

Eksamensprojekt DDU - “Spiludviklingsprojekt”

Elev(er):	Alexander William Bruhn, Julie Kristina Jensen, Marcus Alsted Wegmann og Oliver Alexander Negendahl Jessen
Vejleder(e):	Anders Juul Refslund Petersen (ajrp), Kristian Krabbe Møller (kkm)
Uddannelsesfag:	Digital Design og Udvikling (DDU)
Læreanstalt:	H.C. Ørsted Gymnasiet, Kgs. Lyngby
Afleveringsfrist:	15. maj 2022

Gruppemedlem 1.

Alexander William Bruhn

Gruppemedlem 2.

Julie Kristina Jensen

Gruppemedlem 3.

Marcus Alsted Wegmann

Gruppemedlem 4.

Oliver Alexander Negendahl Jessen

Underskrift 1: _____ Underskrift 2: _____

Underskrift 3: _____ Underskrift 4: _____

Indholdsfortegnelse

Kort abstrakt (15-20 linjer)	4
Indledning	5
Projektbeskrivelse	6
Projektets rammer	6
Rollefordeling	6
Krav til indhold i eksamsprojektet	7
Projektstyring	8
Trello & GitHub	8
Burndown Chart	9
Projektstyringsmodel	9
Problemidentifikation	11
Valg af case	11
Krav ift. valg af case	11
Problemanalyse	12
Problemstilling	12
Produktpincip	13
Idégenerering	13
Kravmatrix ift. brainstorm	14
Valg af program	14
Kravspecifikation	15
Kravspecifikation ift. spilleren	15
Produktdesign	16
Designspecifikation	16
Moodboard	16
Målgruppeanalyse	18
Gruppens spillertype	18
Personae	18
Baggrundshistorie	22
Produktudformning	23
User Stories	23
Tasks	24
Planlagte iterationer med dertilhørende datoer	27
Iteration 1: Grundlæggende gameplay	27
Iteration 2: Design	27
Iteration 3: Udvidet gameplay	27
Iteration 4: Forbedring	27
Klassediagram	28
Tekniske valg	28

Adobe Photoshop	29
Adobe Illustrator	29
Hubspot	29
Java og Processing	29
Visual Studio Code	29
Brugertest	30
Planlægning af test	30
Produktudvikling	31
Skitser	31
Farveteori + gestaltlove	33
Frøen	33
Platform	35
By	35
Fjende 1 - Fuglen	36
Fjende 2 - Flyet	37
Fjende 3 - UFO	37
Samleobjekt	37
Animation	38
Karakter	38
Flue	39
Fjende	39
Parallax	39
Endeligt design	40
Fjender	40
Platforms	40
Forgrund & mellemgrund	41
Baggrund	42
Forventning af hvordan spillet kommer til at se ud:	43
Gameplay	44
De 3 C'er	44
SWOT-analyse	45
Strengths:	45
Weaknesses:	45
Opportunities:	45
Threats:	45
Forklaring af programmet	46
Overordnet flowchart	46
Main-klassen	47
Loading af assets	47
World-klassen og chunk-system	47
Platform-, Coin- og Enemy-klassen	47
Typer af platforme og fjender	48
Spiller-inputs og ButtonManager-klassen	48

Bevægelse og fysik	48
Spilleren og Frog-klassen	48
Kollision	49
Tyngdekraft, vind og vindmodstand/friktion	50
Camera-klassen	50
Baggrunden	51
IT-sikkerhedspolitik	51
GDPR (Databeskyttelsesforordningen)	51
Ophavsret	51
Brugertest	52
Feedback	52
Konklusion på feedback	54
Forbedring af program ift. brugertest	54
Evaluering	55
Opgavekrav	55
Burndown chart	56
Konklusion	57
Logbog	58
Litteraturliste	61

Kort abstrakt (15-20 linjer)

Til dette eksamensprojekt fik gruppen som opgave at udvælge en case, og derefter designe og programmere et program der opfyldte alle kravene for eksamensprojektet. Den følgende rapport indebærer en gennemgang af hele projektets udvikling, som til sidst ender ud i en prototype af et computerspil.

Til eksamensprojektet anvendte gruppen en masse forskellige metoder og kilder til at udforme rapporten og programmet, f.eks. rollefordeling, projektstyringsmodel, tasks, iterationer, klassediagram, brugertest, osv. Alle metoder er blevet brugt til at kunne arbejde videre med næste del af projektet, og var med til at finde på nye måder at lave rapporten og programmet. Det som man kan forvente at læse i projektrapporten, er hvordan gruppen har kommet frem til det endelige produkt. Her bliver der skrevet undervejs om projektstyring, projektudvikling, produktudformning og selve programmet for projektet.

I opgaven skulle man designe sit eget computerspil hvor brugeren ville kæmpe mod computeren og tænke strategisk og være aktiv hele tiden. Man fandt ud af hvilken målgruppe passede bedst til spillet ved at undersøge spillertyper og personaer. Man planlagde alle iterationer for programmet og arbejdede på det og rapporten således. Man fik evalueret gruppens arbejde og diskuteret hvad der kunne have gået bedre i gruppen, samt hvad kunne forbedres på rapporten og programmet.

Rapporten ender ud med konklusionen at gruppen fik udformet en prototype til et computerspil der opfyldte alle kravene til projektet, og fik skrevet en rapport med alt nødvendig information til programmet.

Indledning

Dette eksamensprojekt bearbejder en blanding af case 1 og 2 som er et selvvalgt projekt, der omhandler at skulle programmere et program hvor man spiller mod en computer. Gruppen har af den grund udarbejdet en projektbeskrivelse fra bunden til at kunne bedst som muligt løse casen.

Gruppen vil derefter lave en projektstyring til at kunne holde styr på gruppens arbejde og dermed kun lave relevant arbejde der hjælper med at løse casen. Derefter vil gruppen formulere en problemstilling der skal først besvares for at kunne løse casen. Derfra vil gruppen arbejde med at få besvaret problemstillingen ved at lave produktdesign, produktudformning, og evaluere programmet med en brugertest for at til sidst kunne vurdere og konkludere projektet for at løse casen og afslutte eksamensprojektet.

I denne rapport lægges der mest fokus/vægt på at fremstille et program, der kan direkte løse problemstillingen og casen. Derudover skal programmet også være beskyttet og sikret mod datatyveri, således at brugere ikke kan gå ind i programmet og f.eks. ændre på vigtige data.

Link til projektgruppens program:

Det skal nævnes at det for at programmet skal kunne køre er det nødvendigt at have Java JDK 17. Yderligere er gruppens eftersyn at programmet egenes bedst til Windows.

<https://github.com/marc9622/Eksamensprojekt>

Projektbeskrivelse

Projektets rammer

Gruppen består af fire medlemmer. Eksamensperioden starter mandag d. 21. februar 2022, og slutter d. 15 maj 2022. Opgaven indeholder ca. 80 lektioner på skolen og ca. 33 t elevtid. Dvs. det samlede omfang af projektet er ca. 110 timer pr. gruppemedlem. Der skal afleveres ét produkt (et program / program system) og én rapport pr. gruppe på Lectio. Gruppen skal vælge en ud af 7 cases, eller finde på deres egen case der involverer alle kravene til eksamensprojektet og er godkendt af begge vejledere.

Rapportens omfang er 15-30 normalsider for den første i en gruppe, og 5-15 normalsider pr. ekstra gruppemedlem. Man kan anvende rapport-skabelonen fra bogen Digitalt design og udvikling på Systime.¹

Rollefordeling

Gruppen har indbyrdes foretaget en rollefordeling for at kunne optimere gruppens effektivitetsniveau. Gruppens roller er fordelt i tre områder, og består af en programmør, designer/kunstner og skribenter. Dog er der til tider overlap mellem gruppens roller hvor en skribent eller to hjælper med at arbejde på produktets kode eller assets. Det omvendte kan til tider også være tilfældet, det giver f.eks. mening at den programansvarlige står for at lave og skrive om hvilke metoder der bruges til opbygning af programmet. På baggrund af gruppemedlemmernes baggrund ser rollefordelingen således ud.

Programmør:

Marcus er det gruppemedlem der er ansvarlig for programmet og processen med det. Denne beslutning er taget på baggrund af hans erfaring med programmering som han har beskæftiget sig med siden før han begyndte på gymnasiet. Oliver er på som en form for assistance til programmet og skal hjælpe primært i den tidlige og endelige fase af programmet med at få de konkrete tasks i programmet til at fungere og finpudse programmets implementerede design.

Designer/kunstner:

Julie er gruppens kunstneriske leder, og er den der har defineret produktets udseende. Beslutningen om hvem der skulle stå for den kunstneriske retning af produktet er taget på baggrund af hvem der er bedst til rollen samt kan holde det bedste overblik for processen. Alexander har også været inde over denne process og hjulpet med at supplementere arbejdsprocessen. Fordelingen mellem Julie og Alexanders bidrag til produktets assets er estimeret til at skulle være 80% - 20%, hvor Julie udgør de 80% og Alexander de resterende 20%.

Skribenter:

Oliver og Alexander er skribenterne der står for størstedelen af rapporten. De står for både indholdet der skal være med i rapporten, og organiseringen af hvad der skal stå

¹ <https://ddu.systime.dk/index.php?id=239&L=0>

hvorhenne. Ved at have gruppemedlemmer der styrer rapporten, kan man holde bedre styr på hvorledes arbejdsprocessen, burndown charts, tasks, iterationer, osv er gået. Langsomt over tid, når både designet for programmet og selve programmet er fuldgjort, vil de andre to gruppemedlemmer, Julie og Marcus, deltage i at skrive på rapporten, og primært skrive om de områder de har stået for.

Ved at organisere og fordele arbejdet mellem gruppemedlemmerne, kan man sørge for at overlade et vis stykke arbejde til det gruppemedlem der har noget god erfaring til at arbejde med det. Ved ikke at lave rollefordeling har man sværere med at finde gruppens styrke og svagheder, og dermed en langsommere arbejdsproces.

