre, enten i form av direkte tilbud eller oppfordringer til å søke jobber hos selskaper, som PlayPulse, Tiltspot.io, BreachVR og Serious Games A/S, alle Trondheimsbaserte bedrifter. Når vi tenkte igjennom hvorfor vi lager spill ble det tydelig at for oss handler det om mer enn å bare få jobbe med å lage spill. Vi ønsker å uttrykke oss og lage våre egne spill på våre egne premisser

Å FORBLI I TRONDHEIM

Bedriften vår er lokalisert i Trondheim, en by som har et meget aktivt gründermiljø med tilknytning til NTNU, og et spillutviklingsmiljø som er i vekst. Vi er med på å diversifisere et spillutviklingsmiljø preget av en overvekt av programmerere og en mangel på kompetanse innen design. Med tanke på fremtidig ekspansjon kan dette være en fordel for oss, ettersom det gjør oss til en potensielt attraktiv arbeidsplass for programmerere som ønsker å jobbe med spill.

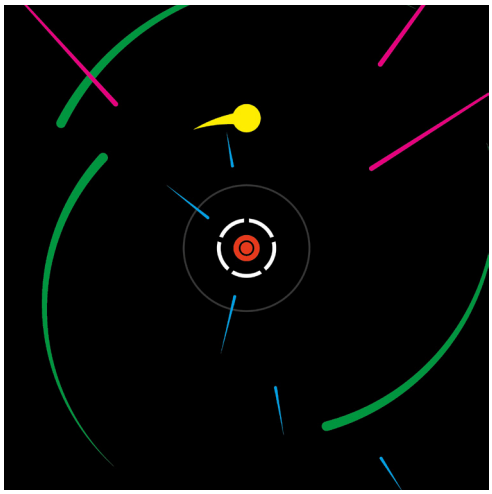
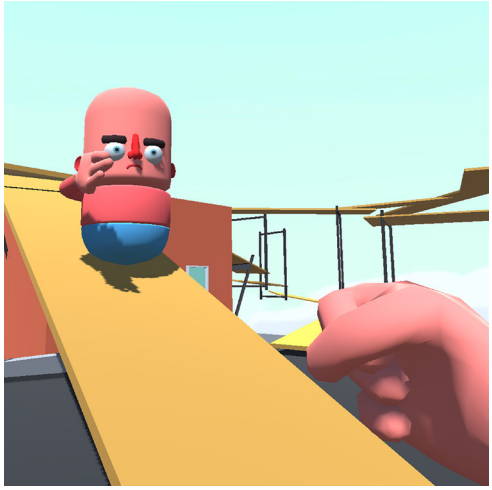
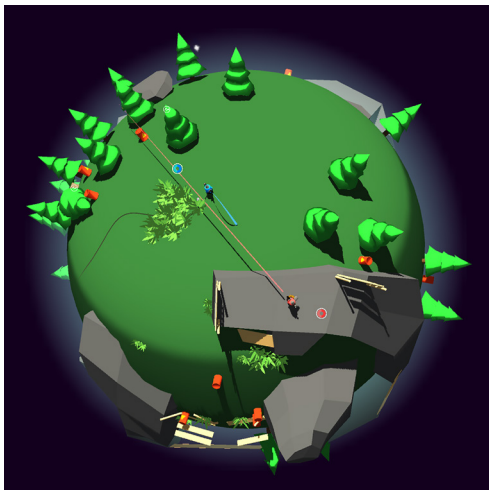
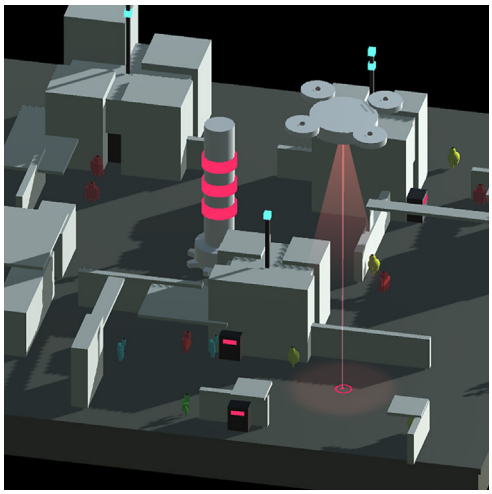
KONKLUSJON

