

TITTEL:

SpaceChase

KANDIDATNUMMER(E):

Asbjørn Frostad

DATO:	EMNEKODE:	EMNE:	DOKUMENT TILGANG:
20.05.2020	IE303612	Bacheloroppgave	
STUDIUM:		ANT SIDER/VEDLEGG:	BIBL. NR:
Bacheloroppgave		21/2	

VEILEDER(E) :

Arne Styve

SAMMENDRAG:

Denne rapporten beskriver grunnlag for, gjennomføring og resultatet av et bachelorprosjekt gjennomført ved NTNU Ålesund. Oppgaven ble gitt og gjennomført av meg selv. Oppgaven var å utvikle et spill for Android ved hjelp av spillmotoren Unity. Spillet skulle handle om en astronaut som faller ut av et romskip i fart mens den reiser i verdensrommet. Astronauten skulle da prøve å ta igjen romskipet med en jetpack mens den unngår asteroider og andre hinder i verdensrommet.

Hovedmål var at spillet skulle være funksjonelt på android, og kunne spilles på de fleste oppløsninger tilgjengelig på android. I tillegg til at spillet skulle være enkelt å sette seg inn i, og komfortabelt å spille. Jeg føler at spillet er både kompatibelt med de fleste android-mobiltelefoner og spillet er veldig enkelt å bruke.

Ved slutten av prosjektet har jeg et fungerende spill som fungerer stort sett som planlagt.

Denne oppgaven er en eksamensbesvarelse utført av studenter ved NTNU i Ålesund.

Obligatorisk egenerklæring/gruppeerklæring

Du/dere fyller ut erklæringen ved å klikke i ruten til høyre for den enkelte del 1-6:		
1.	Jeg/vi erklærer herved at min/vår besvarelse er mitt/vårt eget arbeid, og at jeg/vi ikke har brukt andre kilder eller har mottatt annen hjelp enn det som er nevnt i besvarelsen.	<input checked="" type="checkbox"/>
2.	Jeg/vi erklærer videre at denne besvarelsen: <ul style="list-style-type: none">• ikke har vært brukt til annen eksamen ved annen avdeling/universitet/høgskole innenlands eller utenlands.• ikke refererer til andres arbeid uten at det er oppgitt.• ikke refererer til eget tidligere arbeid uten at det er oppgitt.• har alle referansene oppgitt i litteraturlisten.• ikke er en kopi, duplikat eller avskrift av andres arbeid eller besvarelse.	<input checked="" type="checkbox"/>
3.	Jeg/vi er kjent med at brudd på ovennevnte er å <u>betrakte som fusk</u> og kan medføre annullering av eksamen og utestengelse fra universiteter og høgskoler i Norge, jf. Universitets- og høgskoleloven §§4-7 og 4-8 og Forskrift om eksamen §§14 og 15.	<input checked="" type="checkbox"/>
4.	Jeg/vi er kjent med at alle innleverte oppgaver kan bli plagiatkontrollert i Ephorus, se Retningslinjer for elektronisk innlevering og publisering av studiepoenggivende studentoppgaver	<input checked="" type="checkbox"/>
5.	Jeg/vi er kjent med at høgskolen vil behandle alle saker hvor det forligger mistanke om fusk etter høgskolens studieforskrift §31	<input checked="" type="checkbox"/>
6.	Jeg/vi har satt oss inn i regler og retningslinjer i bruk av kilder og referanser på biblioteket sine nettsider	<input checked="" type="checkbox"/>

Publiseringsavtale

Studiepoeng:

Veileder:

Fullmakt til elektronisk publisering av oppgaven

Forfatter(ne) har opphavsrett til oppgaven. Det betyr blant annet enerett til å gjøre verket tilgjengelig for allmennheten ([Åndsverkloven §2](#)). Alle oppgaver som fyller kriteriene vil bli registrert og publisert i Brage HiM med forfatter(ne)s godkjenning. Oppgaver som er unntatt offentlighet eller båndlagt vil ikke bli publisert.

Jeg/vi gir herved NTNU i Ålesund en vederlagsfri rett til å gjøre oppgaven tilgjengelig for elektronisk publisering:

ja nei

Er oppgaven båndlagt (konfidensiell)?

ja nei

(Båndleggingsavtale må fylles ut)

- Hvis ja:

Kan oppgaven publiseres når båndleggingsperioden er over?

ja nei

Er oppgaven unntatt offentlighet?

ja nei

(inneholder taushetsbelagt informasjon. [Jfr. Offl. §13/Fvl. §13](#))

Dato:

INNHold

SAMMENDRAG	6
1 INNLEDNING	7
2 TEORETISK GRUNNLAG	8
2.1 LIGNENDE SPILL	8
2.2 MENNESKE-MASKIN INTERAKSJON	8
2.2.1 <i>Don Norman's usability principles [2]</i>	8
2.2.2 <i>Gestalt-prinsippene [3]</i>	9
2.3 ARBEIDSPROCESS	10
2.3.1 <i>Agil utvikling [4]</i>	10
2.3.2 <i>Scrum [5]</i>	10
2.4 GOOGLE PLAY [18]	11
3 MATERIALER OG METODE	12
3.1 PROSJEKTSTYRING	12
3.2 TEKNISK	12
3.2.1 <i>Spillmotor</i>	12
3.2.2 <i>Grafiske elementer</i>	12
3.2.3 <i>Lyd</i>	12
3.2.4 <i>Prosjektstyringsverktøy</i>	13
3.2.5 <i>Google Play</i>	13
4 RESULTATER	13
4.1 SPILL	13
4.1.1 <i>Hinder</i>	13
4.1.2 <i>Power-up</i>	14
4.1.3 <i>Coin</i>	14
4.1.4 <i>Våpen</i>	14
4.2 SCENE HIERARKI	15
4.3 GUI	15
4.3.1 <i>Main Menu</i>	16
4.3.2 <i>Upgrades</i>	16
4.3.3 <i>Player selection</i>	17
4.3.4 <i>Controls</i>	17
4.3.5 <i>Reset</i>	17
4.3.6 <i>DeathScreen</i>	18
4.3.7 <i>Pause</i>	19
4.3.8 <i>Loading screen</i>	19
4.4 SCORE SYSTEM	19
4.5 LAGRINGSSYSTEM	20
4.6 OPPGRADERINGER	20
4.7 KONTROLLER	20
4.8 TESTING	21
4.9 PUBLISERING	21
4.9.1 <i>Reklame</i>	21
5 DRØFTING	21
5.1 INSPIRASJON	21
5.2 GUI	22
5.3 «SENSE OF PROGRESS»	22
5.4 «CUSTOMIZATION»	23
5.5 KONSEPT	23
5.6 KONTROLLER	24
5.7 PROBLEMSTILLINGENE	24
5.7.1 <i>Hvordan lage et spill som folk vil spille framfor alle de andre tusenvis av android spill?</i>	24

5.7.2	<i>Hvordan lage et spill så intuitivt og enkelt samtidig som det skal være underholdende?</i>	25
5.7.3	<i>Hvordan gjøre spillet kompatibelt med de mange forskjellige oppløsningene som er tilgjengelig for android? 25</i>	
6	KONKLUSJON	26
7	REFERANSER	26
	VEDLEGG	27

SAMMENDRAG

Denne rapporten beskriver grunnlag for, gjennomføring og resultatet av et bachelorprosjekt gjennomført ved NTNU Ålesund. Oppgaven ble gitt og gjennomført av meg selv. Oppgaven var å utvikle et spill for Android ved hjelp av spillmotoren Unity. Spillet skulle handle om en astronaut som faller ut av et romskip i fart mens den reiser i verdensrommet. Astronauten skulle da prøve å ta igjen romskipet med en jetpack mens den unngår asteroider og andre hinder i verdensrommet.

Hovedmål var at spillet skulle være funksjonelt på android, og kunne spilles på de fleste oppløsninger tilgjengelig på android. I tillegg til at spillet skulle være enkelt å sette seg inn i, og komfortabelt å spille. Jeg føler at spillet er både kompatibelt med de fleste android-mobiltelefoner og spillet er veldig enkelt å bruke.

Ved slutten av prosjektet har jeg et fungerende spill som fungerer stort sett som planlagt.

1 INNLEDNING

Dette prosjektet er gitt og utført av meg selv og handler om å lage et spill i spillmotoren Unity som skal være tilgjengelig for Android.

Spillet skal handle om en astronaut som faller ut av et romskip i fart mens den reiser i verdensrommet. Astronauten skal da prøve å ta igjen romskipet med en jetpack mens den unngår asteroider og andre hinder i verdensrommet.

Dette spillet skal inneholde følgende:

- Funksjonell player-controller
- Interaktive hinder etc i spill.
- Funksjonell main- og pause-menu.
- Funksjonell score system.
- Lyd effekter.
- Animasjoner.
- Spesialeffekter som eksplosjoner etc.
- Character modeller og etc grafikk.
- At man kan lagre informasjon som innstillinger og score etc lokalt på telefonen.
- Minst et funksjonelt nivå.

Målet til prosjektet er å utvikle et underholdende og enkelt å forstå spill, som hvem som helst, når som helst kan spille. At spillet skal være så intuitivt og enkelt å spille at de fleste kan begynne å spille med en gang uten å tenke seg om. Målet er at det skal være behagelig nok å spille at du ikke trenger å tenke deg om noe særlig hver gang du rører skjermen.