Krav til indhold i eksamensprojektet

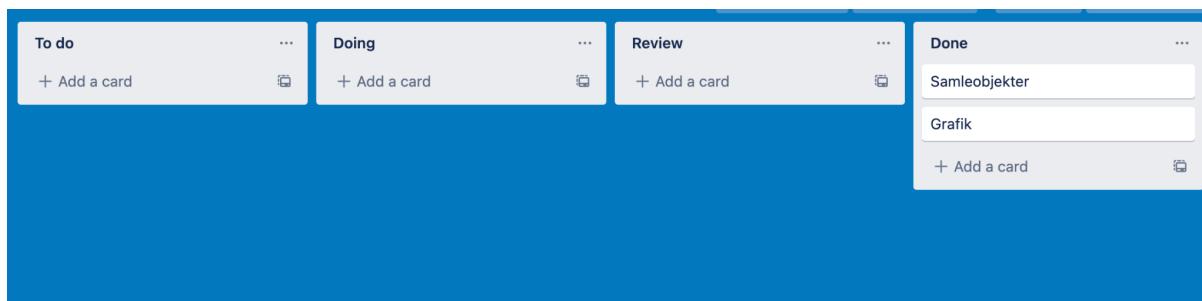
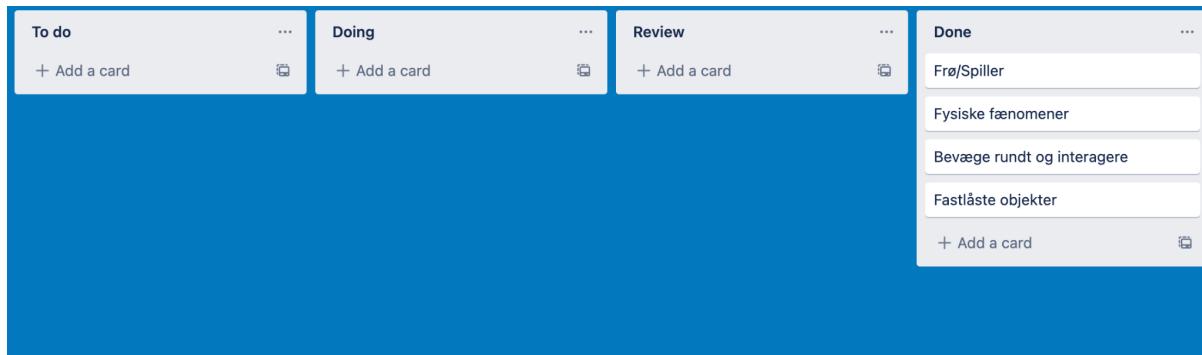
- Der skal **anvendes en database** eller anden form for datasikring.
- Selve produktet skal overvejende være **udviklet med et højniveau sprog**, vi anbefaler Processing, C#, Java, Python eller C++. Vi forbeholder os retten til at **godkende udviklingsmiljø og sprog**, hvilket derfor skal være tydeligt angivet i jeres projektbeskrivelse.
- Vi forventer et højt niveau inden for både valg og udførelse af **projektstyring** igennem hele projektet, herunder en særlig kritisk indstilling til tidshorisonter (for optimistiske/for pessimistiske) og projektomfang.
- Der skal **udvikles en prototype**. Prototypen skal testes og danne grundlag for den endelige udvikling af produktet.

Desuden skal det overvejes hvordan I håndterer: sikkerhed, GDPR, interaktionsdesign og brugertests.

Projektstyring

Trello & GitHub

Gruppen bruger projektstyringsværktøjet Trello til at planlægge projektets forløb. Trello er et meget smart tidsadministrationsværktøj at bruge i projekter, da det lader gruppen holde overblik over hvilke deadlines man skal nå og hvilke individuelle ting der skal laves.

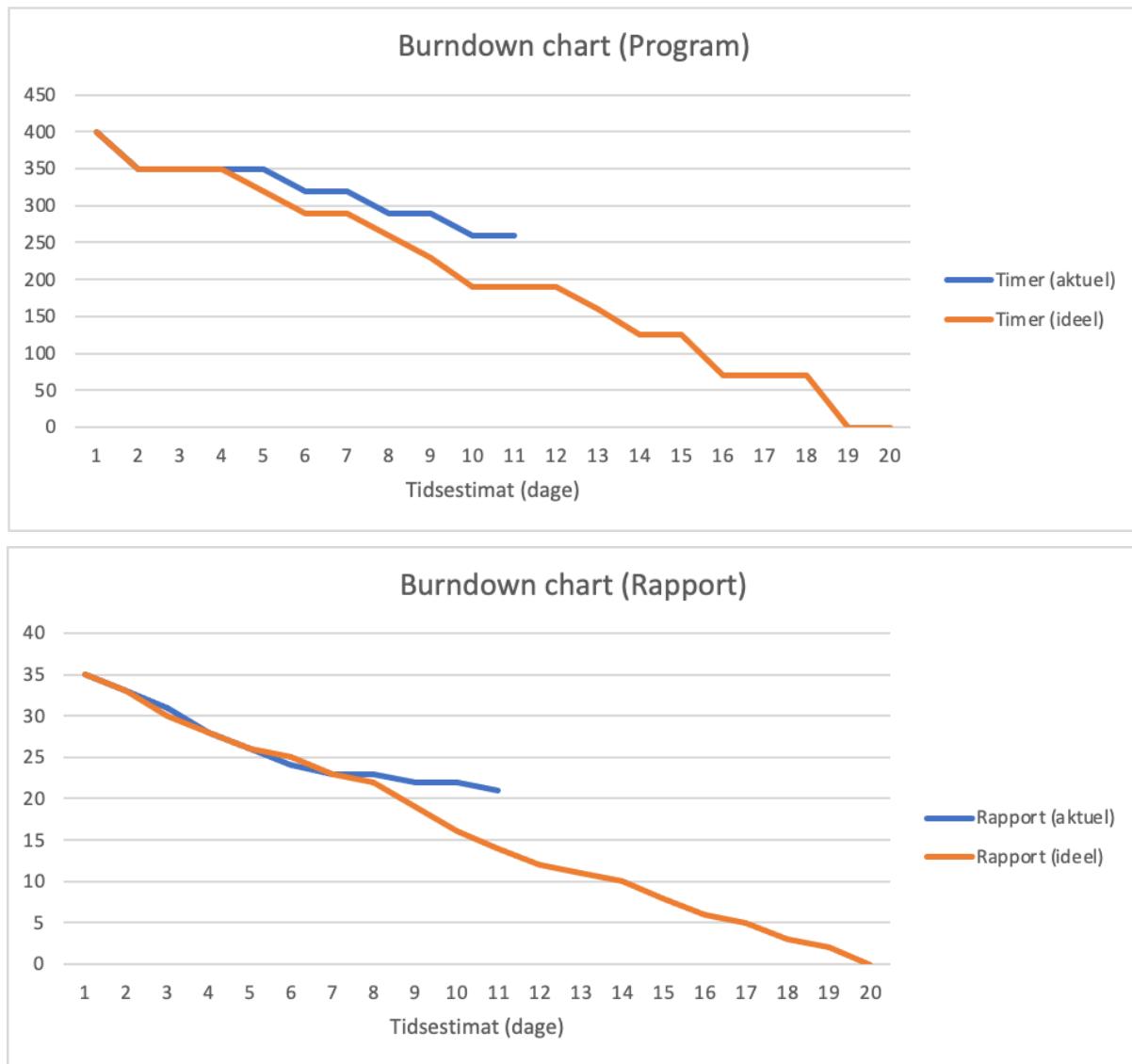


(Figur 1 og 2: Skærbilleder af Trello for de første to iterationer.)

Der bliver også lavet et repository i GitHub. Ideen med at have et repository er først og fremmest for at kunne opbevare koden et sikkert sted. På den måde kan alle gruppemedlemmer kontribuere til programmet ved at give løbende opdateringer til spillet og en beskrivelse af opdateringen, men også reflektere på arbejdsprocessen og dermed forbedre programmet.

Burndown Chart

Gruppen har besluttet at lave to forskellige burndown charts - en for rapporten, det andet for programmet. der kommer også til at være lavet det samme burndown chart - første gang midt i projektet, og en gang til sidst i projektet (se figur 3 og 4).



(Figur 3 og 4: Burndown chart for både programmet og rapporten. Disse charts er dem der er blevet foretaget midt i projektet, og er grunden til at grafen pludselig stopper.)

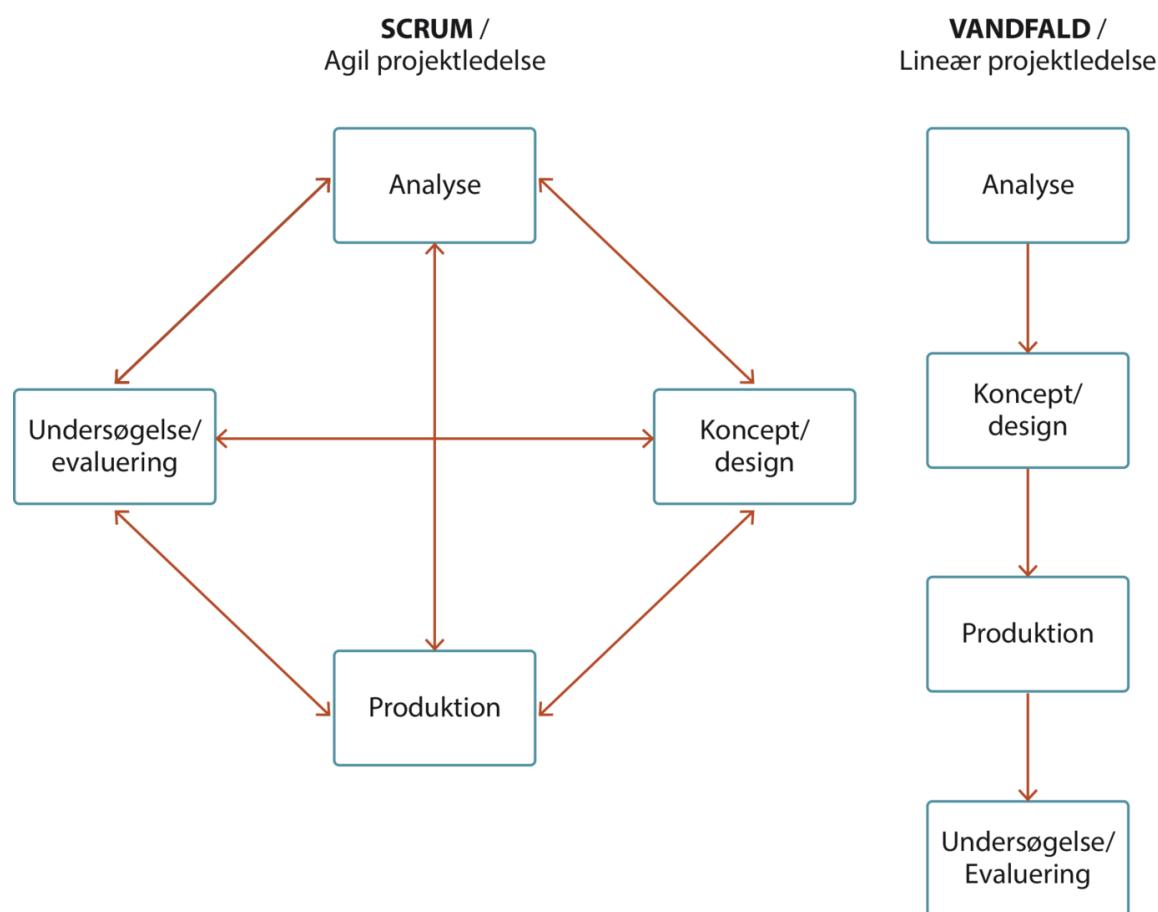
Projektstyringsmodel

Til at bedre kunne holde styr på arbejdsprocessen, er det vigtigt at have organiseret og diskuteret hvilken slags projektledelse som man vil arbejde med. Gruppen har begrænset deres projektledelse ned til to forskellige slags; den lineære og den agile projektledelse (se figur 5). Den lineære projektledelse, også kendt som Vandfald-metoden, er ofte brugt til forsøg og projekter hvor man ender op med et konkret produkt. Vandfald-metoden er meget fastlåst og kan derfor bruges i projekter der varer i længere perioder (et par måneder).²