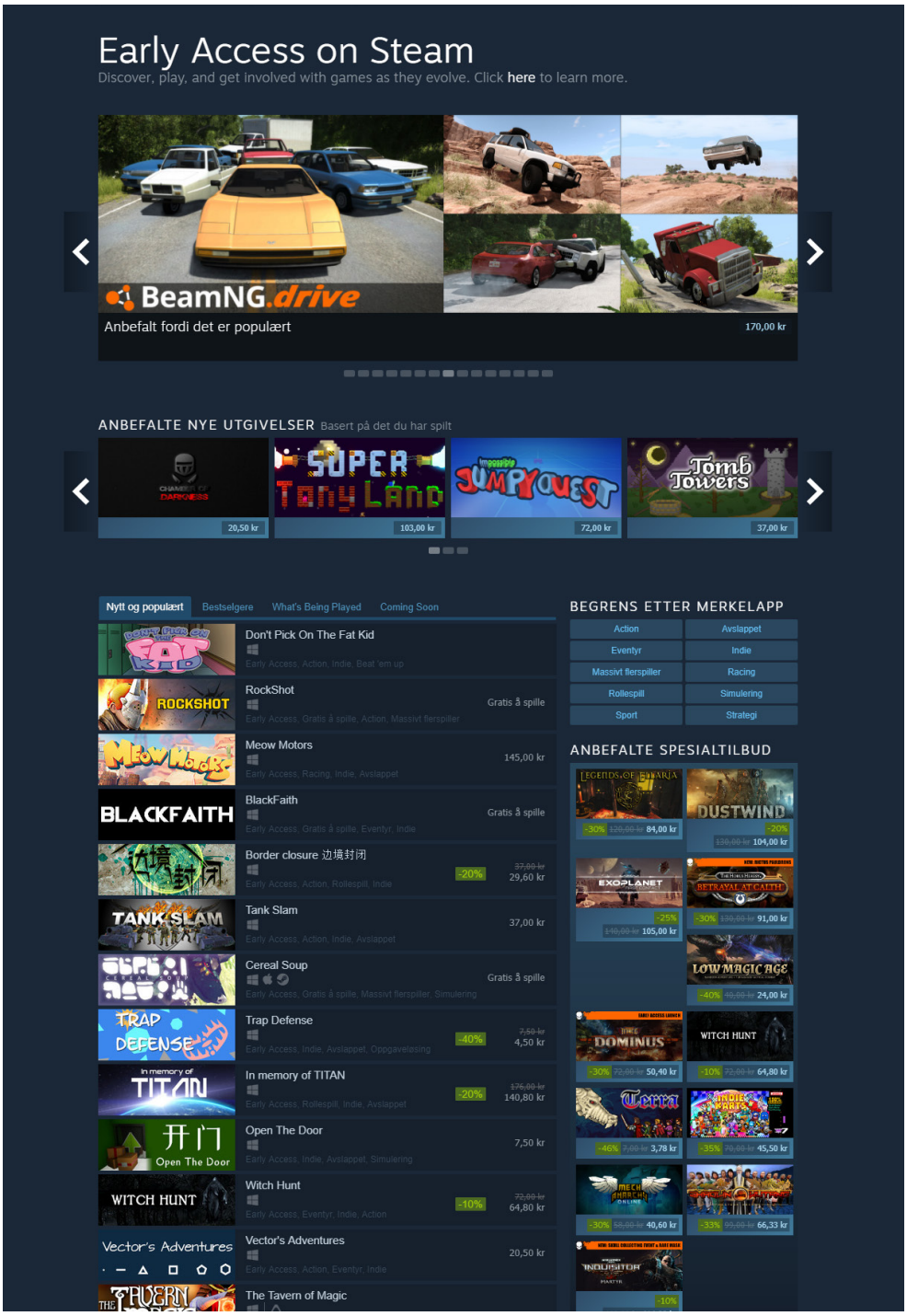
KONKLUSJON

HER ER VI

Etter seks måneders arbeid sitter vi med et godt gjennomarbeidet spillkonsept som har et stort markedspotensiale. Bonkies har vunnet to priser, lokale samarbeidsspill er en trend i markedet og brukertester har gang på gang vist at spillere og publikum har det gøy mens det spilles.