Problemstillinger:

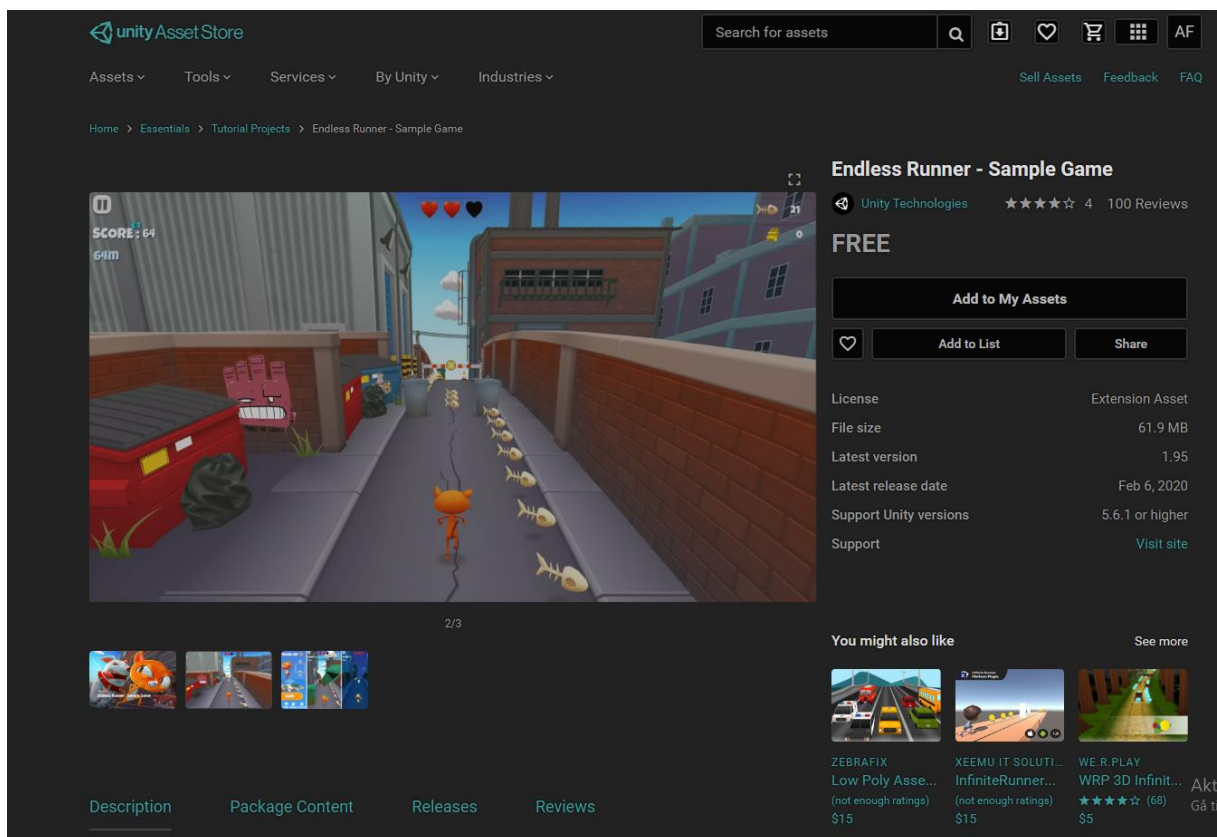
- Hvordan lage et spill som folk vil spille framfor alle de andre tusenvis av android spill?
- Hvordan lage et spill så intuitivt og enkelt samtidig som det skal være underholdende?
- Hvordan gjøre spillet kompatibelt med de mange forskjellige oppløsningene som er tilgjengelig for android?

2 TEORETISK GRUNNLAG

2.1 Lignende spill

Det finnes en god del lignende spill lagd i Unity [6] fra før. Det finnes i tillegg mange tutorials og assets laget av Unity selv. Et relevant eksempel er «Endless Runner – Sample game» [1] i Unity asset store.

Kjennetegn på slike spill er at du skal kunne spille så lenge du klarer uten å komme til et mål. I disse spillene blir det mer fokus på å slå highscoren din ved å tjene penger å oppgradere ting.



2.2 Menneske-maskin interaksjon

2.2.1 Don Norman's usability principles [2]

- **Visibility** er prinsippet som sier at hvor mer synlig et element er, jo mer sannsynlig er det at brukere vil vite om det og hvordan å bruke det. Like viktig er det omvendte; når noe er ute av syne, er det vanskelig å vite om det og bruke.
- **Feedback** er prinsippet om å gjøre det klar til bruker hvilken handling som er blitt tatt og hva det har resultert i. Det som er viktig her er å designe på en måte som gjør at bruker aldri trenger å gjette hvilken handling som er blitt tatt og konsekvensene for handlingen.

- **Constraints** handler om å limitere mengden av handlingsmuligheter for bruker, med mål å simplifisere brukergrensesnittet og lede brukeren til den korrekte neste handling. Constraints er brukt for å klargjøre hva brukeren burde gjøre. Endeløse muligheter gjør ofte at bruker blir forvirret.
- **Mapping** handler om å ha et klart forhold mellom kontroller og hvilken effekt det har. Du vil at mappingen skal føles så naturlig som mulig.
- **Consistency** refererer til å ha lignende operasjoner og lignende element for å oppnå lignende oppgaver. Ved å gi konsistente elementer gjennom hele opplevelsen, vil dette gjøre det mye lettere å forstå.
- **Affordance** refererer til egenskapen til et objekt å tillate folk å vite hvordan å bruke det. «Essentially to afford means to give a clue». Den fysiske knappen på en mus gir et hint om at den kan bli klikket for å gjøre en handling. Når et objekt har sterk affordance, er det veldig klart hvordan det kan bli brukt.

2.2.2 Gestalt-prinsippene [3]

Gestalt-prinsippene er reglene som forklarer hvordan hjernen vår organiserer stimuli og det vi ser. Disse prinsippene hjelper oss å forstå hvorfor vi ser ting på den måten vi gjør.

- **Similarity** – Det er helt naturlig å gruppere ting i lag. I gestalt, blir like elementer visuelt gruppert uavhengig av hvor nært de er hverandre. De kan bli gruppert basert på farge, form eller størrelse. Likhet kan bli brukt til å binde sammen elementer som kanskje ikke er rett ved siden av hverandre i et design.
- **Continuation** – Objekter blir oppfattet som hele gjenstander i stedet for løse gjenstander som ikke hører sammen.
- **Closure** – Closure tillater øynene dine til å følge noe som en prikket linje til enden. Enkelt sagt vil hjernen vår automatisk lukke åpne former.
- **Proximity** – Proximity refererer til hvor nærmere elementer er til hverandre. De sterkeste proximity forhold er de med overlappende subjekter, men bare å gruppere objekter inn i et enkelt område kan også ha en sterk proximity effekt.
- **Figure/Ground** – Dette prinsippet er likt closure prinsippet i det at det utnytter måten hjernen prosesserer negative space. Hjernen din vil skille mellom objekter den anser som i foreground av et bilde og bakgrunnen.
- **Symmetry and order** – Dette prinsippet sier at hjernen din vil oppfatte tvetydige former i den enkleste måten som mulig.
- **Common fate** – Dette prinsippet sier at folk vil gruppere sammen ting som peker til eller bevege seg i samme retning.

2.3 Arbeidsprosess

2.3.1 Agil utvikling [4]

Agile metoder handler om tilpasningsdyktig planlegging, selvorganisering og korte leveringstider. Det er fleksibelt, raskt, og har som mål kontinuerlige forbedringer i kvalitet.

- **Rasker, mindre** – I stedet for den tradisjonelle software utviklingsmetoden, så fokuserer agile metoder å utgi det første inkrementet i de første få ukene.
- **Kommunikasjon** – Agile team jobber daglig i lag på hvert steg av prosjektet ved hjelp av face-to-face møter. Dette samarbeidet sørger for at prosessen går smurt selv om tilstander endres.
- **Feedback** – agile team sporer suksessen og hastigheten til utviklingsprosessen regelmessig. Hastigheten måles etter levering av hvert inkrement.
- **Tillit** – agile team er selv-organiserende. I stedet for å følge et manifest av regler fra ledelsen, forstår teamet målene og lager sin egen vei til å nå dem.
- **justering** – deltagere justerer prosessen kontinuerlig, som gjør det mulig være veldig tilpasningsdyktig.

2.3.2 Scrum [5]

Scrum rammeverket er basert på kontinuerlig læring og justeringer til fluktuerende faktorer. Scrum erkjenner at teamet ikke vet alt i begynnelsen av et prosjekt, og vil utvikle seg gjennom erfaringer. Scrum er strukturert til å hjelpe å naturlig tilpasse seg til endrende tilstander og bruker-krav, med re-prioritering bygd inn i prosessen og korte release cycles, så teamet ditt kan kontinuerlig lære å forbedre seg.

- **Sprints** – en sprint er en kort, tids-rammet periode der et scrum team jobber for å bli ferdig med en bestemt mengde arbeid.
- **Sprint-review** – et sprint-review er holdt i slutten av en sprint for å se gjennom det som har blitt gjort og se om det er nødvendig å justere noe eller tilpasse backlog.
- **Sprint retrospective** – Sprint retrospective er et møte som skjer etter sprint review. I dette møtet blir det diskutert hva som gikk bra i siste sprint, hva som kan bli bedre og hva som må forbedres i neste sprint.

2.4 Google play [18]

Under skal jeg gå gjennom et par punkt for «best practice» når skal laste opp en app til google play.

- **Forstå utvikler-retningslinjene** – Utvikler-retningslinjene er designet for å sørge for at Play butikken forblir en pålitelig ressurs for Android brukere. Det er viktig å løse retningslinjene på grunn av at det kan bli konsekvenser for brudd.
- **Lag en utvikler konto** – Registrer en utvikler konto og sjekk at konto detaljene er riktig. Hvis du skal selge produkter, må du også lage en selger konto.
- **Utvikle for et nylig API nivå** – Google Play krever at nye apper skal bli utviklet for minst Android 9.0 (API nivå 28).
- **Bygg Android App bundle 'en din** – Når du er klar til å gjøre appen din tilgjengelig for brukere, enten for testing eller som et ferdig produkt, bygg din Android App Bundle. Google Play sin «Dynamic Delivery» vil bruke din App Bundle til å generere, signere og gi optimerte APK'er til hver bruker sin enhet, som vil resultere i en mindre app å laste ned og installere.
- **Planlegg appen din sin Play butikk-side** – Forbered forklaringer, reklameringsgrafikk, skjermbilder, og videoer du vill legge til appen din sin Play-butikk-side.
- **Sett opp appen din sin pris og hvor den skal distribueres** – Når du har bestemt din «monetization model», setter du appen din som gratis eller betalt og velger hvilke land den skal bli distribuert til.
- **Bestem appen din sin aldersgrense** – å gi appen en passende aldersgrense er et krav av utvikler-retningslinjene, men det sørger også for at appen din blir sett av den riktige aldersgruppen.
- **Siste sjekker og publisering** – Først gå tilbake og dobbeltsjekk at du har gjort alt på lista. Nå er du klar til å publisere appen. Hvis du skal gi en oppdater, bruk «Staged rollouts» for å gi ut oppdateringene mer progressivt til brukere. Dette gir deg mulighet til å stoppe oppdateringen hvis du finner et problem, så du kan limitere mengden brukere det påvirker.

3 MATERIALER OG METODE

3.1 Prosjektstyring

I dette prosjektet har jeg tatt i bruk rammeverket Scrum, som ligger innenfor agile metoder. Prosjektet ble delt inn i sprinter på 14 dager hver. I slutten av hver sprint er det et møte med veileder der det blir Sprint review og Sprint retrospective i tillegg til noen eventuelle spørsmål.