² <https://ddu.systime.dk/?id=271>

Scrum-metoden er en af de mest anvendte metoder inden for agil projektledelse. Den bygger på principperne fra den agile tankegang hvor der arbejdes ud fra en iterativ tilgang uden langsigtede planer og estimerer, dvs. man laver mange gentagelser og sprints for hver del af projektet. Tværtimod er agil udvikling mere fleksibel og åben for ændringer undervejs end lineær. Projekter der anvender Scrum er ofte korte projekter der varer 2-4 uger.

Gruppen har valgt at gå videre i projektet ved at bruge en kombination af Scrum- og Vandfald-metoden, dvs. man har både et planlagt estimat og går primært projektet an ifølge den lineære projektledelse, men samtidigt man vender tilbage til andre punkter af projektet for at gentage, opdatere eller omskrive nogle dele af teksten ligesom den agile projektledelse. Man kan også vende tilbage for at skrive noget helt nyt, og dermed lave en helt ny lineær projektledelse.



(Figur 5: Illustration af Scrum- og Vandfald-metoden. Hver metode har noget specielt at tilbyde, så gruppen har besluttet at anvende begge metoder.)

Problemidentifikation

Valg af case

Gruppen vil gerne til dette eksamsprojekt prøve på at udvikle et program som er et computerspil og kan spilles af mange, og som derfor er dens inspiration en blanding af case 1 og 2. Programmet skal kunne være begyndervenligt samt stige i sværhedsgrad og skal kunne benytte databaser eller anden form for datasikring undervejs (både gemme og implementere data).

Case 1: Spiludvikling

Et klassisk spil hvor brugeren spiller mod computeren. Der skal være mulighed for at navigere, score point og spilleren taber, hvis der ikke er noget brugeraktivitet (tiden løber ud eller niveauet for liv bliver mindre og mindre). Der er mulighed for at udvide med gemmefunktion undervejs, brugerhåndtering af forskellige brugere, topscorerlister og egentlig data opsamling af brugernes adfærd i spillet.

Case 2: Realtids strategispil

Der skal være 2 sæt brikker, med forskellige karakterer fx.

- Bueskytter
- Riddere
- Fodfolk
- Kongen

Hver type skal have sine egne karakteristika mht. bevægelse og aktion. Computeren skal styre det ene sæt og brugeren det andet sæt. Brikkerne skal fungere som 'levende', dvs. de handler selvom brugeren ikke påvirker dem. Formålet er underholdning.

For at gruppen kan bedst beslutte, har man lavet nogle krav ud fra casene til bedst at vægte hvilken case opfylder de fleste af gruppens krav.

Krav ift. valg af case

1. Spilleren skal altid være aktiv.
2. Spilleren skal overveje sine valg.
3. Computeren/programmet skal kunne udfordre spilleren.
4. Inaktivitet skal straffe spilleren.

Efter tung overvejelse har gruppen valgt at programmets inspiration vil mest komme fra case 1, da det hovedsageligt omhandler at brugeren skal konstant være i bevægelse og score point. Inspirationen fra case 2 omhandler at spilleren vil styre ét sæt af karakterer, og computeren vil styre et andet sæt.

Problemanalyse

Problemstilling

Efter at have vurderet at programmet skal være en blanding af case 1 og 2, har gruppen fremstillet en problemstilling der burde både dække over kravene for casene, og som er realistisk at producere i et eksamensprojekt:

Hvordan kan man lave et computerspil der kan sørge for at spilleren er både aktiv, tænker strategisk, og får spilleren til at planlægge sit næste træk, men samtidigt også er begyndervenligt og tilgængeligt for alle i aldersgruppen fra 6 årige til og med 86 årige?

Gruppen vil senere hen også gerne prøve på at markedsføre programmet ved at gøre spillet fængende, så det kan konkurrere med andre af den samme slags spil på spilmarkedet. Derudover vil der også senere hen blive valgt en intenderet målgruppe, som spillets tilgængelighed skal fokusere på.

Produktprincip

Idégenerering

Til at hjælpe på at få inspiration til et program der kan gøre de fleste af disse ting, har gruppen lavet en brainstorm af diverse computerspil som møder kravene til gruppens program, som så bruges til at finde på varianter eller program-idéer derfra. Alle spillene i brainstormen skal også være 2D, siden at gruppen ikke har nok erfaring med 3D programmer. Kravene som spillene bliver bedømt på er ens med dem man brugte til casene.

1. Oregon Trail (Text adventure + choice-impact game)
 - Handler om man skal guide en gruppe af udvalgte bosættere fra Independence, Missouri, til Willamette Valley i Oregon, mens man rejser langs den famøse Oregon Trail i 1848. Målet med spillet er at kunne tænke strategisk og planlægge sit næste træk.
2. Earthbound (Roleplaying simulator)
 - Handler om at man skal styre en gruppe karakterer, der rejser gennem landsbyer, byer, huler og fangehuller for at kæmpe mod fjender, og man får erfarringspoint hvis man vinder. Målet med spillet er at være aktiv og planlægge strategisk.
3. DownWell (Vertically scrolling shooting + platform game)
 - Handler om at man skal styre en mand, som udforsker dybden af en brønd. Som spilleren bevæger man sig nedad og dræber fjender for at fortsætte længere ned og samle på skatte. Målet med spillet er at være aktiv og tænke strategisk, samt planlægge det næste træk.
4. Tower Defense Simulator (Progressive idle upgrade game)
 - Handler om at man skal forsvarer et territorium ved at blokere fjendens angribere eller ved at forhindre fjender i at nå udgangene, normalt opnået ved at placere defensive strukturer på eller langs deres vej der automatisk blokerer, hindrer, angriber eller ødelægger fjender. Målet med spillet er at tænke strategisk og planlægge det næste træk.
5. Plants vs Zombies (Progressive idle upgrade game)
 - Handler om at man skal forsvarer et hus ved at placere forskellige defensive strukturer der angriber imødekommande fjender og forhindre dem fra at nå enden af banen. Målet med spillet er at tænke strategisk og være aktiv.
6. Tumbleseed (roguelike balance + shooting game)
 - Handler om at man skal balancere et frø på et bræt op ad et farligt og skiftende bjerg. Undgå huller og besejor korrupte væsner for at komme op på toppen. Målet med spillet er at være aktiv, tænke strategisk, og planlægge sit næste træk.
7. Shovel Knight (platform game)

- Handler om at man skal bevæge igennem en masse baner og besejre fjender undervejs for at komme videre til den næste bane, imens man samler skatte op. Målet med spillet er at være aktiv og tænke strategisk.

Kravmatrix ift. brainstorm

Spillene vægtes ud fra en skala af 1-5 ud fra hvor højt de opfylder kravene. Til sidst vil vægtene blive lagt sammen for en samlet score, og den idé med den største score vil blive arbejdet videre med (se figur 6).

- Spilleren skal altid være aktiv.
- Spilleren skal overveje sine valg.
- Computeren/programmet skal kunne udfordre spilleren.
- Inaktivitet skal straffe spilleren.

Spil idé	Krav nr. 1	Krav nr. 2	Krav nr. 3	Krav nr. 4	Samlet score
Oregon Trail	2	5	3	1	11
Earthbound	1	4	4	1	10
Downwell	5	4	5	5	19
TDS	4	5	4	4	17
Plants vs Zombies	3	4	4	4	15
Tumbleseed	5	4	4	5	18
Shovel Knight	3	3	4	4	14

(Figur 6: Kravmatrix for valg af spilidé.)

Valg af program

Efter at have vurderet de diverse spil ud fra krav matrixen, har gruppen valgt at arbejde videre med en variation af spilidé nr. 3, som var platform spillet "Downwell".

Kravspecifikation

Kravspecifikation ift. spilleren

Der skal være nogle krav til hvilke elementer der skal være i spillet og hvad elementerne i spillet skal kunne. Gruppen vil så godt som muligt prøve at forenkle hvilke knapper skal gøre hvad i programmet, men det kan variere eller skiftes senere hen i udviklingen af programmet pga. brugertest eller gruppebeslutning.