I tillegg har vi fungerende prototyper av Tank Planet og Y.H.T.T.M.I.Y.H. som har vekket interesse i sosiale medier og som vi har lyst å videreutvikle. Bedriften begynner å bli gjenkjent i spillutviklermiljøet i Trondheim, og vi har blitt en node i et nettverk av personer og aktører i det lokale startup-miljøet.

Skjermdump fra STEAM

Steam sin Early Access-tjeneste gjør det mulig å selge spill som fortsatt er under utvikling (skjermdump hentet 05.06.18)

DETTE GJØR VI VIDERE

Snart skal vi annonsere og publisere Bonkies på Steam Early Access. Dette blir en lavterskel prøve på hvordan det er å publisere et større spill til et betalende publikum. I forkant av dette må spillet promoteres, og denne innsatsen må fortsette etter Early Access-utgivelsen.

Den første milepælen for Bonkies er å slippes på Steam, og deretter den neste store milepælen å lanseres på Nintendo Switch. I det vi har promoteringsmateriale klart for Bonkies og en produktside for spillet på Steam kan vi tenke på å pitche det inn til Nintendo for å få tilgang til utviklerverktøyene nødvendige for å få det over på Switch.

På et tidspunkt må vi også formelt opprette et selskap. Når dette bør skje har vi problemer med å avgjøre, men foreløpig avventer vi dette så lenge som mulig for å kunne fokusere mest mulig på utvikling og markedsføring av produktet.

I forbindelse med stiftelse av selskap bør vi se på muligheten for å utvide antall personer i bedriften. Ideelt sett inngår vi samarbeid med noen som i tillegg til å ta seg av de formelle bedriftsgjøremålene, kan bidra til generell utvikling av markedsføringsstrategi og praktisk markedsføring,

Det endelige målet er å drifte en levedyktig spillbedrift, så publisering av Bonkies blir første forsøk på å generere inntekter nok til å kunne lønne oss selv og fortsette på nye prosjekter.

NI NYTTIGE LÆRDOMMER

1. Å utvikle prototyper under strengt tidspress er effektivt, lærerikt og gøy. En målsetningen om at to uker skal holde til ferdigstilling, markedsføring og publisering krever god avgrensing og tidsprioritering. Vi opplevde premisset om å hoppe rett på en ny utviklingssyklus som ødeleggende for helhjertet markedsføringsinnsats på spillene.

2. Markedsføring er tidkrevende, og en kunst man må lære seg. Dersom det ikke blir satt av dedikert tid til denne jobben kan den fort bli nedprioritert. Små, spissede drypp av innhold så ut til å fenge vel så godt som å presentere et helt konsept på en gang.

3. MVPer lages fort, men har en tendens til å preges av dårlig koding og prosjektorganisering. Når MVPen har gjort sin nytte vil man kunne være tjent med å rydde opp i koden og organisere prosjektet bedre, eventuelt bare starte på nytt.

4. Det er alltid lett å bli opphengt i detaljer, både under koding, modellering, tegning og design av effekter. Dette er risikabel tidsbruk så lenge grunnmekanikkene ikke er avklart. Ikke bruk mer tid enn nødvendig på assets man risikerer å måtte kaste.

5. Brukertesting gir uvurderlig feedback, både i form av verbale tilbakemeldinger og observerbare handlinger blant brukerne. Brukere kommer gjerne med velmenende forslag til hva som kan gjøres bedre. Ta et steg tilbake og analyser hva som egentlig ikke fungerte. Dette for å unngå en cocktail av endringer som gjør det vanskelig å holde styr på hva det egentlig er som fungerer og ikke fungerer.

6. For å sikre en god spillopplevelse er riktig innpakning bortimot like viktig som selve gameplayet. Spilleren bør ledes inn og ut av gameplayet, og dette krever logiske overganger mellom før-, under- og etterspilltilstander - en god game loop.

7. For å lette markedsføringsjobben er det ideelt om spillet kan "markedsføre seg selv". Design spillet med en tanke om at det bør inneholde hooks, elementer som mer eller mindre alene kan interessere eller fenge en bruker som hører om spillet for første gang.