Første oppgave som måtte bli gjort var å lage en kravspesifikasjon der det står alle elementene som spillet skulle inneholde. Basert på dette, begynte jeg å lage første utkast av eventuelt design.

Før hver iterasjon la jeg til veldig spesifikke oppgaver i backloggen, og fant ut hva jeg ville fått gjort i neste sprint. Oppgaver ble også ofte lagt til eller justert i løpet av en sprint etter på behov.

3.2 Teknisk

3.2.1 Spillmotor

I dette prosjektet har jeg tatt i bruk spillmotoren Unity [6]. Unity er en cross-platform spillmotor som gir brukere mulighet til å lage spill og annet i både 2D og 3D.

Spillmotoren gir et API i C# og «drag and drop» funksjonalitet i tillegg til mange komponenter og plug-ins som gjør det mye lettere og mer oversiktlig å lage spill.

Ettersom API'en og Unity editoren generelt bruker C# har jeg også brukt C# for å lage dette spillet. Måten C# script blir brukt i Unity er at hvert script må være bundet til et spillobjekt, og så scripte ut ifra objektets ståsted. Det er mange innebygde komponenter i Unity som du kan legge til et spillobjekt som for eksempel «collidere», «rigidbody», «rendere» etc.

3.2.2 Grafiske elementer

For alt visuelt og bilder lagd i dette prosjektet har jeg brukt Gimp [16] og i et tilfelle Blender [17]. Framgangsmåten min når jeg lagde en modell eller bilde, var å søke etter et referansebilde, etterfulgt av å tegne liknende eller hele etter referansebildet.

Animering av alle modeller ble lagd med innebygd verktøy i Unity.

3.2.3 Lyd

For bakgrunnsmusikken brukt i spillet ble det brukt musikk [7] under «creative commons attribution license» funnet på en webside linket i referanser.

For andre lydeffekter som skyting og eksplosjoner, ble det lagd selv ved hjelp av Audacity [8] og et verktøy på webside JFXR [9].

3.2.4 Prosjektstyringsverktøy

Alt prosjektstyringsverktøy brukt i løpet av prosjektet var Jira [10] og Confluence [11]. Jira ble brukt for alle arbeidsprosesser gjort i løpet av prosjektet, sånt som oppgave-definering og -kategorisering etc i backlog og sprint. Confluence ble brukt for å dokumentere alt fra notater til sprint-reviews og -retrospective etc. Annet enn det ble forprosjektrapport og bachelor-rapport lagd i Microsoft Word.

3.2.5 Google Play

For å publisere spillet brukte jeg Google Play først og fremst fordi mobilen min kjører android. Bortsett fra det er det også billigere å registrere seg som utvikler på Google Play.

4 RESULTATER

4.1 Spill

Nedenfor er et skjermbilde av spillet. Illustreringen viser at etter spiller faller ut av romskipet, holder spiller fortsatt farten med hjelp av jetpack'en og kan dukke unna hindre. Spiller har også en pistol som kan brukes til å skyte hindre istedenfor.



4.1.1 Hinder

Det er 3 forskjellige typer hinder; søppel, asteroide og fiende romskip. Søppel har 3 forskjellige modeller, men fungerer på samme måte. Hvis spiller kommer i kontakt med søppel gjør det ikke at du dør, men at du blir dyttet et stykke nedover. Hvis du blir dyttet for langt nedover og ut av synet er det game over. Hvis spiller kommer i kontakt med asteroide er det game over med en gang. Fiende romskip fungerer slik at etter de blir

viset holder de seg i toppen av skjermen til de er skutt ned. Fiende romskip kan skyte rett ned, og hvis spiller kommer bort i skuddet dør spiller.

Når du går inn i spill scenen, er det først bare søppel som viser. Etter at scoren din blir 100 kan asteroider også vise, og etter 200 kan fiende romskip vise i tillegg.

Måten det blir bestemt hvilke hinder som skal bli viset når og hvor, er ved å bruke 3 if statements; hvis timer er større en «cooldown» er det for hver mulig vis-posisjon (3), er det random [14] om det skal bli viset noe, og random hvilke hinder som skal bli viset. «Cooldown» blir mindre og mindre etter tid til den kommer til en minimumsverdi. Farten til hinder blir samtidig større og større etter tid til den kommer til en maksimumsverdi.

4.1.2 Power-up

Det er 4 forskjellige typer power-ups; drivstoff, ammunisjon, skjold og magnet. Hvis spiller kommer i kontakt med drivstoff objektet, blir jetpack'en din fylt på så du ikke går tom for drivstoff. Hvis spiller kommer i kontakt med ammunisjons objektet, får du 5 mer ammunisjon. Hvis spiller kommer i kontakt med skjold, får du en barrier rundt deg som gjør at alle hinder barrier kommer i kontakt med blir ødelagt. Hvis spiller komme i kontakt med magnet, tiltrekker du deg alle power-ups og coins som kommer nært spiller.

Måten det blir bestemt hvilke power-ups skal vise er den samme måten som det blir bestemt med hinder, bortsett fra at farten til power-ups ikke endrer seg over tid, og at skjold og magnet kan vise separat fra de andre power-ups og hvor som helst.

4.1.3 Coin

Hvis spiller kommer i kontakt med en coin, får du 5 mer coins. Coin blir har en sjanse til å vise på en random posisjon, og «cooldown» blir raskere og raskere over tid.

4.1.4 Våpen

Det er 4 forskjellige våpen i spillet; pistol, maskingevær, bazooka og laserpistol. Skyter en normal kule med konstant fart, maskingevær oppfører seg samme som pistol, men kulen går raskere og maskingevær kan skyte raskere. Bazooka skyter et prosjektil som på kontakt med hinder, blir en eksplosjon som kan ødelegge ting innen radius.

Laserpistol fungerer som en linje som bare manifesteres noen millisekund og ødelegger hinder den rører.

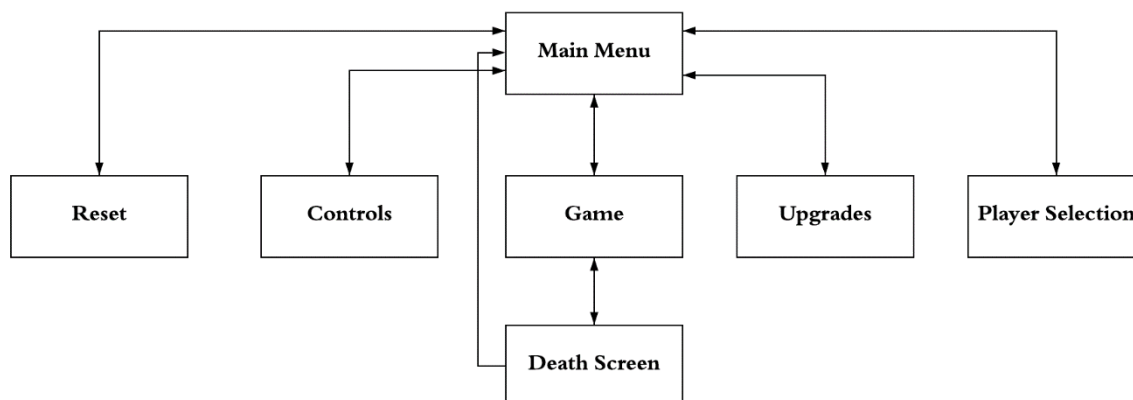
Det er hovedsakelig to måter å skyte i Unity. En måte er ved å bruke raycasting [15], og den andre måten er ved å instansiere et objekt med egen collider fra våpenet. For pistol,

maskingevær og bazooka har jeg brukt den andre metoden, på grunn av at det er mer praktisk for å kunne simulere farten og de andre variablene til prosjektilet. Laserpistol bruker raycast, fordi det har ingen fart eller noe prosjektil i det hele tatt.

4.2 Scene hierarki

Nedenfor er en illustrering av hierarkiet mellom Scenene i Unity. Denne modellen viser med pilene hvilken scene du kan navigere til fra hvilken scene.

Hierarkiet ble lagd på den måten hovedsakelig på grunn av Don Norman's consistency principle. Dette prinsippet sier at like situasjoner burde ha konsistente handlinger, som her vil si at hierarkiet burde være likt lignende mobilspill. Det er etter egen erfaring veldig typisk at du skal kunne ha hovedmenyen som den plassen du kan navigere til de forskjellige scenene. I tillegg skal det kunne gå an å gå direkte tilbake til hovedmenyen fra hvor som helst.



4.3 GUI

Her skal det det grafiske brukergrensesnittet beskrives.

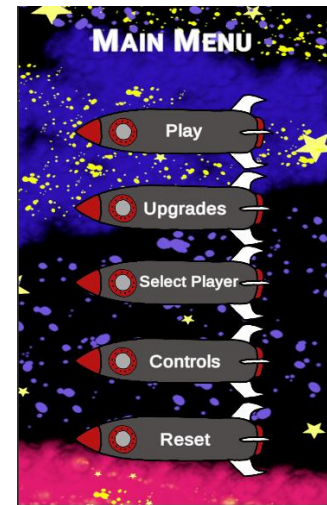
For å passe «space» temaet er bakgrunnen i alle scenene unntatt Deathscreen et bilde av space, og knappene er romskip. Knappene har en animasjon for når de ikke er blitt presset og en som spiller mens den blir presset. Når knappene er urørt roterer knappene fram og tilbake, samtidig som et partikkel-system spiller av en motorflamme fra høyresiden av knappen. Når knappen er presset stopper partikkel-systemet, og størrelsen til knappen blir strekt etterfulgt av presset sammen langs x akse.

4.3.1 Main Menu

Main Menu består av en header på topp med vertikale navigasjonsknapper under.

På skjermbildet over ser du Main Menu. Formålet til denne scenen er at det er her du skal kunne navigere hvor du vil, og denne GUI'en skal fungere som tilgangspunkt for de andre scenene.

På headeren står det Main Menu og fra Main Menu kan du navigere til følgende scener; Play, Upgrades, Player selection, Controls og Reset.