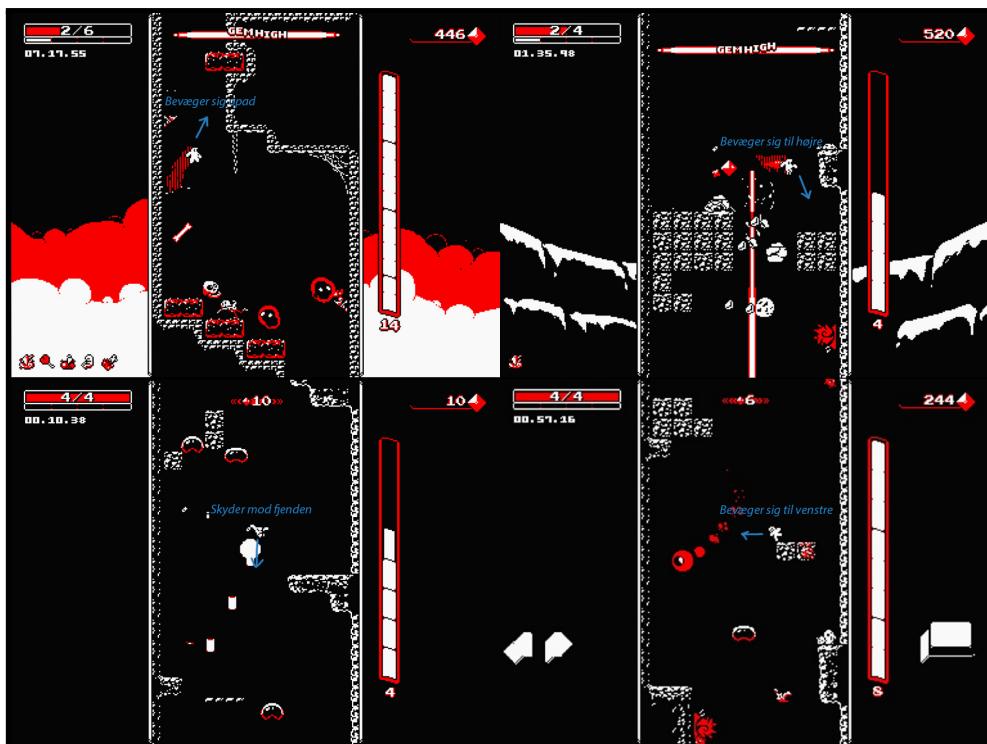
1. Når venstre museknap trykkes, bliver der skudt et objekt fra spillerens position. Skuddene skal interagere med fjender og ikke-fastlåste objekter i forgrunden.
2. Når knapperne "a" eller "d" bliver trykket, bevæger spilleren sig til højre eller venstre.
3. Når mellemrumstasten trykkes, hopper spilleren.
4. En tæller skal hele tiden vise antal samleobjekter spilleren har.

Disse er de primære og essentielle elementer/krav i programmet. De henvender til spillerens bevægelse og handlinger, og behøver ikke at kunne mere end beskrevet ovenover. Der findes også kravspecifikationer til fjender og diverse elementer, men er adskilt fra kravspecifikationerne til spilleren.

Produktdesign

Designspecifikation

Eftersom at det højest vurderede spil ud fra krav matrixen var platform spillet "Downwell", har gruppen derfor valgt at kode spillet som et platformer. Designet kommer til at være anderledes, men det vil have det samme princip. Gruppen har derfor valgt at tage udvalgte skærmbilleder og vise dens elementer ved hjælp af små pile og tekstbokse (se figur 7).



(Figur 7: Skærmbilleder af spillet "Downwell", hvor spilleren udfører forskellige tasks.)

På skærmen vil man skulle kunne se antal liv, skud og penge som spilleren har på ethvert tidspunkt. Karakteren skal bevæge sig til venstre, højre, både op og ned. Den skal også have et våben eller noget lignende der kan interagere med objekter og/eller fjender.

Måden man skal spille spillet på indtil videre, vil være at bevæge sig opad og ramme/undvige fjender, samt samle på objekter og interagere med fastlåste objekter. Grafikken for spillet vil være designet så det ikke er ligesom spilidéen, og kan derfor variere i format, indhold og elementer.

Moodboard

Et moodboard er blevet udarbejdet i gruppen for at gøre design processen mere effektiv og visuelt forklarende.³ Formålet med moodboard er at slå det konkrete ønske for spillets

³ <https://www.toptal.com/designers/visual-identity/guide-to-mood-boards>

vision/stil på plads. Moodboardet hjælper gruppen med at udvælge visse farver og inspiration til kunst stilen.



(Figur 8: Moodboard for programmets design.)

Moodboardet er opbygget af en række billeder og en farvepalette. Bunden af moodboardet definerer de farver som projektet skal anvende i sin stil, mens billederne definerer hvilken stil og laktation spillet skal have. Det er blevet besluttet at spille skal have en mere simple strømliniet still hvor der bruges farver frem for texture.

TV-serien Gurli Gris og spillet Pikuniku er hvad der primært har inspireret gruppens visuelle design. Yderligere er der også blevet brugt et fotorealistisk billede af Central Park, da det tegner et godt billede af hvilket miljø spillet skal foregå i. Yderligere er platform spillet "Happy Pudding Jump" også repræsenteret på gruppens moodboard, da spillet har lidt af den samme type gameplay samt stil som gruppens program går efter.

Målgruppeanalyse

Til hvert spil er der en målgruppe som spillet appellerer til. Målgruppen vil ende med at definere hvordan spillet skal blive spillet. Til at hjælpe med at finde ud af den intenderede målgruppe (den målgruppe gruppen gerne vil have skal spille spillet), finder man nogle forskellige metoder og roller som kan bruges til at beskrive de forskellige slags målgrupper, og hvad de går efter når de spillet. Til dette platform spil har spillet en særligt bred målgruppe da spillet er begyndervenligt og appellerer til de fleste aldersgrupper.

Målgruppen til gruppens spilprogram kan defineres ved at anvende enten Marczewskis eller Bartles spillertyper til at bestemme hvilken spillertype målgruppen er.⁴ Her bruges Marczewskis spillertype ofte, da der er seks forskellige spillertyper som man ofte ser i diverse mainstream spil, og nogle gange har man fundet en spillertype som man ikke forventer ville være til stede. Dette fænomen hedder den intenderede og aktuelle spillertype.

Da det kom til at vælge imellem at analysere spillet for Marczewskis eller Bartles spillertype, var det et relativt nemt valg at vælge Marczewskis spillertyper. Grunden til det er fordi Bartles spillertyper appellere mere til MMORPG's (Massively Multiplayer Online Role Playing Game). Dette er en spiltype hvor der bliver dyrket oftest rollespil og kæmpe online spil, og selve opfinderen Bartle nævnede i et interview at den kun kan virkelig bruges til den genre af spil. Eftersom at gruppens spil er ikke et MMORPG på nogen måde, siden man hverken "leger" rollespil på nogen måde eller spiller online med andre. Det er således derfor at gruppen valgte at bruge Marczewskis spillertyper.

Gruppens spillertype

Angående spillets spillertype, så har gruppen ikke nødvendigvis en intenderet spillet type, da det er så bredt at det ligefrem kan spilles af mennesker i alle aldersgrupper. Dog har gruppen fundet en af de seks spillertyper som ikke var intenderet at appellere til, men som i nærmere eftertragtning er perfekt. Denne spillertype kaldes for "The Achiever", og er kategoriseret efter at spilleren er særligt motiveret til at mestre spillet og blive bedre til det.

"The Achiever" er en spillertype som leder efter spil som de kan mestre, eller lære nye ting fra, bare så længe de kan få en udfordring ud af det. Her kan gruppe se hvordan "The achiever" appellere til spillet. Disse type mennesker ser dem ikke selv nødvendigvis som folk der vil bruge masser af tid, men bruger undersøgelserne viser at folk ikke kan lade være med at spille videre. Målgruppen er mennesker der vil prøve at mestre spillet så at de kan nå så langt som de nu kan og hele tiden prøve at blive bedre og slå deres tidligere high-score.

Personae

Eftersom målgruppen er meget bred, har gruppen lavet forskellige personaer som skal give et indblik i hvilke slags personer som ville tage glæde i at spille spillet, om det så er i en kort stund eller over en længere periode af tid. Her har gruppen skullet tage hensyn til at vise en person fra de mange aldersgrupper som spillet appellerer til.

⁴ <https://www.interaction-design.org/literature/article/bartle-s-player-types-for-gamification>

Her har gruppen lavet et scenarie for hver person som ses forneden. De forskellige personer er alle mennesker som er interesseret i at spille spil såsom dette, og har deres egne grunde til hvorfor. Gruppen har lavet én person som er med fra den yngre aldersgruppe og arbejde videre derfra til de ældre aldersgrupper. Hver persona har hver især deres egen kompetencer og problemer.

Når man laver personaer til at bedømme et produkt, er et tal mellem 3 til 4 personaer en godt antal, siden at man skal have forskellige mennesker fra alle steder med, men ikke for mange personer da man hurtigt kan miste overgreb. Hvis man f.eks. kun havde 1 eller 2 personaer med til programmet, har man ikke kunnet se mængden af diversitet i svarene, som man kan have i alle målgrupper. Det er derfor utroligt vigtigt at have den rette mængde af mennesker med når man laver personaer til et produkt.

Henry



Age
18 to 24 years

Highest Level of Education
Some college, no degree

Social Networks



Industry
Flower shop

Organization Size
1-10 employees

Preferred Method of Communication
Henry bruger for det meste messenger som hans primære form for at kommunikere med sine medarbejder og sine venner/familie

Tools They Need to Do Their Job
han bruger sin telefon for at kigge på bestillinger, dog bruger han også sin telefon da det er en lille butik så der kan være ret stille, og derfor kan godt lige at spille spil, for at slå tiden ihjel.

Job Responsibilities
Henry har en lille blomster foretning som han har meget kær, her skal han åbne og lukke foretningen, samt betjene kunder.

They Gain Information By
han får meget sin information fra instagram og facebook

BIO.
Henry er en ung mand som elsker natur og blomster, han har startet en lille butik i københavn hvor han sælger blomster og buketter. Det en lille butik og der kan ofte være ret stille om eftermiddagen. Han bruger lang tid på at gøre sin butik i pæn stand og for den til at stå ud fra de andre, dog oplever han ofte at der ikke er nok at gøre, og derfor bruger mere til på telefonen. Her kan godt lige at spille små spil som kan slå tiden i hjel.

(Figur 9: Persona af Henry.)

Scenarie:

Ved at analysere Henrys persona, kan gruppemedlemmerne se at han spiller oftest spil i enten hans fritid eller når hans butik ikke har nogle kunder eller har travlt. Spillet som gruppen har lavet ville derfor være særligt brugbart i dette scenarie, siden spillet ikke handler om at spille en bane på en bestemt mængde af tid, men hellere at man kan spille så længe man har lyst til og genstarte hvis man fejler. Til at konkludere, er denne persona et af det perfekte scenarier for gruppens spil.

Stine



Job Title
Postnord udbringer

Age
25 to 34 years

Social Networks



Industry
Manufacturing

Organization Size
11-50 employees

Preferred Method of Communication
Stine bruger for det meste sin telefon for at kommunikere med arbejde og familie. samt bruger hun også Emil for at repottere tilbage til hendes job

Tools They Need to Do Their Job

- Email
- Employee Scheduling Software
- Reporting Software

Job Responsibilities
Hun skal udlevere pakker.

They Gain Information By
the local newspaper

BIO.
Stine bruger meget til på at køre rundt med post og pakker. Hun har ikke meget tid til at kunne tjekke sms'er og emils, samt underholde sig selv når hun køere og udleverer. Men når Stine har sin fokost pause kan hun godt lide at side på telefonen og spille spil for at have lidt sjov inden hun skal ud og køre igen, dog spiller hun ikke lang tid, da hun også vilde tjekke sociale medier.