8. Selv om det er fristende å stadig lære flere ting som kan benyttes i senere prosjekter, som å utvikle for virtual reality eller online-spill, så er det en krevende jobb å mestre og vedlikeholde en så bred kompetanse for et lite team. Identifiser heller teamets styrker, dyrk disse, og lær nytt når det trengs.

9. Businessperspektivet av å starte en spillbedrift er ganske annerledes enn gamedev-perspektivet. Som utviklerteam er det lett å fokusere på å utvikle produktet til å bli som bra som mulig for å kunne hevde seg i den harde konkurransen. Som bedrift bør man tenke på strategier som involverer å få en versjon av produktet ut på markedet så fort som mulig. Et team motivert mest av design og utvikling av produktet har godt av å skaffe krefter som drar i retning av markedsføring og utgivelse.

THE GAME IS ON!

REFERANSER

Clark, R., 2018. How to Consistently Make Profitable Indie Games. Brace Yourself Games.

Gray, K., Gabler, K., Shodhan, S., Kucic, M., 2005. How to Prototype a Game in Under 7 Days. Gamasutra.

Kidwell, E., 2018. How Sunless Sea devs found success through open production. Gamasutra.

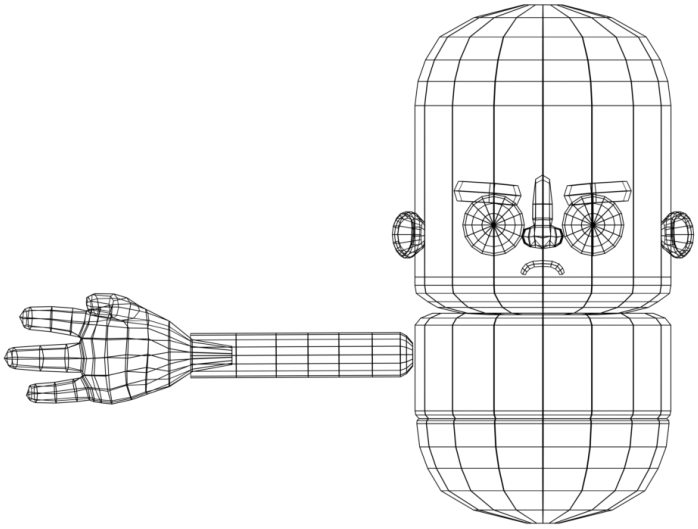
Portnow, J., 2014. Fail Faster - A Mantra for Creative Thinkers. Extra Credits.

Ries, E., 2011. The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses, 1 edition. ed. Currency, New York.

Schell, J., 2014. The Art of Game Design: A Deck of Lenses, Second Edition, 2nd edition. ed. Schell Games, Pittsburgh, PA.

Tufte, E.R., 2001. The Visual Display of Quantitative Information, 2nd edition edition. ed. Graphics Press, Cheshire, Conn.

Ware, C., 2012. Information Visualization, Third Edition: Perception for Design, 3 edition. ed. Morgan Kaufmann, Waltham, MA.



TO BE CONTINUED @STUDIOGAUNTLET...

APPENDIX

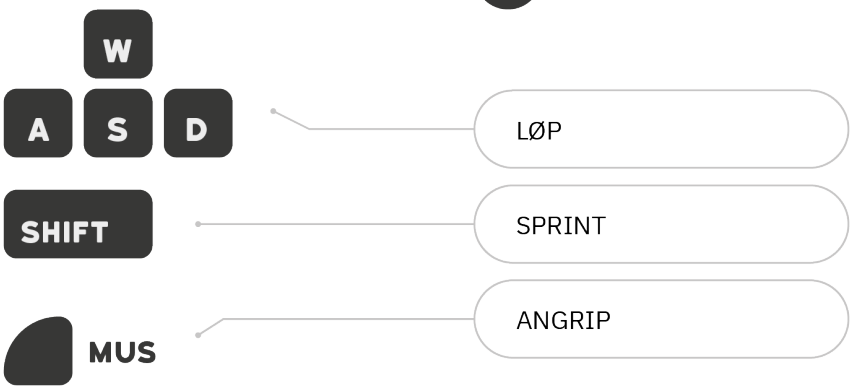
HVORDAN SPILLE

SUPER R.P.S.