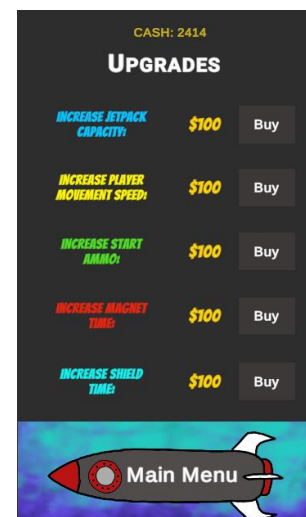


4.3.2 Upgrades

På Upgrades er det en tekst på toppen og under er det en header etterfulgt av 5 rader med 2 tekst objekter og en knapp. I bunnen er det en navigasjonsknapp.

På skjermbildet over ser du Upgrades. Det er denne scenen du skal kunne bruke for å oppgradere ting og gjøre det lettere for spilleren å komme lenger i spillet.

Teksten i toppen viser alle pengene du har samlet, og på headeren står det Upgrades. Radene viser forskjellige oppgraderingsmuligheter du kan velge basert på om du har nok penger samlet opp. Navigasjonsknappen i bunnen fører til Main Menu.

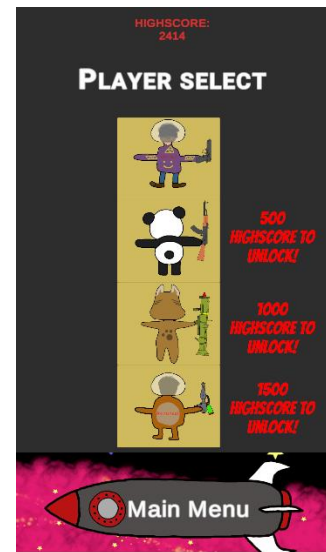


4.3.3 Player selection

Player selection består av en tekst på toppen, header under og 4 rader med en knapp og en tekst. I bunnen er det en navigasjonsknapp.

På skjermbildet ved siden av ser du Player selection. Denne scenen fungerer som scenen du kan velge mellom spillermodeller du har låst opp som du vil bruke.

Teksten i toppen viser hvor mye highscoren din er, og på headeren står det Player select. Radene under er det en knapp med bilde av spillermodeller og en tekst til høyre som står hvor mye highscore som er trengt for å låse den opp. Navigasjonsknappen i bunnen tar deg til Main Menu.

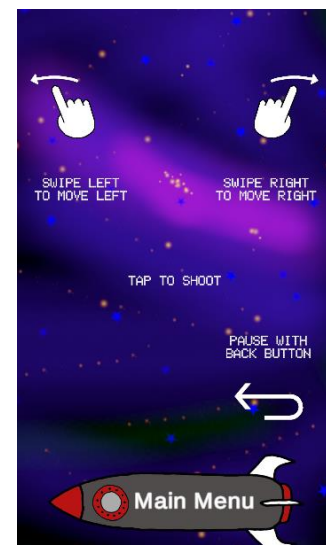


4.3.4 Controls

Controls består av bilder og tekst som forklarer hvordan du spiller spillet.

På skjermbildet ved siden av ser du Controls. Denne scenen skal vise hvordan du kan samhandle med spillet, i tilfelle noen ikke er vant til disse kontrollene.

I bunnen er det en navigasjonsknapp som tar deg til Main Menu.



4.3.5 Reset

Reset består av en tekst i toppen, og to knapper vedsiden av hverandre under.

På bildet ved siden av ser du Reset. På denne scenen kan du resette all brukerdataen som om du har nettopp lastet ned appen.

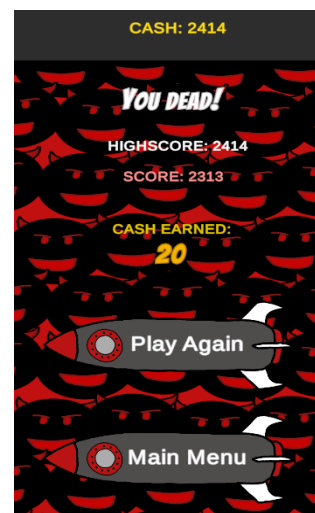
I toppen står det «Are you sure you want to reset your data?» og på knappene står det yes og no. Om du trykker yes resetter du lagringsfilen, om du trykker no går du tilbake til Main Menu.



4.3.6 DeathScreen

På DeathScreen er det en tekst i toppen der det står hvor mye penger du har totalt. Under er det en header der det står «You dead!», etterfulgt av tre tekster der det står highscore, score, og penger tjent den runden. Under der er det to knapper der du kan spille igjen eller gå tilbake til Main Menu.

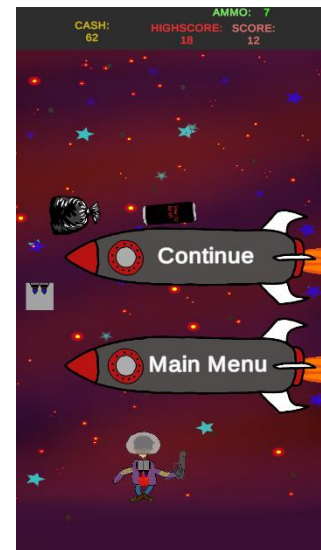
Formålet med denne scenen er å vise spiller først og fremst at de døde, og hva det var du oppnådde før du døde. Vil også gi brukeren muligheten til å gå til hovedmenyen eller spille igjen.



4.3.7 Pause

Pause GUI'en er i samme scene som spill scenen. I Pause blir mesteparten av spill objekter og skript pauset og to knapper dukker opp der du kan unpause eller navigere til Main Menu.

Formålet med pause scenen er at hvis brukeren trenger å stoppe å spille for å gjøre noe annet en stund, skal de kunne gå tilbake til hvor de var før de måtte stoppe.

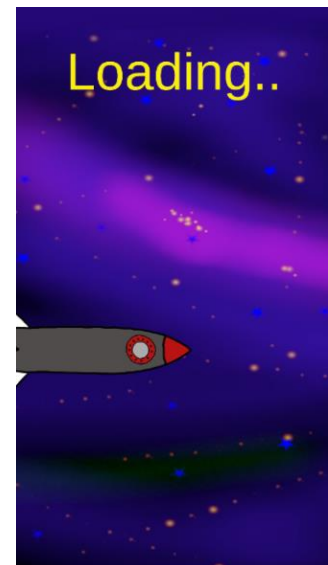


4.3.8 Loading screen

Loading screen er i samme scenen som Main Menu, og dukker opp over Main Menu når du trykker på Play, og forsvinner når game scenen har blitt ferdig lastet.

Loading screen består av en header i toppen der det står «Loading...» og under er det animasjon som spilles der et romskip reiser fra venstre til høyre langs x akse igjen og igjen til game scenen er klar.

Selve Loading screen er laget på grunnlag av Don Norman's feedback principle, for å gi brukeren en tilbakemelding på hva som foregår i bakgrunnen etter at du har gjort en handling.



4.4 Score system

Score blir økt over tid fra du begynner å spille, og score blir også ganget med en multiplier som også blir økt over tid. Dette gjør at score øker raskere og raskere jo lenger du klarer å overleve.

Her er det også designet ut ifra Don Norman's consistency principle, ettersom at denne løsningen er veldig vanlig denne type spill.

4.5 Lagringssystem

Lagringssystemet fungerer slik at hver gang du oppgraderer, endrer spiller, får ny highscore eller tjener mer cash blir det lagret i lagringsfilen. Lagringsfilen blir lastet inn hver gang du starter spillet, og hver gang lagringsfilen blir oppdatert.

Måten data blir lagret er ved hjelp av BinaryFormatter [13] og FileStream [12] API'er. Alt som blir lagret i lagringsfilen er en variabel i et script i et game object som eksisterer i hver scene. Hver gang du lagrer noe blir verdiene fra de variablene lagret i lagringsfilen, og hver gang du laster inn filen, blir verdiene i lagringsfilen lastet inn i de variablene.

4.6 Oppgraderinger

Måten oppgraderinger fungerer er at når du trykker på en oppgradering, blir det sjekket hvilket nivå oppgradering du har av den typen ved å sjekke variablene til oppgraderingsscriptet. Etter det blir variabelen endret basert på hvilken oppgradering du har kjøpt, samtidig som det blir trukket fra penger avhengig av prisen. Etter det blir lagringsfilen lagret og lastet inn på nytt.

4.7 Kontroller

Kontrolleren fungerer slik at om du swiper mot venstre går spilleren et hakk til venstre, og om du swiper til høyre går spilleren et hakk til høyre. Du kan skyte ved å tappe skjermen, og du kan pause ved å trykke android tilbake knappen.

Måten det blir sjekket om spiller har swipet eller tappet og i hvilken retning, er ved å sjekke posisjonen til touch når finger får kontakt med skjermen, og sjekke posisjonen til touch når finger løftes fra skjermen. Du ut ifra det kan du regne ut om du swipet til høyre, venstre eller om det var en tap.

Denne løsninger er lagd på denne måten basert på følgende prinsipper:

- Don Norman's consistency principle – Denne typen løsningen er veldig standard for lignende spill som gjør at brukeren er mer sannsynlig til å være vant til denne typen kontroller.
- Don Norman's mapping principle – Swiping og tapping er i tillegg til å være ganske standard, veldig naturlig etter min erfaring, som vil si at det kan bli veldig komfortabelt for bruker å bruke denne type kontroller.

4.8 Testing

For å teste hvor lett det er å bruke spillet i tillegg til å finne mulige godt gjemte bugs, Lot jeg 4 folk få teste spillet. 2 var i aldersgruppen 40 - 60 år med lite spillerfaring, og 2 i aldersgruppen 15 - 25 år.

Resultatene fra testen var veldig nyttig på grunn av at det ble funnet mange bugs som var vanskelig å finne selv, på grunn av at når jeg testet spillet selv, brukte jeg spillet på måten det var ment å spille. I tillegg til bugs som ble funnet, fant jeg ut at folk i 40 - 60 års alderen fant spillet veldig vanskelig å bruke. Etter dette ble oppdaget laget jeg scenen Controls, der du kan se hvordan du spiller.

4.9 Publisering

Når jeg begynte å komme meg mot slutten av det jeg hadde planlagt skulle vært gjort, merket jeg at jeg hadde litt tid igjen. Det var det jeg bestemte meg for at det kunne vært lærerikt å få spillet publisert på Google Play Store. Så jeg begynte med å bygge en google build for å publisere. Etter det registrerte jeg meg som utvikler på Google og fylte ut all appinformasjonen som var trengt for å publisere. Hvordan dette ble gjort kan du se under Google Play i teori-delen.