(Figur 10: Persona af Stine.)

Scenarie:

Ved at analysere Stines persona, kan man se at hendes arbejde tager en masse af hendes hverdag og fritid. Den måde hvorpå gruppen kunne se Stine spille spillet på, kunne være i hendes arbejdspauser. Dette er dog ikke oplagt da hun også gerne vil bruge en mængde tid på sociale medier, og kan ikke være alt for distraheret til sit arbejde. En brugerundersøgelse med spillet viser at det er meget afhængende når man først begynder at spille det. Stine vil bruge spillet sjældnere end gruppen havde håbet på. Til at konkludere, er denne persona en god, men ikke det bedste scenarie for gruppens spil.

Lau



Age
5 years

Highest Level of Education
kindergarten

Social Networks



Reports to
Mom and dad

Bigest Challenges
Lau's største udfordringer er at mor og far har meget travlt siden deres arbejde er meget vigtig, og derfor elsker Lau at kigge på ipad og spille spil.

BIO.
Lau er en energisk dreng som elsker at lege uden for, og have sjov, dog er hans forældre meget travle og har ikke altid tid til at kunne lege. Derfor har Lau sin egen Ipad som han bruger, og spiller med. Han elsker spil med massere af farver og sjove karaktere.

(Figur 11: Persona af Lau.)

Scenarie:

Ved at analysere Lau's persona, kan man se at Lau har en masse fritid på hænderne, og bruger ofte elektroniske apps til at distrahere sig selv når hans forældre har det travlt. Lau er den type dreng hvor gruppens spil kan bringe lidt sjov og farve hver gang hans mor og far har travlt med arbejde eller andet vigtigt. Lau kan godt lide spil med masser af farver og sjove karakterer, som er en fordel for gruppen. Til at konkludere, er denne persona et af de perfekte scenarier for gruppens spil.

Stefan



Job Title
Head of Research

Age
35 to 44 years

Highest Level of Education
Doctorate (e.g. PhD, EdD)

Social Networks

Industry
Pharmaceutical

Organization Size
1001-5000 employees

Preferred Method of Communication
SMS eller Telefonsamtale

Tools They Need to Do Their Job
Computer og laboratorie

Job Responsibilities
Overser forskning for en større farmaceutisk virksomhed

They Gain Information By
Enter text here

Bio
Stefan er i alderen mellem 35 til 44 og bor med hans kone og to børn. Han er meget højtudannet og har stort fokus på sin karriere samt familie. Stefan bruger meget tid på at deligere arbejdet mellem hans medarbejder samtidigt at validere deres arbejde hvilket han til tider kan finde en smule tungt. Før i tiden brugte han meget tid sammen med sine børn, men efterhånden som de er blevet ældre er de så småt begyndt at passe sig selv. Stefan har af den grund fået mere tid om hænderne som han ikke helt ved hvad han skal stille op med.

(Figur 12: Persona af Stefan.)

Scenarie:

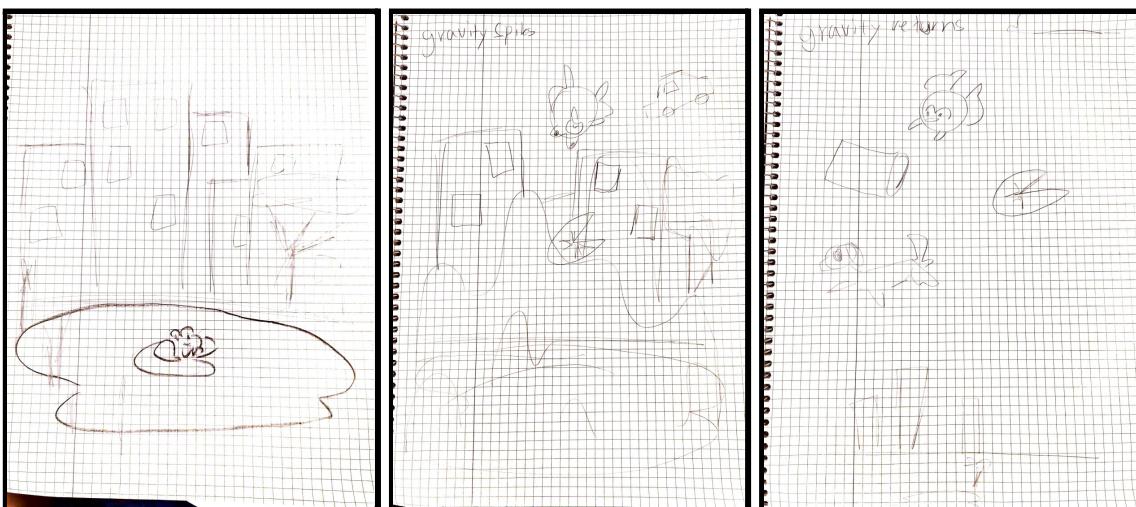
Ved at analysere Stefans persona, så virker han ikke særlig interesseret i computerspil. Man læser at Stefan har fået tid på hænderne og ofte bruger sin telefon til at distrahere sig selv, så virker det oplagt at en ældre forælder vil bruge tiden på at spille computerspil i stedet for at hjælpe til med huset eller andre pligter. Dog er det værd at nævne, at på grund af hans arbejde, kan han godt lide at spille spil som ikke behøver den største mængde koncentration og kunne finde på at spille gruppens spil. Til at konkludere, er denne persona ikke den bedste til gruppens spil.

Baggrundshistorie

Inden at man går videre med at lave klassediagrammer, user stories, iterationer, osv. så vil gruppen forklare noget baggrundshistorie for programmet/spillet.

Det er besluttet at man spiller spillet som en lille frø, der sidder og nyder den skinnende sol i storbyens park i en unavngiven by. Spillet tager plads i fremtiden, hvor global opvarmning har forårsaget noget som man har kaldt for "Gravity Spikes". "Gravity Spikes" fremkommer på forskellige tidspunkter, som gør at tyngdekraften ophæves helt i en mindre periode, hvor alt og alle som ikke er spændt fast til jorden bliver vægtløse i luften.

Frøen bliver så påvirket af disse spikes og bliver kastet op i luften sammen med alle mulige løse objekter omkring parken og i byen, f.eks. bænke, løjje puder, skraldespande og trafik kegler. Hovedmålet for frøen er nu at komme ned sikkert igen, ved at hoppe fra objekt til objekt. Spillet tager meget inspiration fra andre platform spil, men er original i sin baggrundshistorie (se figur 13).



(Figur 13: Skitser af de forskellige skærmbilleder som man vil kunne se i spillet. Skitserne skildrer en frø og forskellige genstande i et byområde, og man burde en hurtig fornemmelse for hvad historien for spillet er.)

Gruppen har valgt denne baggrundshistorie/scenario, da det virkede som en god historie der appellerede til målgruppens yngre brugere, ligesom det populære platform spil Doodle Jump.⁵

⁵ https://en.wikipedia.org/wiki/Doodle_Jump

Produktudformning

User Stories

Titel	Frø
Beskrivelse	Frøen er den spilbare karakter. Brugeren skal kunne styre frøen så den kan hoppe og interagere med objekter rundt omkring den. Dens udseende er en tegnet grøn frø, ligesom man ser i tegneserier og illustrationer.

Titel	Platform
Beskrivelse	Platformen er et enkelt objekt som skal interagere med brugerens spilkarakter. Der skal være flere objekter som konstant bevæger sig opad y-aksen og hvert objekt forskellige koordinater på x-aksen. Platformene er forskellige i design.

Titel	Kollision/interaktion
Beskrivelse	Frøen skal kunne lande og stå på platformene, altså den skal ikke kunne bevæge gennem platformen og skal blive med dens y-værdi hvis den er ovenpå.

Titel	Point
Beskrivelse	Der skal ligge samleobjekter til spilleren, som kan bruges til at vise hvor god spilleren er.

Titel	Grafik
Beskrivelse	Der skal laves en række assets til at repræsentere spillets grafik, såsom baggrund, karakter og meget andet.

Titel	Animationer
Beskrivelse	Animationer skal mindst optræde en gang på en eller anden måde i spillet

Titel	Dynamisk Baggrund
Beskrivelse	Spillet skal have en dynamisk baggrund for bedre at kunne simulere bevægelse i spillet.

Titel	Fjende
Beskrivelse	Der skal være en fjende som udfordre spilleren på en eller anden vis.

Tasks

Titel:	Frø/Spiller
Estimeret tid i alt:	5 timer
Prioritet:	10



Task 1:	Lav klasser med nødvendige variabler og metoder
Estimeret tid:	2 timer

Task 2:	Lav en tegning der bevæger med spilleren
Estimeret tid:	3 time

Titel:	Bevæge rundt og interagere
Estimeret tid i alt:	8 timer
Prioritet:	10



Task 1:	Spilleren skal kunne bevæge sig rundt
Estimeret tid	2 time

Task 2:	Spilleren skal kunne interagere med fastlåste objekter
Estimeret tid	3 timer

Task 3:	Skal kunne skyde
Estimeret tid	3 timer

Titel:	Fastlåste objekter
Estimeret tid i alt:	14 time
Prioritet:	10



Task 1:	Lav klasse med variabler og metoder.
Estimeret tid:	2 timer

Task 2:	De skal kunne laves med tilfældig størrelse og koordinater.
Estimeret tid:	3 timer

Task 3:	Lav tegninger til platforme
Estimeret tid:	6 timer

Task 3:	Lav kode til kollision
Estimeret tid:	3 timer

Titel:	Fjender
Estimeret tid i alt:	10 timer
Prioritet:	10



Task 1:	Lav klasse med variabler og metoder.
Estimeret tid:	2 timer

Task 2:	Fjenderne skal kunne registrere interaktion med spilleren og platformene.
Estimeret tid:	2 timer

Task 3:	De skal kunne laves med tilfældig størrelse og koordinater.
Estimeret tid:	2 timer