1 MÅLET MED SPILLET

Bekjemp motspillerne dine ved å angripe dem med stein, saks eller papir. Alle må velge stein, saks eller papir på nytt hvert halve minutt.

2 KONTROLLER



AVSLUTT SPILL: ALT + F4

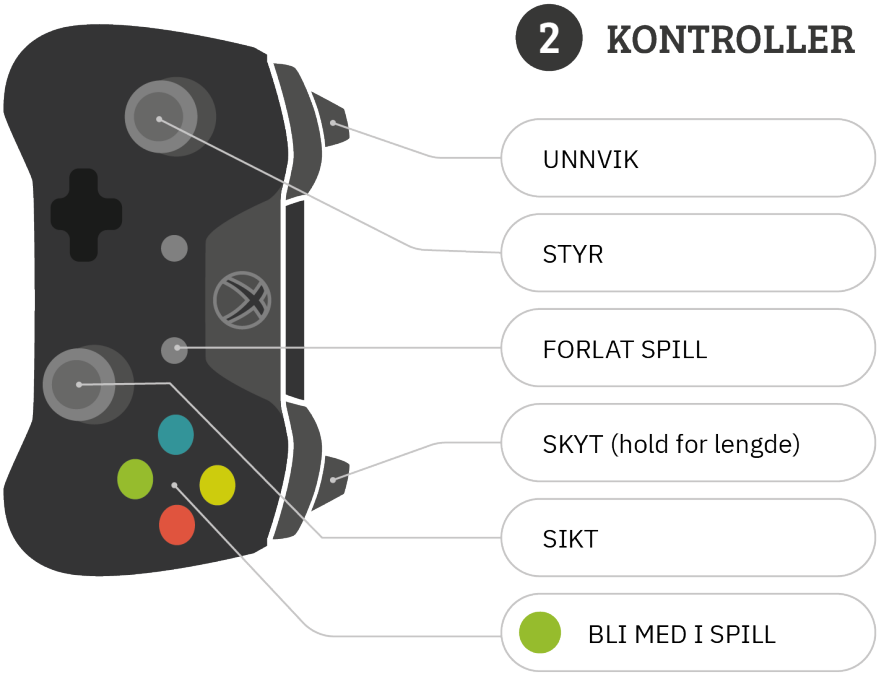
HVORDAN SPILLE

TANK PLANET

1 MÅLET MED SPILLET

Bekjemp motspillerne dine.
Plukk opp kasser for å oppgradere våpen eller få ekstra liv.

2 KONTROLLER



AVSLUTT SPILL: ALT + F4

HVORDAN SPILLE

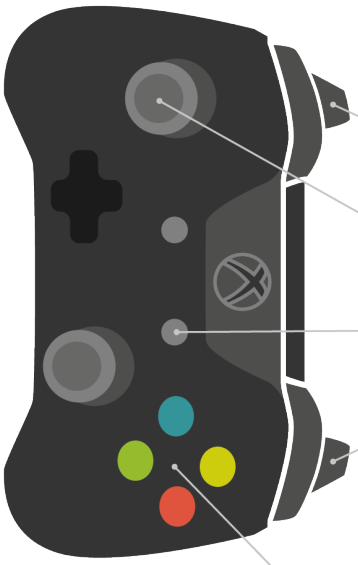
Y.H.T.T.M.I.Y.H.

1 MÅLET MED SPILLET

Dronen skal skyte tre menneskelige spillere eller overleve i 5 minutter.

Menneskene må samle 3 minnepinner og levere dem til en åpen terminal før tiden er ute.

2 KONTROLLER



DRONE	MENNESKE
ROTER BRETT	TENN LYS
FLY	GÅ
NULLSTILL SPILL	
ROTER BRETT	TENN LYS
MARKÉR	BLI MED / FORKLED
	FORKLED
SKYT	FORKLED
	FORKLED

AVSLUTT SPILL: ALT + F4

HVORDAN SPILLE

BONKIES

1 MÅLET MED SPILLET

Samarbeid om å bygge figurene før tiden renner ut.
Vær forsiktig så hjelmene ikke knuses.

2 KONTROLLER