4.9.1 Reklame

Etter dette bestemte jeg meg for å legge til reklame i appen. Dette ble gjort ved hjelp av Admob [19] på grunn av at det så ganske enkelt ut å integrere med Unity. Måten jeg implementerte reklame i min app, er at hver gang du kommer inn i hovedmenyen etter at du har vært i spill scenen, blir det vist en reklame. Det ble gjort på denne måten på grunn av etter egen erfaring, blir spill med for mye reklame avinstallert veldig fort om det er for mye reklame.

5 DRØFTING

5.1 Inspirasjon

Under idé-fasen tenkte jeg mye på hvilke mobilspill jeg liker best, og fant ut at det de spillene hadde til felles, er at jeg ikke trenger å tenke særlig mye og at det er enkelt komme inn i. I tillegg hadde spillene en «sense of progress» som vil si at du må kunne føle at du blir bedre eller kommer deg videre i spillet. De vanlige løsningene kunne være enten oppgraderinger av spiller eller at du låste opp nye nivåer. Så når det kommer til mobilspill så tar jeg for meg følgende hovedkriterier:

- Grafikk og animasjon – Det visuelle i spillet og hvordan objekter beveger seg.
- «Sense of progress» - At bruker kan føle at de kommer seg videre i spillet når de gjør det bra.

- Lett å lære og spille – At det er lett og naturlig for bruker å sette seg inn i spillet.
- Design av brukergrensesnitt – At brukergrensesnittet ser bra ut og at det er lett å bruke og forstå.

5.2 GUI

Alle navigasjonsknappene er lagd med tanke på grunn av Don Norman's consistency og mapping principle. Disse prinsippene er nevnt her på grunn av at alle navigasjonsknappene ser helt like ut, noe som bidrar til å gi brukeren et klart inntrykk av at knapper som ser slik ut er en navigasjonsknapp. Dette går innpå mapping prinsippet på grunn av at denne konsistensen gir brukeren et klart inntrykk av at når du trykker på knappen vil det lede til en navigasjon til en annen scene.

Don Norman's feedback principle er også blitt utnyttet i navigasjonsknappene ved at mens du trykker på knappen spilles en annen animasjon enn når knappen er urørt. I tillegg til dette er også gestalt prinsippene blitt utnyttet her. Spesifikt similarity og common fate prinsippet, ettersom knappene er helt like og beveger seg likt.

For hver GUI-scene er det brukt følgende design prinsipper for å lage brukergrensesnittet:

- Don Norman's visibility principle – Har brukt farger som gjør at du kan se lett se og bruke det. Upgrades og Player selection så kan du se dette i action på grunn av det svarte panelet bak teksten, som måtte utnyttes for å kunne se de forskjellige fargede tekstene.
- Don Norman's consistency principle – På grunn av mange folks tidligere erfaringer med slike spill, er det lettere for brukere å forstå hvilken funksjon elementene har.
- Gestalt-prinsippet Figure/Ground – Går ut på at du kan lett skille mellom det i bakgrunnen og foreground.

5.3 «Sense of progress»

Her tenkte jeg først at det skulle bli flere nivåer med forskjellige hinder og effekter, men etter det ble lagd et par nivåer skjønte jeg fort at det trengtes veldig mye arbeid til for å få dette til å funke. Noen utfordringer med denne løsningen var:

- Grafikk, animasjon og spesialeffekter – å lage nye nivåer betydde at jeg måtte lage flere skyboxer, hinder, spesialeffekter som eksplosjoner etc, animasjoner til hvert hinder.

- Vanskelighetsgrad – Løsningen krever at for hvert nivå måtte jeg justere vanskelighetsgraden for flere nivåer, og dette vil si i praksis å bruke lang tid på å gå over alle variablene som påvirker hvor vanskelig spiller oppfatter nivået for hvert nivå. Dette kan ta veldig lang tid.
- Konsept – Ideelt sett ville jeg at hvert nivå skulle ha et eget konsept, noe som gjorde at måten du spiller for hvert nivå kan være drastisk forskjellig, og kunne krevet nesten like mye som å lage et nytt spill for hvert nivå. I tillegg til det kunne det ta veldig lang å komme opp med idéer som faktisk var i min mening brukbare.

Så etter det ble lagd et par nivåer skjønnte jeg at dette ville enten ta for lang tid eller bli mye dårligere kvalitet en ønsket. Så jeg bestemte meg for å lage et nivå, der vanskelighetsgrad og hinder endret seg over tid. Sense of progress løsningen ble løst ved å legge til mulighet for å kjøpe oppgraderinger, i tillegg til å kunne låse opp forskjellige spillermodeller med egne våpen.

Denne løsningen tror jeg er helt grei, men hadde vært litt mer fornøyd om jeg kunne ha lagt til «missions», oppdrag som kunne blitt gjort for ekstra cash eller mer score eller noe sånt i tillegg. Men i forhold til tiden jeg hadde er jeg ganske fornøyd med denne løsningen.

5.4 «Customization»

Når jeg begynte å tenke på dette, tenkte jeg først at dette kunne bli gjort bare ved å ha en character der du kunne endre fargen på klærne for eksempel. Når tiden kom for å implementere dette, så jeg at dette ble for kjedelig, og at jeg i tillegg ville ha flere våpen med forskjellige våpen jeg kunne låse opp. Så jeg valgte å lage et par forskjellige characters med egne våpen du kunne låse opp.

Jeg er ganske fornøyd med løsningen jeg kom opp med. Eneste jeg er litt misfornøyd med er at spiller modellene og andre effekter kunne vært litt bedre, men siden jeg ikke har så mye erfaring med dette, tenker jeg at det gikk så godt som forventet.

5.5 Konsept

I idé-fasen tenkte jeg at konseptet skulle være at astronauten faller ut av romskipet i fart, og at spilleren måtte prøve å ta igjen romskipet (noe som vil aldri skje) mens spilleren unngår forskjellige hinder etc. Men når jeg begynte å implementere det, virket det ikke veldig realistisk på grunn av følgende:

- Hvis romskipet har en konstant fart og motoren ikke er på, vil spilleren lett kunne ta igjen romskipet med jetpack ettersom at det ikke er noen motstand i rommet.

Hvis spiller faller ut av romskipet og har ingen motstand, har spiller samme fart som romskipet, og hvis spiller bruker jetpack, akselerer spiller hele tiden og når aldri noe maksimal fart.

- Hvis romskipet har motoren på hele tiden, vil spiller aldri ta igjen romskipet med en jetpack.

Etter refleksjon over dette, tenkte jeg at et spill trenger jo egentlig ikke være realistisk. Et spill er bare lagd for å være gøy, og ikke noe simulasjon av virkeligheten. Så jeg valgte å bare ignorere realistikheten med konseptet og gå framover med opprinnelig konsept.

For å være helt ærlig ville jeg ønsket å ha et bedre konsept, men siden det ble oppdaget så sent i prosessen var det egentlig litt for sent å endre det helt. Det jeg har lært fra dette er å bruke litt mer tid på å planlegge prosjektet for jeg begynner.

5.6 Kontroller

I starten tenkte jeg at det kunne være en god ide å ha en slags joystick på skjermen der du kunne dra til venstre eller høyre for å bevege deg. Jeg implementerte dette og det funket helt fint, men det tok opp litt plass på skjermen og når jeg lot venner teste det, synes de at det var vanskelig å bli vant til. Så jeg istedenfor implementerte en annen løsning der bruker bare trenger å «swipe» finger mot høyre for å gå til høyre eller «swipe» mot venstre for å gå til venstre. For å skyte er det bare å «tappe».

Jeg er veldig fornøyd med denne løsningen på grunn av at denne løsningen er veldig standard i lignende spill, i tillegg til at dette føles veldig naturlig og for mange mobilspill brukere føler jeg at dette vil være første-instinkt til brukere uten å se på controls scenen.

5.7 Problemstillingene

Her vil jeg forklare hvordan problemstillingene jeg satt for meg i innledningen ble håndtert.

5.7.1 Hvordan lage et spill som folk vil spille framfor alle de andre tusenvis av android spill?

Selvfølgelig vet vi at alle folk er forskjellige og liker forskjellige ting, så måten jeg taklet denne problemstillingen var først og fremst limitere målgruppen til folk som generelt liker «endless runner» spill. Bortsett fra det, fokuserte jeg på å legge inn det som jeg ville likt å ha i et «endless runner»-spill. Utenom det er det folk fokuserer på mest om musikk, animasjon, grafikk og etc, men ettersom det ikke er noe jeg er særlig god på, så jeg la jeg mer fokus på spill-temaet og andre elementer.

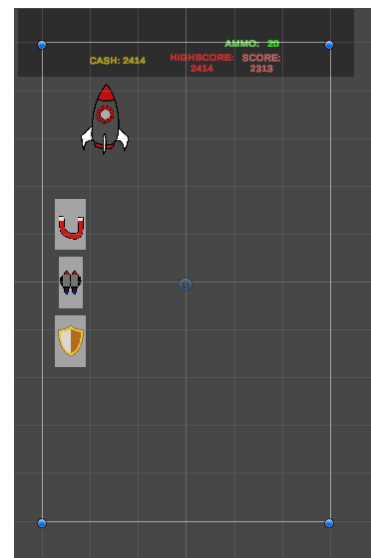
5.7.2 Hvordan lage et spill så intuitivt og enkelt samtidig som det skal være underholdende?

Når det kommer til et intuitivt og enkelt å forstå spill, betyr det først og fremst at kontrollere og navigasjon-hierarkiet er veldig viktig. Så ut ifra det fokuserte jeg på at kontrollere skulle være veldig kontrollere i andre «endless runner» spill, der standarden er du swiper mot høyre for å gå mot høyre og vice versa, i tillegg til tapping for skyting eller lignende. Etter testing forsto jeg også at folk som ikke var veldig vant til mobil-spill, var det ikke veldig klart. Så jeg la også til en scene der du kan se hvordan du spiller.

Måten jeg taklet navigasjons-hierarkiet er at jeg fokuserte på å følge standarden, i tillegg til å gjøre det veldig klart hvilken knapp gjorde hva. Standarden her er at du har en hovedmeny som fungerer som plassen du kan navigere til de fleste scenene, i tillegg til at du skal kunne gå tilbake til hovedmenyen fra hvilken som helst scene.