Task 4:	Lav tegninger til fjender
Estimeret tid:	4 timer

Titel:	Fysiske fænomener
Estimeret tid i alt:	4 timer
Prioritet:	10



Task 1:	Lav tyngdekraft-metode i Spilleren
Estimeret tid:	1 time

Task 2:	Lav vind-metoden
Estimeret tid:	1½ time

Task 3:	Lav vindmodstand-metoden
Estimeret tid:	1½ time

Titel:	Point
Estimeret tid i alt:	10 time
Prioritet:	30



Task 1:	Lav en point-tæller
Estimeret tid:	1½ time

Task 2:	Lav SQLite database og hent SQLite library
Estimeret tid:	2½ time

Task 3:	Lav kode til at gemme og hente score
Estimeret tid:	3 time

Task 4:	Lad spilleren starte, hvor de sluttede
Estimeret tid:	3 time

Titel:	Baggrund
Estimeret tid i alt:	10 timer
Prioritet:	20



Task 1:	Lav baggrunds-elementer
Estimeret tid	7 time

Task 2:	Lav kode til visning af tegninger med parallax
Estimeret tid	3 time

Titel:	Samleobjekter
Estimeret tid i alt:	6 timer
Prioritet:	20

Task 1:	Lav klasse med variabler og metoder.
Estimeret tid	2 timer

Task 2:	De skal kunne laves med tilfældighed
Estimeret tid	2 timer

Task 3:	Lav tegninger til objektet
Estimeret tid	2 timer

Titel:	Grafisk forbedring
Estimeret tid i alt:	15 timer
Prioritet:	40

Task 1:	Pænere spiller bevægelser
Estimeret tid	3 time

Task 2:	Rotér samle objekter og fjender
Estimeret tid	3 time

Task 3:	Forbedret baggrund
Estimeret tid	3 time

Task 4:	Flere platforme/fjender
Estimeret tid	6 time

Planlagte iterationer med dertilhørende datoer

Det er vigtigt at have et par iterationer, da de kan hjælpe med udviklingen af programmet samt holde styr på hvilke elementer skal konstrueres først. I gruppen har man lavet 4 iterationer, alt efter vigtighed. Tidsestimaterne er angivet i timer. Eftersom at hvert gruppemedlem har ca. 110 elevtimer, så har man brugt gruppens samlede antal timer og fordelt det ud over de fire iterationer for at give et realistisk bud på arbejdet.

Realistisk set, så vil det ikke være muligt at arbejde 100% koncentreret 100% af tiden. Sygdom og lignende kan også gå ud over arbejdet. Derudover skal der også tages højde for den tid, hvor vi diskutere mindre detaljer og beslutninger om spillet. Disse kan sammenlagt tage lang tid, da der skal være enighed om spillets retning inden, der bliver lavet for meget, for at undgå spildt arbejde. Selvom taskene både har arbejde grafisk og kode-mæssigt (samtidig med rapportskrivning), så vil der også være tidspunkter, hvor nogle gruppemedlemmer vil have svære ved at arbejde, da deres arbejde afhænger af noget, som et andet gruppemedlem er i gang med. Derfor antages det, at 70% af den angivne elevtid bliver brugt på taskene. Yderligere skal der også bruges tid på eksempelvis bugfixing og play-testing. Rapportskrivningen er også en del af projektet, og denne del optager meget tid. Med eftertanke antages det, at gruppens samlet tid er splitter på midten når det angår rapportskrivning versus produktudvikling.

Iterationer og tasks prioriteres i intervaller af 10, fra 10 og op, hvor 10 er den højeste prioritet. Specifikt, så står iterations-listerne med de tasks, som gruppen i sidste ende fik lavet til iterationerne. Ændringen af iterationerne er dokumenteret under logbogen.

Iteration 1: Grundlæggende gameplay

Deadline d. 01/05/2022. 175 elevtimer i alt.

Titel:	Frø/Spiller	Bevæge rundt og interagere	Fastlåste objekter	Fysiske fænomener	Rapportskrivning osv.
Tidsestimat:	5	8	14	4	Rest
Prioritet:	10	10	10	10	

Iteration 2: Design

Deadline d. 05/05/2022. 100 elevtimer.

Titel:	Samleobjekter	Baggrund	Rapportskrivning osv.
Tidsestimat:	6	10	Rest
Prioritet:	20	20	

Iteration 3: Udvidet gameplay

Deadline d. 08/05/2022. 70 elevtimer.

Titel:	Fjender	Rapportskrivning osv.
Tidsestimat:	10	Rest
Prioritet:	20	

Iteration 4: Forbedring

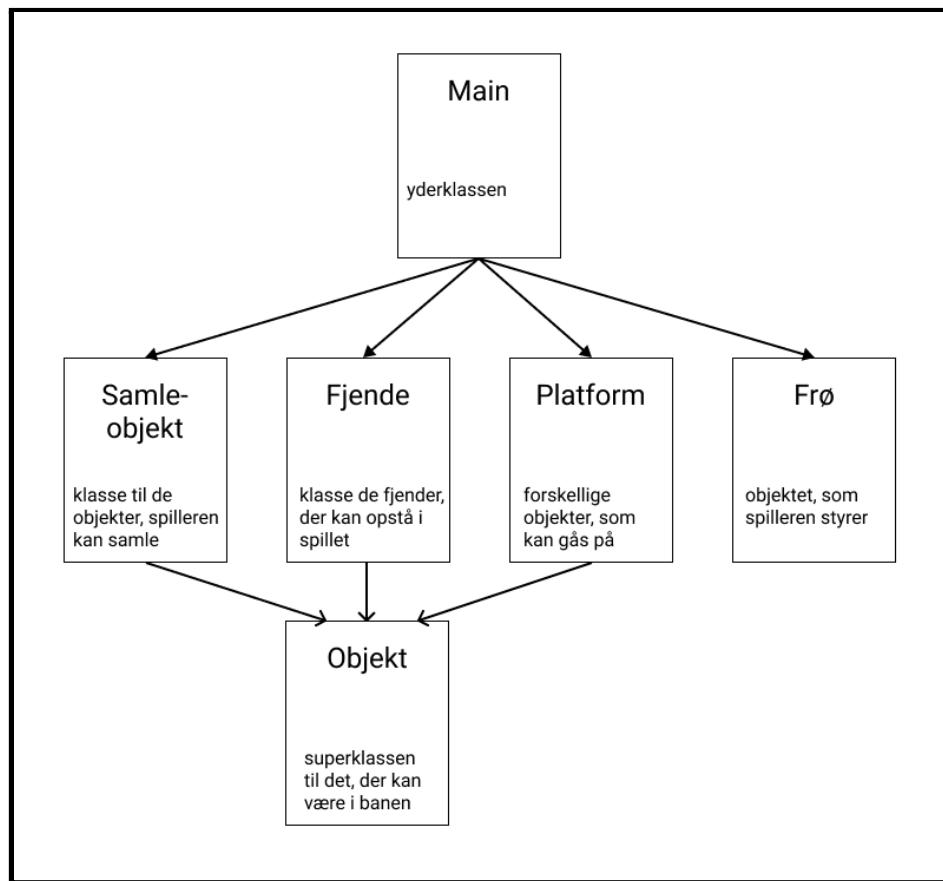
Deadline d. 13/05/2022. 70 elevtimer.

Titel:	Point	Grafisk forbedring	Rapportskrivning osv.
Tidsestimat:	10	15	Rest
Prioritet:	30	40	

Klassediagram

Ud fra de flere forskellige user stories, tasks og iterationer ovenover, har gruppen lavet et simpelt klassediagram over de klasser, som de forventer at skulle bruge i programmet. Programmet kommer til at have en enkelt hoved-klasse, som kommer til at indeholde og have styr på de fleste andre klasser.

De udfyldte pilespidser betyder, at klassen indeholder objekter af den klasse, som peges på. Dem der ikke er udfyldte betyder, at klassen nedarver fra denne klasse (se figur 14).



(Figur 14: Illustration af gruppens midlertidige klassediagram til projektet.)

Tekniske valg

Hver gang gruppen har skullet lave noget arbejde til projektet, om det så end var kode til programmet eller skitsere illustrationer af prototypen, har gruppen skulle bruge de bedste tekniske programmer som de har til rådighed. I dette afsnit bliver der skrevet om hvilke programmer gruppen har og kommer til at bruge i projektet, og hvorfor de er blevet brugt i stedet for andre.

Adobe Photoshop

Når gruppen har skullet lave de forskellige assets til spillets design og indhold, brugte gruppen Adobe Photoshop for at designe både de forskellige karakterer, platforme, objekter, osv.. Photoshop har været nemmest at bruge til at laver karakterer og avancerede former

siden de har mange flere former som ville være sværere at lave i f.eks. Adobe Illustrator, hvor det er primært vektorbaseret med former. Gruppen bruger Photoshop både fordi det er det program som gruppens medlemmer har mest erfaring med til at skitsere, og er mere simpelt at bruge end Krita eller GIMP, som er nogle af de større og avancerede tegneprogrammer.

Adobe Illustrator

For at designe de forskellige platforme og bygningerne til spillet, har gruppen brugt Adobe Illustrator, da dette program er bedre til at lave mere simple men tonede former. Gruppen bruger dette program eftersom at gruppen ikke havde andre tekniske programmer som mindede om dette, og kunne erstatte Illustrator.

Hubspot

For at lave personaerne til målgruppeanalysen, har gruppen brugt hjemmesiden [Hubspot](#). Hubspot forsyner tonsvis af skabeloner til forskellige moduler. Hubspot er nemt at bruge og gratis at anvende. Hubspot er en af flere digitale værktøjer der forsyner digitale persona skabeloner, såsom f.eks. Xtensio og Figma. Det er dog blevet besluttet at bruge hubspot i stedet for, da det er yderst simpelt og ikke kræver en masse tid at fremstille og skrive personaer.

Java og Processing

Til at køre programmet bliver der brugt Java version 17 og Processings Java-library. Java version 17 har et par enkelte flere features end ældre version såsom 5. Det er features såsom brugen af lambda-funktioner, Stream-objekter, record-klasser og simplere switch-syntaks. Processing bliver brugt som et Java-library, da det tillader at bruge et bedre kode-redigerings program end Processings egen IDE.

Visual Studio Code

VS Code er blevet brugt hele vejen igennem projektet til at redigere, debugge, køre og bygge koden. VS Code er et meget fleksibel kode-redigerings-program, eftersom man kan downloade officielle og brugerlavet udvidelser, der hjælper endnu mere med programmerings-processen. Der findes bl.a. udvidelser der hjælper med at lave Java-projekter, Java language support, debugging, kompilering, osv.. Derudover er der også blevet gjort brug af en SQLite udvidelse, der hjælper med håndteringen af en SQLite-database, som sammen med BezierSQLib-biblioteket, der er et Processing-library, gør det nemt at arbejde med en SQLite-database i programmet og gemme vigtige information.

SQLiteManager

Det er et program, som er blevet brugt i undervisningen og kan bruges til at lave og redigere en SQLite database.