5.7.3 Hvordan gjøre spillet kompatibelt med de mange forskjellige oppløsningene som er tilgjengelig for android?

Her limiterte jeg først og fremst enhetsmålgruppen til mobiler på grunn av de mange forskjellige oppløsningene tilgjengelig om jeg skulle tilrettelegge til alt annet. Måten jeg gjorde det kompatibelt for alle mobil oppløsningene var at jeg først fant ut hvilken oppløsning som passet til de fleste mobiler, som jeg fant ut etter testing og research var 1600 * 900. Etter testet jeg spillet på mange forskjellige oppløsninger på en android simulator og fant hvor langt jeg måtte strekke UI elementer for at det skulle se bra ut på alle oppløsningene. Som du kan se på bildet under, har jeg strukket ut panelet i toppen ut av det brukeren kan se for å tilrettelegge for mobiler med høyere oppløsning.



6 KONKLUSJON

jeg kunne ha gjennomført dette prosjektet mye bedre om jeg hadde brukt mer tid på å planlegge detaljene før jeg begynte. Men med tanke på tiden jeg hadde på prosjektet og at dette er det første prosjektet jeg hadde gjennomført på denne måten, føler jeg at det gikk veldig som forventet.

Hver gang det oppstod en uforutsett problemstilling prøvde jeg å komme opp med en god løsning fortløpende, noe som gjorde at jeg lærte mye om den agile metoden og hvorfor den agile metoden er god når det kommer til disse prosjektene. Jeg lærte også hvor viktig det er å planlegge godt i startfasen.

I løpet av prosjektet lærte jeg også veldig mye om Scrum, den agile metoden og verktøyet Jira. Jeg lærte også at det er veldig nyttig å skrive ned masse notater i løpet av prosjektet, i tillegg til at det er viktig å dokumentere koden selv om du jobber alene. Når jeg jobber med mange forskjellige script uten dokumentasjon, kan det være lett å glemme hvorfor du gjorde ting på den måten og bli forvirret. Samme går for notatene, Når jeg jobber med så mange idéer er det viktig å notere ned hvorfor og hvordan de skal gjennomføres for ikke å bli forvirret senere i prosessen når du må gå over noe.

7 REFERANSER

- [1] Endless Runner – Sample Game, 06.02.2020, <https://assetstore.unity.com/packages/essentials/tutorial-projects/endless-runner-sample-game-87901>
- [2] Don Normans principles of interaction design, <https://medium.com/@sachinrekhi/don-normans-principles-of-interaction-design-51025a2c0f33>
- [3] Gestalt-prinsipper, <https://utforsksinnet.no/gestaltprinsipper-hvordan-organiserer-vi-det-vi-ser/>
- [4] Agile metoder, <https://stackify.com/agile-methodology/>
- [5] Scrum, <https://www.atlassian.com/agile/scrum>
- [6] Unity, <https://conceptartempire.com/what-is-unity/>
- [7] Bakgrunnsmusikk, <http://www.nosoapradio.us/>
- [8] Audacity, <https://www.audacityteam.org/about/>

- [9] JFXR, <https://jfxr.frozenfractal.com/>
- [10] Jira, <https://www.atlassian.com/software/jira>
- [11] Confluence, <https://www.atlassian.com/software/confluence>
- [12] FileStream, <https://docs.microsoft.com/en-us/dotnet/api/system.io.filestream?view=netcore-3.1>
- [13] BinaryFormatter, <https://docs.microsoft.com/en-us/dotnet/api/system.runtime.serialization.formatters.binary.binaryformatter?view=netcore-3.1>
- [14] Random, <https://docs.unity3d.com/ScriptReference/Random.Range.html>
- [15] Raycast, <https://docs.unity3d.com/ScriptReference/Physics.Raycast.html>
- [16] Gimp, <https://www.gimp.org/>
- [17] Blender, <https://www.blender.org/>
- [18] Google play, <https://developer.android.com/distribute/best-practices/launch/launch-checklist>
- [19] Admob, <https://developers.google.com/admob/unity/start>

VEDLEGG

Vedlegg 1	Unity prosjektet
Vedlegg 2	Forprosjektrapport
Vedlegg 3	Sprint retrospectives
Vedlegg 4	Sprint review notes

FORPROSJEKT - RAPPORT

FOR BACHELOROPPGAVE



Kunnskap for en bedre verden

TITTEL:

«Space Chase»

KANDIDATNUMMER(E):

Asbjørn Frostad

DATO:	EMNEKODE: *	EMNE:	DOKUMENT TILGANG:
	IB303312 IE303612	Bacheloroppgave (Bygg) Bacheloroppgave (Data)	- Åpen
STUDIUM:	ANT SIDER/VEDLEGG:		BIBL. NR:
	/		• Ikke i bruk -

OPPDRAGSGIVER(E)/VEILEDER(E):

Arne Styve

OPPGAVE/SAMMENDRAG:

Formålet med prosjektet er å lage et underholdende og enkelt spill, som hvem som helst, når som helst kan ha det gøy med. Dette spillet handler om en astronaut som faller ut av romskipet sitt i fart, og derfor må prøve å ta igjen romskipet sitt ved å bruke jetpack'en sin til å unngå og/eller ødelegge hinder i sin vei.

Denne oppgaven er en eksamensbesvarelse utført av student(er) ved NTNU i Ålesund.

Postadresse
NTNU i Ålesund
Postboks 1517
N-6025 Ålesund

Besøksadresse
Larsgårdsvegen 2
Internett
www.ntnu.no

Telefon
70 16 12 00
Epostadresse
postmottak@ntnu.no

Telefax
70 16 13 00

Bankkonto
7694 05 00636
Foretaksregisteret
NO 947 767 880

INNHold

1 INNLEDNING	3
2 BEGREPER	3
3 PROSJEKTORGANISASJON	3
3.1 PROSJEKTGRUPPE.....	3
3.1.1 <i>Oppgaver for prosjektgruppen - organisering</i>	3
3.2 STYRINGSGRUPPE (VEILEDER OG KONTAKTPERSON OPPDRAGSGIVER).....	3
4 AVTALER	4
4.1 AVTALE MED OPPDRAGSGIVER.....	4
4.2 ARBEIDSSTED OG RESSURSER.....	4
4.3 GRUPPENORMER – SAMARBEIDSREGLER – HOLDNINGER.....	4
5 PROSJEKTBESKRIVELSE	4
5.1 PROBLEMSTILLING - MÅLSETTING - HENSIKT.....	4
5.2 KRAV TIL LØSNING ELLER PROSJEKTRESULTAT – SPESIFIKASJON.....	4
5.3 PLANLAGT FRAMGANGSMÅTE(R) FOR UTVIKLINGSARBEIDET – METODE(R).....	5
5.4 INFORMASJONSINNSAMLING – UTFØRT OG PLANLAGT.....	5
5.5 VURDERING – ANALYSE AV RISIKO.....	6
5.6 HOVEDAKTIVITETER I VIDERE ARBEID.....	6
5.7 FRAMDRIFTSPLAN – STYRING AV PROSJEKTET.....	7
5.7.1 <i>Hovedplan</i>	7
5.7.2 <i>Styringshjelpemidler</i>	8
5.7.4 <i>Utviklingshjelpemidler</i>	8
5.7.5 <i>Intern kontroll – evaluering</i>	9
5.8 BESLUTNINGER – BESLUTNINGSPROSESS.....	9
6 DOKUMENTASJON	9
6.1 RAPPORTER OG TEKNISKE DOKUMENTER.....	9
7 PLANLAGTE MØTER OG RAPPORTER	9
7.1 MØTER.....	9
7.1.1 <i>Møter med styringsgruppen</i>	9
7.2 PERIODISKE RAPPORTER.....	10
7.2.1 <i>Framdriftsrapporter (inkl. milepæl)</i>	10
8 PLANLAGT AVVIKSBEHANDLING	10
9 REFERANSER	10

1 INNLEDNING

Valgte å la et 2D spill for Android i Unity på grunn av tidlige personlig erfaring i Unity, og fordi dette er et prosjekt jeg føler jeg kan holde meg motivert med.

Dette spillet handler om en astronaut som faller ut av romskipet sitt i fart, og derfor må prøve å ta igjen romskipet sitt ved å bruke jetpack'en sin til å unngå og/eller ødelegge hinder i sin vei.

Formålet med prosjektet er å lage et underholdende og enkelt spill, som hvem som helst, når som helst kan ha det gøy med. Også for å få bacheloren min.

2 BEGREPER

3 PROSJEKTORGANISASJON

3.1 Prosjektgruppe

Studentnummer(e)
Asbjørn Frostad

3.1.1 Oppgaver for prosjektgruppen - organisering

Siden jeg er den eneste på dette prosjektet, kommer jeg til å gjøre alle oppgavene selv.

3.2 Styringsgruppe (veileder og kontaktperson oppdragsgiver)

Veileder – Arne Styve

4 AVTALER

4.1 Avtale med oppdragsgiver

Dette prosjektet har ikke en ekstern arbeidsgiver. For dette prosjektet er jeg min egen arbeidsgiver.

4.2 Arbeidssted og ressurser

Tilgang til arbeidsplass på skolen og hjemme.

Rapportering annenhver uke mandag kl. 09:00.

4.3 Gruppenormer – samarbeidsregler – holdninger

I dette prosjektet vil jeg strebe etter å:

- Jobbe minst 8 timer hver ukedag.
- Holde alt godt dokumentert
- Reflektere og justerte arbeids-mengde og/eller -prosess etter hver sprint.
- Distribuerer arbeid mellom sprinter så godt som mulig, som formål å jobbe så konsistent som mulig gjennom prosjektets levetid.

5 PROSJEKTBSKRIVELSE

5.1 Problemstilling - målsetting - hensikt

Formålet med prosjektet er å lage et underholdende og enkelt spill, som hvem som helst, når som helst kan ha det gøy med. Dette spillet handler om en astronaut som faller ut av romskipet sitt i fart, og derfor må prøve å ta igjen romskipet sitt ved å bruke jetpack'en sin til å unngå og/eller ødelegge hinder i sin vei.

I forhold til arbeidsprosess, er målet å holde alt godt dokumentert, og holde arbeidsmengden linear gjennom prosjektets levetid.

5.2 Krav til løsning eller prosjektresultat – spesifisering

Spesifikasjoner for en ferdig løsning er:

- Funksjonell player-controller
- Interaktive hinder etc i spill.
- Funksjonell main- og pause-menu.
- Funksjonell score system.
- Lyd effekter.
- Animasjoner.
- Spesialeffekter som eksplosjoner etc.
- Character modeller og etc grafikk.
- At man kan lage informasjon som innstillinger og score etc lokalt på telefonen.
- Minst et funksjonelt nivå.

Dette prosjektet burde ikke koste noe.

for at dette prosjektet skulle blitt sett på som en suksess, spiller må kunne bevege og være interaktiv med character. Det er også behov for minst et funksjonelt nivå inkludert hinder, utfordringer og et score system. Det må også være noen form for grafikk, animasjoner og spesialeffekter. Å lagre informasjon som innstillinger og score etc lokalt på telefonen, er også nødvendig.

Dette spillet skal utvikles for Android.

5.3 Planlagt framgangsmåte(r) for utviklingsarbeidet – metode(r)

Dette prosjektet skal bruke Scrum som bruker sprints. En sprint er en kort, tidsdefinert periode når et team (eller individ) jobber for å bli ferdig med et sett mengde arbeid. I slutten av sprinten skal det være en sprint review og retrospective. I Sprint review går du gjennom sprinten og hva du har

gjort og hvorfor du har gjort det. I Retrospective reflekterer du om hvorfor det har gått som det gikk og hva du kan gjøre for å gjøre det bedre.

Styrker:

Scrum er bra for prosjekt hvor det er mange usikkerheter. I en sånn situasjon vil du ikke sette faste tidsgrenser fordi uforventede problemer er nesten garantert til å dukke opp, og når du bruker smidige metoder kan du enkelt bare legge til endringer i neste iterasjon.

Svakheter:

Siden Scrum fokuserer mye på direkte verbal kommunikasjon og spesifisering fleksibilitet, er det vanskelig å integrere dette inn i store organisasjoner.

Muligheter:

På grunn av iterasjoner i smidige metoder, kan du på når som helst tidspunkt endre ting eller implementere nye design og ideer.

Trusler:

Med smidige metoder, er det egentlig ikke mye standard når det kommer til sikkerhetsstruktur, som kan lede til sikkerhets problemer.

Svakhetene og truslene her er ikke helt relevant for dette prosjektet på grunn av størrelsen på prosjektet og nødvendigheten for sikkerhet er ikke så viktig.

I dette prosjektets tilfelle vil det være en ny sprint annenhver uke, da det skal være et sprint review, retrospective og skal planlegge hva som skal være gjort neste sprint.

For å implementere Scrum lettere, skal prosjektet ta i bruk Jira for å holde sprint osv oversiktlig. Bruker også NTNU Wiki for å lagre dokumenter relatert til sprint reviews og retrospective etc.

5.4 Informasjonsinnsamling – utført og planlagt

Unity har en veldig godt dokumentert manual som vil bli brukt til å samle informasjon, feilsøke eller finne mulige løsninger. Link i referanser.

Ettersom Unity er veldig populært og har et stort og aktivt community på forumer, kommer jeg mest sannsynlig til å finne mesteparten av informasjonen og/eller støtte fra forumer.

5.5 Vurdering – analyse av risiko

Dette prosjektets spesifikasjoner er enkle og klare, men målet og spesifikasjonene vill sannsynligvis bli oppdatert etter hvert prosjektet går framover, og så lenge som målet og spesifikasjonene blir holdt realistiske, prosjektet kan ha en god sjanse på suksess.

På grunn av de mange forskjellige måtene du kan gjøre dette prosjektet, er det mange aspekter som ikke er klar i dette stadiet. De uklare aspektene av prosjektet må bli klarere etterhvert som prosjektet går framover.

De viktigste aspektene for suksess:

- *Tidsstyring*
 - o I et prosjekt som dette, er det nesten garantert til å oppstå mange problemer og oppgaver som du ikke var forventet. For å kjempe dette er det viktig å forvente at oppstår uforventede problemer fra begynnelsen, og distribuere arbeidsmengden som så.
- *Realistiske mål*
 - o I prosjekt som dette, er det endeløse ting du kan gjøre for å forbedre produktet. I en sånn situasjon er det viktig å holde målene realistisk og oppdatert.
 - o Om du er for ambisiøs i planleggingen kan du ende opp med å bruke for lang tid på aktiviteter som ikke er essensielle, og gå tom for tid uten å ha full funksjonalitet.

Risiko:

- Mister arbeid på grunn av harddisk failure etc.
 - For redusering av skade i et sånt tilfelle, vil jeg lagre all dokumentasjon online på Google drive, og selve spillet og dens assets i github.
- Pc-en svikter helt.
 - For å være klar for et sånt tilfelle har jeg tilgang til 2 andre pc-er. Ettersom alt blir lagret online, kan jeg arbeide på hvilken som helst pc.
- Veileder utilgjengelig
 - I et sånt tilfelle må jeg bare være klar til å gå gjennom og dokumentere sprint reviews og retrospective selv eller evt med andre involverte. Om jeg har et viktig spørsmål må jeg bare få tak noen andre for råd.
- Sykdom eller ikke i stand til å arbeide av andre grunner.
 - For å forebygge dette må jeg prøve å forutse dette på forhånd og prøve å legge meg litt i forkant på arbeidet. I tillegg til å prioritere hindring av sykdom eller andre grunner som gjør meg utilgjengelig.

Ettersom dette spillet vil bare lagre informasjon lokalt på telefonen, vil spillet ikke trenge noe særlig sikkerhet annet en hash. Hvis noen klarte å manipulere systemet ville det ikke ha noe konsekvens for noen andre enn seg selv.

5.6 Hovedaktiviteter i videre arbeid

Nr	HovedAktivitet	Kost	Tid/Arbeid
A1	Storyboard og etc spill design.	0	5 – 10%
A2	Programmere player-controller	0	5 – 15%
A3	Lage character modell	0	5 – 15%
A4	Lage annet grafikk (skybox, asteroids etc)	0	15 – 25%
A5	Lage animasjoner og spesialeffekter	0	20 – 30%
A6	Nivå design	0	10 – 20%
A7	Lage lydeffekter og etc lyd	0	5 – 15%
A8	Implementering og testing.	0	10 – 20%

Tidsrammen av disse aktivitetene har et høyt nivå av usikkerhet, fordi det er umulig å forutse alle utfordringene og problemene som vil oppstå før aktivitetene har begynt.

Alle ressursene du trenger for disse aktivitetene et en arbeidsplass (pc). På dette stadiet, er det ingen kostnader forutsett for disse aktivitetene. Tidsrammen for dette prosjektet vil være 4 måneder fra prosjektets start.

5.7 Framdriftsplan – styring av prosjektet

5.7.1 Hovedplan

Ettersom prosjektet vil bruke smidige metoder, er start- og slutt-tid for disse aktivitetene for det meste usikker til de er inkludert i en iterasjon.

Aktiviteter:

- Programmere player-controller
 - Dette går ut på å få spilleren til å være i stand til å bevege og være interaktiv med character.
 - Siden player-controller'en er veldig nyttig i tidlige stadier for å samhandle og teste med andre objekter i spillet, vil denne aktiviteten være nødvendig i begynnelsen av prosjektet.
- Lage character modeller
 - Dette går ut på tegne hvordan character kommer til å se ut og separere delene som kommer til å trenge å bevege seg uavhengig i animasjon-fasen.

- Det visuelle er normalt ikke helt nødvendig før slutten av prosjektet, men i min erfaring kommer denne aktiviteten vanligvis med en del uforutsette problemer som trenger oppmerksomhet. Character modellen er også nødvendig for å starte animasjons-fasen, så dette burde være gjort ganske tidlig i prosjektet.
- Lage annen grafikk
 - Dette går ut på å tegne skybox (bakgrunn), asteroider og andre hindre, romskip etc.
 - Dette burde vært startet på rundt midt i prosjektet og bli jobbet med konsistent gjennom prosjektet. Det tar vanligvis ikke lang tid per figure, men du trenger vanligvis mye mer enn du planlegger.
- Lage animasjoner og spesialeffekter
 - Dette går ut på å designe og animere hvordan character og spaceship etc skal bevege seg gjennom flere typer bevegelser. Spesialeffektene er normalt designet gjennom Unity komponenter og litt kode for når det skal aktiveres etc.
 - Dette er veldig viktig for at spillet skal virke reaktiv og morsomt spillet vil se ut og vil ta litt tid, men dette er ikke essensielt til funksjonalitet av spillet. Denne aktiviteten burde ta plass i rundt midt til slutt stadiene av prosjektet når de mest essensielle aktivitetene er blitt gjort.
- Designe nivå
 - Dette går ut på å planlegge og plassere hinder og andre samhandlingsbare objekter strategisk i spillet, med formål å gjøre det så reaktivt, utfordrende og morsomt som mulig, samtidig som du gir spilleren sense of accomplishment når de klarer det.
 - Dette kan ta sin tid og burde bli jobbet på konsistent gjennom prosjektet.
- Lage lydeffekter og annen lyd.
 - Dette går ut på å lage lyd for når spilleren gjøre noe, sånn som hopping eller dør. I tillegg til å lage bakgrunnslyd eller mer.
 - Dette er ikke nødvendig før i slutten av prosjektet, men kan ta en stund, så burde bli begynt på tidlig i slutt stadiet av prosjektet.

Alle disse aktivitetenes tidsramme kan bli endret når som helst, avhengig av om de er nødvendig for å begynne en annen aktivitet.

5.7.2 Styringshjelpemidler

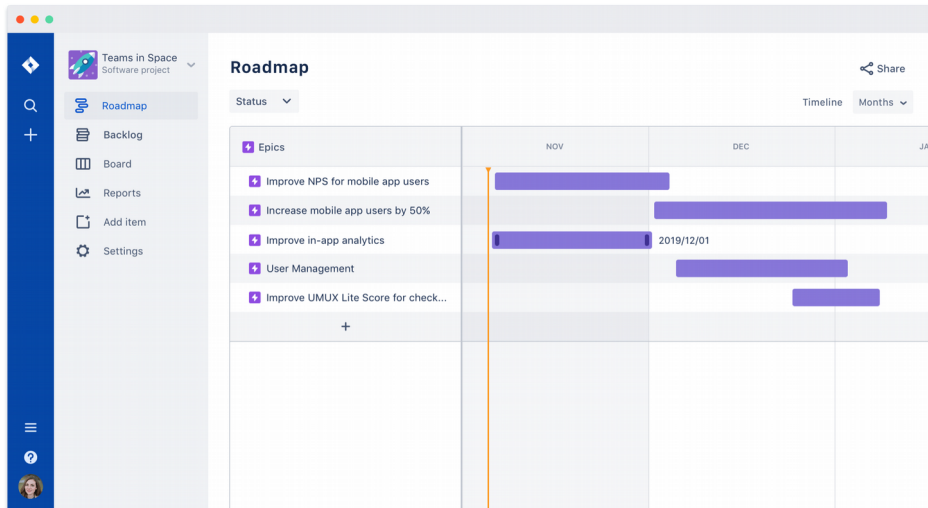
I dette prosjektet planlegger jeg å bruke Jira (Issue & Project Tracking Software) til å holde oversikt over mine aktiviteter og problemer, i tillegg til at det er et verktøy for smidig utvikling.