Brugertest

Her er den planlagte fremgangsmåde til hvordan testen af produktet ville foregå. Det er godt at have en planlagt test af sit program, da det kan hjælpe med at bestemme hvilke elementer der fungerer og hvilke kan forbedres. Det er også godt at høre omkring folks meninger med programmet ang. design og tasks.

Planlægning af test

Produktet er et spil som omhandler en frø der forsøger at forholde sig i live ved at hoppe rundt fra platform til platform. Spillets kerne mekanismer omhandler de såkaldte "gravity spikes" hvor spillets tyngdekraft pludseligt kan ændre sig til at være den modsatte retning.

Før demo-testen vil gruppen bede testperson om at besvare en række spørgsmål vedr. spilgenren for at sikre at programmet er tiltalende til testpersonen og om man har nogle forudbestemte meninger om den slags spil. Testpersonen er anonym.

1. Har du spillet mange platform spil før?
 - a. Hvis ja: Hvad finder du sjovt i platformspil & hvilke platform spil har du spillet?
2. Hvilke spil genre kan du bedst lide?

Under demo-testen vil gruppen bede testperson om at foretage en række handlinger i spillet for at sikre at programmets design og tasks er intuitive nok.

1. Bevæg fra side til side og hop op og ned. Se om testpersonen får en god følelse når man bevæger sig rundt eller om noget skal ændres.
2. Prøv at interagere med fjender og fastlåste objekter. Se om testpersonen får en god følelse når man interagerer eller om noget skal ændres.
3. Samle en mønt
4. Gentag denne proces 3-4 gange og se om spillerens attitude til spil mekanismer og/eller elementerne ændres undervejs, f.eks. bliver spilleren tilpasset til spillets unikhed, eller er det konstant?

Efter testen vil gruppen bede testperson om at besvare en række spørgsmål vedr. spillet for at sikre at programmets design og tasks er intuitive nok for testpersonen og om det er noget der skal ændres på.

1. Hvordan føles spillets karakter at spille?
2. Hvordan var sværhedsgraden?
3. Er spillet sjovt?
4. Minder spillet dig om nogle andre spil?

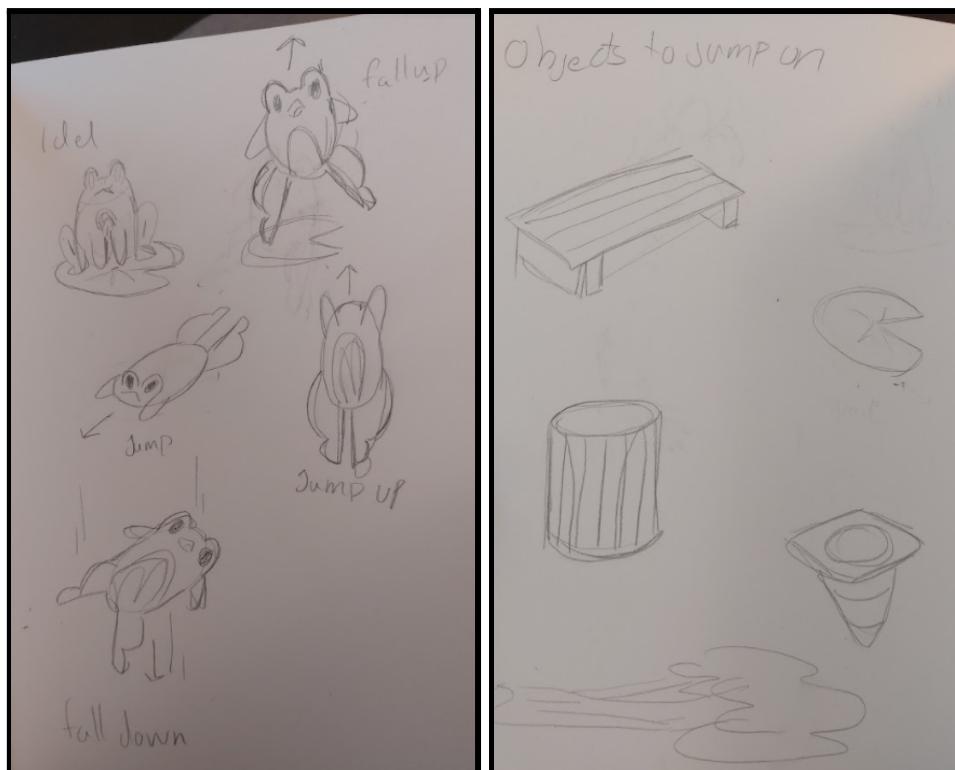
Efter at have foretaget testene, vil gruppen diskutere resultaterne og konkludere om noget skal ændres på programmet eller ej. Hvis det skal, skal muligvis endnu en test foretages for at dobbelt tjekke hvad brugeren mener.

Produktudvikling

Skitser

Til at starte med, har gruppen lavet en masse skitser, for at få en bedre visualisering af hvordan selve spilleren og spillet kommer til at se ud. F.eks. har gruppen lavet skitser til karakteren, baggrunden og de forskellige platforme som frøen skal kunne interagere med. Man har også skullet skitsere de forskellige bevægelser som frøen skal kunne, som at hoppe fra side til side eller idle (se figur 15 og 16). Dette er vigtigt at have med i et computerspil, da det gør bevægelsen af karakteren mere flydende og dermed mere roligt at se på.

For at give frøen en smule personlighed har gruppen tænkt på at lave et feature for frøen, som vil være at når man er i starten og ikke begynder spillet, så blinker frøen og falder i søvn. Dette er inspireret af spil som Super Mario Bros. og andre platform spil, hvor at karakteren har nogle specielle træk såsom at falde i søvn eller tjekke tiden når spilleren forlader spillet åbent. Dette kan også ændres til at eliminere spilleren hvis man har begyndt spillet og er inaktiv.



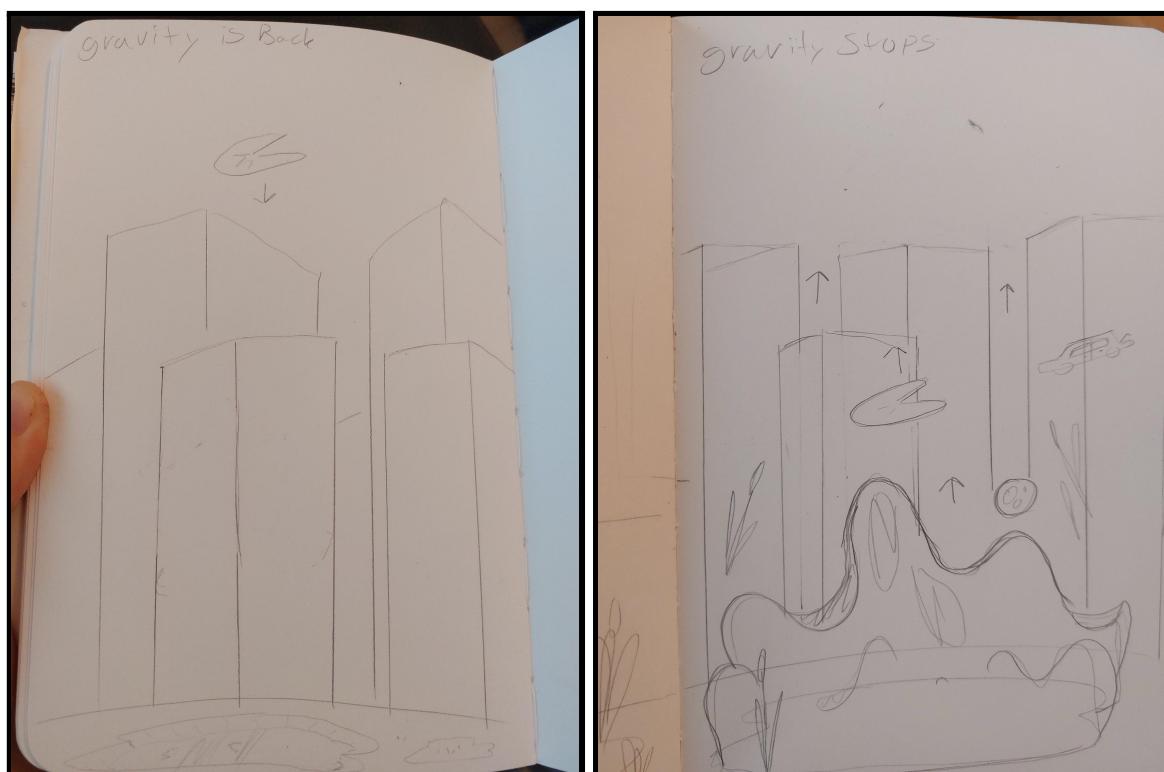
(Figur 15 og 16: Lo-fi skitser af spil karakteren og platformene til programmet.)

Hovedkarakteren/spilleren er en frø, som skal have forskellige bevægelses-tilstande som den kan være i. På figur 15 og 16 kan man se hvordan de forskellige bevægelser ser ud i forhold til hvilken vej man hopper til og sidder på. På den anden illustration kan man se nogle skitser af de platforme som kan ende med at være i spillet. Eftersom at baggrundshistorien for spillet fortæller at man befinner sig i en bypark eller et skovområde,

så virkede det som en god idé at designe platforme som minder om ting der eksisterer i byparker og skovområder. Det kan f.eks. være grene, lilje puder, skraldespande og bænke (se figur 16).

Når det kom til at skitsere baggrunden, mener gruppen at et skovområde eller en by ville være oplagt at vælge, da det også passer med temaet for spillet. Man kunne så ændre på skoven/byens perspektiv for at illustrere at byen er langt væk og giver fornemmelsen at man svæver op eller falder ned i spillet (se figur 17).

Til baggrunden er der derfor indtil videre blevet skitseret en meget simpel by med flere lag og forskellige perspektiver. der er blevet skitseret en by fra toppen og fra bunden da frøen bevæger sig opad. Der er derudover blevet skitseret forskellige platforme for spillet, for at give en reference til hvor store både platformene og baggrunden skal være. Når man så når længere op i spillet, vil platformene langsomt blive udskiftet med skyer, for at vise brugeren at man er nået længere i spillet (se figur 18).