Kommer også til å bruke NTNU Wiki for å lagre dokumentasjon fra sprint reviews og retrospective.

(Kan ikke logge in til Jira for å få illustrasjoner enda, så her er noen fra Atlassian.com)

The screenshot shows a Jira board with four columns: TO DO (3 issues), IN PROGRESS (3 issues), IN REVIEW (2 issues), and DONE (3 issues). A filter 'Super important' is applied, showing 11 issues. The issues are:

- TO DO:**
 - When requesting user details the service should return prior trip info (SEESPACEEZ PLUS, TIS-37)
 - Engage Jupiter Express for outer solar system travel (SPACE TRAVEL PARTNERS, TIS-25)
 - Create 90 day plans for all departments in the Mars Office (LOCAL MARS OFFICE, TIS-12)
- IN PROGRESS:**
 - Requesting available flights is now taking > 5 seconds (SEESPACEEZ PLUS, TIS-8)
 - Register with the Mars Ministry of Labor (LOCAL MARS OFFICE, TIS-11)
 - Engage Saturn Shuttle Lines for group tours (SPACE TRAVEL PARTNERS, TIS-20)
- IN REVIEW:**
 - Register with the Mars Ministry of Revenue (LOCAL MARS OFFICE, TIS-13)
 - Draft network plan for Mars Office (LOCAL MARS OFFICE, TIS-49)
- DONE:**
 - Add pointer to main css file to instruct users to create child themes (LARGE TEAM SUPPORT, TIS-56)
 - Homepage footer uses an inline style - should use a class (LARGE TEAM SUPPORT, TIS-68)
 - Engage JetShuttle SpaceWays for travel (SPACE TRAVEL PARTNERS, TIS-23)



5.7.3 Utviklingshjelpemidler

Dette prosjektet kommer hovedsaklig til å bruke game engine Unity og visual studio code editor.

For å utvikle grafikk det bli brukt Gimp. For lyd kommer det til å bli brukt Audacity og for de små lydeffektene som hopping, kommer det til å bli brukt en web-side kalt Leshylabs til å generere små lydeffekter. Link i referanser.

5.7.4 Intern kontroll – evaluering

Siden dette prosjektet bruker Scrum, vil det bli gjort sprint review og retrospectives i slutten av hver sprint, som i dette tilfelle er annenhver uke.

For at et mål skal bli sett som nådd, trenger ikke resultatet være perfekt, men det trenger å være funksjonelt på måten det var planlagt. Hvis tiden brukt til å nå målet er uakseptabelt eller ikke kan ikke bli bestemt før målet er nådd, på grunn av nivået av uforutsigbarheten i dette prosjektet.

5.8 Beslutninger – beslutningsprosess

Siden jeg er det eneste medlemmet i dette prosjektet har jeg, og vil gjøre alle viktige valgene selv. Viktige ting å tenke på når jeg gjør et valg er om prosjektet vil være funksjonelt og oppfyller spesifikasjonene på den spesifiserte tiden, enn om det vil ha mange features eller hvor morsomt det vil være. Dette vil bli sett som extras hvis det er tid til det i slutten.

6 DOKUMENTASJON

6.1 Rapporter og tekniske dokumenter

Dokumentasjon som vil bli lagd i løpet av prosjektet vil være bachelor rapporten, sprint reviews og retrospectives.

Sprint review'ene og retrospectives vil skje etter hver sprint, som i dette tilfelle er annenhver uke.

Bachelor rapporten vil bli arbeidet på konsistent gjennom prosjektet for å ikke glemme viktige detaljer. Sprint review og retrospectives vil bli lagret på NTNU Wiki.

7 PLANLAGTE MØTER OG RAPPORTER

7.1 Møter

7.1.1 Møter med styringsgruppen

Møter med veileder annenhver uke mandager 09:00.

7.2 Periodiske rapporter

7.2.1 Framdriftsrapporter (inkl. milepæl)

Sprint reviews og retrospective hver sprint (annenhver uke)

8 PLANLAGT AVVIKSBEHANDLING

Ettersom dette prosjektet ikke har konsekvens for noen andre enn meg selv, og har ingen finansielt betydning, dette prosjektet vil bare bli tatt til side om det feiler.

9 REFERANSER

Scrum - <https://www.atlassian.com/agile/scrum>

Unity - <https://docs.unity3d.com/Manual/UnityManual.html>

Leshylabs - <https://www.leshylabs.com/apps/sfMaker/>

VEDLEGG

2020-02-17 Tilbakeblikk

Dato	17 Feb 2020
Deltakere	Asbjørn Frostad

Tilbakeblikk

Hva fikk vi bra til?

- Jeg jobbet med prosjektet omtrent hver dag.
- Jeg kom fort inn i arbeidsmodus, og klarte å konsentrere meg, selv om jeg jobbet hjemme.

Hva burde vi ha gjort bedre?

- Jeg burde ha vært litt mer optimistisk med arbeidsmengde i sprinten
- Jeg burde ha litt mer fast rutine for når jeg arbeider. (startet til forskjellige tider ofte).
- Jeg burde ha vært mer forberedt til møte.

Handlinger



2020-03-02 Retrospective

Date	02 Mar 2020
Participants	Asbjørn Frostad

Retrospective

What did we do well?

- Jobbet jevnt hver dag.
- Sov godt.
- Høy energi.

What should we have done better?

- Kunne ha gjort mer.
- Våknet ikke til samme tid hver dag.

Actions



2020-03-16 Retrospective

Date	16 Mar 2020
Participants	Asbjørn Frostad

Retrospective

What did we do well?

- Høy energi
- Fikk gjort mye på liten tid

What should we have done better?

- Jobbet ikke mye med prosjektet siste uka pga eksamen i annet fag kom brått på, i tillegg til andre småting pga corona

Actions



2020-03-30 Retrospective

Date	28 Mar 2020
Participants	Asbjørn Frostad

Retrospective

What did we do well?

- Jobbet mye pga lite annet å gjøre.

What should we have done better?

- Uregelmessig arbeidstider.
- En del arbeid ikke spesifisert i jira.

Actions



2020-04-13 Retrospective

Date	13 Apr 2020
Participants	Asbjørn Frostad

Retrospective

What did we do well?

What should we have done better?

- Inconsistent work times

Actions



2020-04-27 Retrospective

Date	27 Apr 2020
Participants	Asbjørn Frostad

Retrospective

What did we do well?

What should we have done better?

- Focus could have been better
- Mostly not working routinely

Actions



2020-02-17 Meeting notes

Date

17 Feb 2020

Attendees

- [Asbjørn Frostad](#)
- [Arne Styve](#)

Goals

- *Status-oppdatering*

Discussion items

Time	Item	Who	Notes
30 min	Bruk av Jira	Arne Styve Asbjørn Frostad	<ul style="list-style-type: none">• Gikk over hva en story er.• Etc dokumentasjon og når de skal gjøres.
10 min	Hvordan presentere produkt	Asbjørn Frostad Arne Styve	<ul style="list-style-type: none">• Snakket om at behov for å vise mobil skjerm på storskjerm.
10 min	Presentere hva som har blitt gjort så langt	Asbjørn Frostad Arne Styve	<ul style="list-style-type: none">• Viste spillet på mobilen.

Action items



2020-03-02 Meeting notes

Date

02 Mar 2020

Attendees

- [Asbjørn Frostad](#)
- [Arne Styve](#)

Goals

Discussion items

Time	Item	Who	Notes
20 min	Status	Asbjørn Frostad	<ul style="list-style-type: none">• Vise resultat så langt.
20 min	Diskutere prioritet i prosjektet.	Asbjørn Frostad	<ul style="list-style-type: none">• Diskutere om det skal bli mer fokus på utseende/polish eller funksjoner og mekanikker etc.
20 min	Etc	Asbjørn Frostad Arne Styve	

Snakket om:

- En story skal være en full fungerende funksjon.
- I stedet for å lage placeholder grafikk etc, er det bedre å bare lage ferdig hele storyen, så den skal være klar til å møte sluttbruker. Kan endre senere likevel.
- Skriv ned alle ideene uansett om det er usikkert om å følge de gjennom. Viktig å få dokumentert avgjørelser og ting som er vurdert.
- Husk mobil-ledning for demonstrering på pc ...
- Kan bruke den ekstra tiden til å legge til ads.
- Kan vurdere å vise en high score liste, for å gi spilleren en bedre sense of progress.
- Vurdere å lage flere nivå med forskjellige særtrekk (grafikk vis og funksjon vis) og vanskelighet.

Action items



2020-03-16 Meeting notes

Date

16 Mar 2020

Attendees

- [Asbjørn Frostad](#)
- [Arne Styve](#)

Agenda

- Status

Snakket om:

- Burde begynne å samle sammen tankene og fokusere litt mer på konseptet og det endelige prouktet.
- Skal man kunne ta igjen romskip eller burde romskip bare forsvinne/eksplodere elns.
- Burde begynne å noter ned litt mer om valg og hvorfor etc. kanskje begynne litt på prosjektrapporten.
- Forskjellig våpen?
- Kaste ut realismen og lage et par player models man kan velge i?

Action items



2020-03-30 Meeting notes

Date

28 Mar 2020

Attendees

- [Asbjørn Frostad](#)
- [Arne Styve](#)

Agenda

- Status

Snakket om:

- Lurt å bruke release version system på jira. Binde tasks etc til version.

2020-04-13 Meeting notes

Date

13 Apr 2020

Attendees

- [Asbjørn Frostad](#)
- [Arne Styve](#)

Agenda

- Status

Snakket om:

- Begynne på rapport.
- begynne å bli ferdig med app.

2020-04-27 Meeting notes

Date

27 Apr 2020

Attendees

- [Asbjørn Frostad](#)
- [Arne Styve](#)

Agenda

- Status

2020-02-03 Meeting notes

Date

03 Feb 2020

Attendees

- [Asbjørn Frostad](#)
- [Arne Styve](#)

Goals

- Learn to use Jira and confluence.

Discussion items

Time	Item	Who	Notes
60min	Jira, Confluence	Asbjørn, Arne	<ul style="list-style-type: none">• Going over how to use Jira and confluence.

Action items