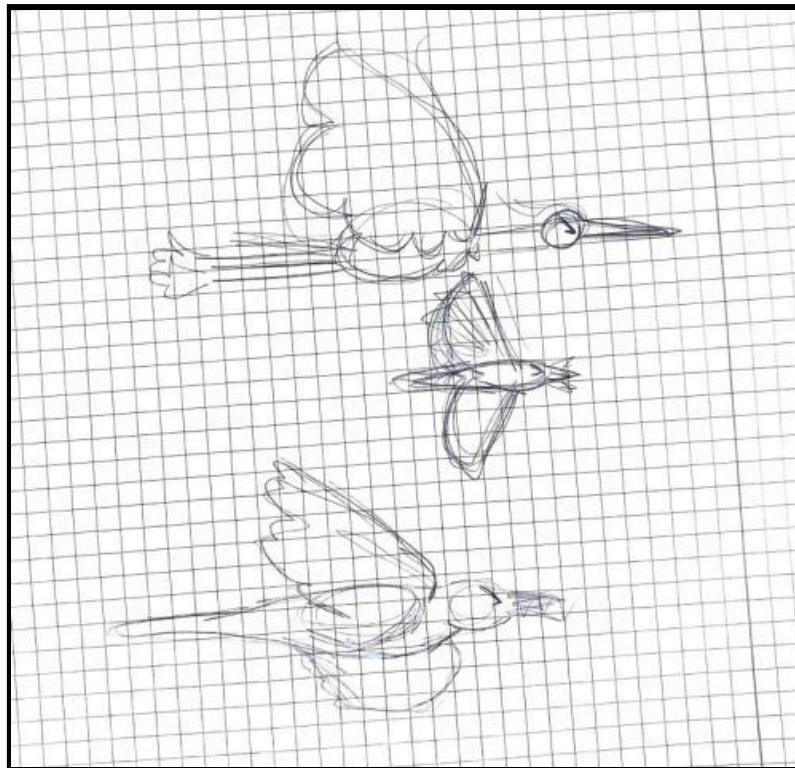


(Figur 17 og 18: Skitser af karakter, platforme samt baggrunden.)

Til sidst har gruppen skitseret et samlerobjekt for at sørge for at spilleren er aktiv og er motiveret. For at gå med temaet, vil samleobjektet komme til at være en blanding af en flue og en mønt, da en flue er frøens naturlige snack, og mønten er standarden for samleobjekter i platformspil.

For at gøre spillet svært har gruppen tænkt os at implementere et par fjender. Dette tænkte gruppen af en af dem kunne være en fugl, fordi det er en af frøens naturlige fjender. Til fjenden har man lavede nogle få skitser af forskellige fugle, som potentielt kunne være i spillet (se figur 19). Gruppen tænkte først på en fiskehejre, men denne fugl følte gruppen

ikke passede særlig godt ind i bylivet, så gruppen tænkte at den skulle være mindre. Den almindelige Grackle, en lille fugl som lever i moser og parker, og endda spiser frør. Denne fugl passede perfekt og havde kriterierne til at være en fjende i spillet. Gruppen har også lavet to andre fjender, men har ikke nogle skitser.



(Figur 19: Skitse af forskellige fugle, som kunne bruges som fjende.)

Farveteori + gestaltlove

En vigtig ting for ethvert spil er farveteori. De forskellige farver for et computerspil er ofte udtaenkt meget grundigt for om noget skal stå ud i mængden, blande sig i mængden, være modsætninger, ligheder, kategorisere, osv.. Gruppen har derfor skullet tænke på hvordan farverne for de forskellige elementer skulle se ud sammen og hvilke følelser som farverne udgiver. Eftersom at spillet ikke kan være et AAA-spil, går gruppen efter tegneserie farver end realisme, så er det ikke så vigtigt hvis farverne ikke er 100% korrekte.

Feks. når spilleren kigger på frøen eller platformen som frøen står på eller fjenden og mønten. Alle farver er vigtig for at spillet skal fungere, og derfor har gruppen brugt så meget fokus på farveteori til at hjælpe med at vælge de bedste farver for hvert element.

Frøen

Frøen er hovedkarakteren i spillet. Selve spillet er centreret om frøen, og spilleren kommer altid til at se frøen uanset hvor langt man er kommet i spillet. Når det så kom til at vælge en farve til frøen, gik gruppen med en meget indlysende farve, som er grøn. Frøen vil bestå af både lysegrønne og mørkegrønne farver. Grunden til det er fordi når man forestiller sig en frø i hovedet, er det altid en lille mørkegrøn frø med lysegrøn mave.



(Figur 20: Rentegning af hovedkarakteren som spilleren styrer.)

Der bliver derfor gået efter de mest beroligende og genkendte grønne farver for at give det bedste billede af en normal frø. Ift. farvepaletten, er der flest farver i en mat grøn farve, som skal vise at frøen er mere venlig, siden den er hovedkarakteren. Dernæst kommer to slags lysegrønne farver for maven og kanten, og sort og hvid for øjnene. For ikke at have få mange farver på hovedkarakteren, siden man kigge mest på den, anvender gruppen ikke mere end 3 slags farver for frøen.

De tre farver består af den dominerende matte grønne farve, som er selve frøens nøgle farve, dvs. den farver der fylder mest af tegningen (se figur X). Den anden farve er for at skabe små detaljer på frøen, som prikker på ryggen, mund og næse. disse simple detaljer er med til at vise hvorfra man kigger på karakteren, når man se pletterne på ryggen ved man at man kigger på karakteren bagfra. og den sidste sorte farve som bruges til øjnene. Disse tre farver er vigtige at have for at skabe perspektiv og for at overbevise spilleren at det er en frø.

Denne form at farve karaktere eller apps, gør at man laver en farvepalette som balancere farver man vælger. Denne regel hedder 60-30-10 reglen, og bliver brugt således. 60 % karakteren skal være en dominerende farve, 30 % skal være den sekundære farve eller tekstur, og de sidste 10 % skal være en accent. Lige Netop dette bliver brugt for at lave frøen.



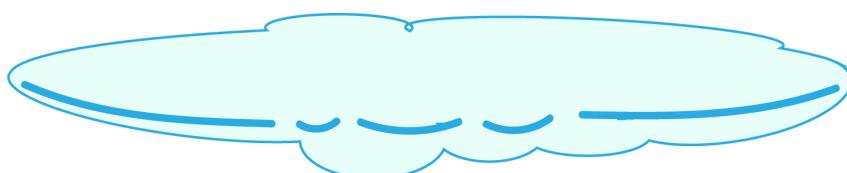
(Figur 21: 60-30-10 regel.)

Siden gruppen har at gøre med et dyr, så skal farverne matche de naturlige farver man se på de rigtig frøer. den naturlige jord grønne farve, gør frøen en smule mere naturlig, selvom den er meget tegneserieagtig.

Platform

Platformene er også særlig vigtige, da de ændrer sig jo højere man kommer op i spillet. Når platformen først blev lavet, ville gruppen gerne have at de skulle virke både mere realistiske, og at de ikke skulle blande sig med karakteren så man ikke kunne se forskel mellem dem. Farverne som bliver brugt til platformene er derfor mere rene og skrigende i forhold til frøens farver. Farverne er også mere naturlige end frøens, f.eks. bænken er lavet af træ så derfor er den brun. Vejarbejde keglerne er orange som de normalt er i den virkelig verden. Det er vigtigt for gruppen at vælge de rigtige farver når man vil have sine assets til at se naturlige ud, men hellere ikke have en anderledes farve stil end resten af spillet.

Ligesom når man kan forestille en frø i sit hoved, så forestiller man også andre ting. Det betyder så at når man ser en genstand såsom grene, skraldespande, osv., laver hjernen automatisk et billede af den genstand, og et sæt farver som man kan tydeligt se på billedet. Ved at tænke sig frem til dette, har gruppen lavet nogle farver for platformene (se figur X). Feks. Når man tænker konstruktion kegle, tænker man skrigende orange samt en hvid strib, og dette koncept bruges til alle platforme. I forhold til skyerne, har gruppen valgt at gøre dem meget lyse blå i stedet for hvide.



(Figur 22: Rentegning af en af platformene i spillet.)

Gruppen valgte den blå farve til skyerne, da tegneserier normalt maler skyer lyseblå, og siden spillet er meget tegneserieagtig og en smule barnligt, virkede det som en god idé at følge temaet og male skyerne blå i stedet for. En gestalt lov som platformene følger er loven om lighed. Eftersom at de forskellige skyer ligner hinanden i udseende og farver, må de også passe sammen. Park objekterne har samme farver og passer sammen, og går så ind for denne lov.

By

Storbyen er en vigtig del af spillet, siden den sætter scenen for spillet. En storby bliver ofte set som at være en livlig og farverig by, især i tegneserier. Derfor har byen en masse skrigende/rene farver, som passer sammen med platformene da de alle kommer fra det samme sted. De skrigende farver giver liv til byen og det skaber opmærksomhed. Dog er farverne dulmet så de ikke tager opmærksomhed væk fra selve spillet eller andre elementer.

Storbyen er opbygget af primære farver som blå, en blanding af lilla og bodo rød og mørke lilla. Alle farverne er bevidst matte, så de ikke står så meget ud i mængden eller blander med spilleren.



(Figur 23: Rentegning af hele byen i baggrunden.)

Da det kom til at farve storbyen var det vigtig at have det rigtige perspektiv for farverne, Dette betød at den første del af byen skal være lysere end den bagerste del af byen. Denne effekt gør at nogle bygninger er længere væk, mens resten af bygningerne er tættere på. Loven om lighed bliver også anvendt her, da man grupperer tætte og fjerne bygninger ved at give samme farver og højde.

Når alle high-fi skitserne af bygningerne bliver sat sammen, kan man tydeligt se hvordan en by bliver formet. En storby med liv og massere af farve. Når de forskellige assets også bliver sat sammen kan man se hvordan farverne og skaleringen af de forskellige bygninger virker. man kan tydeligt se hvilke bygninger der er forrest og hvilke der er bagerst.

Fjende 1 - Fuglen

Da det kom til at give fuglen farve, ville man brug naturlige farver, som resten af verdenen. I virkeligheden er fuglen sort, med et skær af mørke blå fjer. gruppen prøver at imitere de naturlige farver i fuglen ved at bruge en specifik pensel som laver et gradient, som skal ligne de mørkeblå fjer. Som alt andet, så er fjenden meget simpel, simple farver og simpel form.



(Figur 24: Rentegning af fjende nr. 1 for spillet.)

Fjende 2 - Flyet

Flyets design er meget gammeldags for at give en tegneserieagtig følelse, og man har allerede dannet sig en vis farveteori til spillet, så flyets farver vil være skrigende og rene, og for at skabe mest modsætning til baggrunden og hovedkarakteren er flyet farvet med rødt, gult, sort og lyseblåt.



(Figur 25: Rentegning af fjende nr. 2 for spillet.)

Fjende 3 - UFO

UFOen er meget enkel i design og farver, der er ikke meget at nævne på nær at den også står i kontrast til baggrunden og hovedkarakteren.



(Figur 26: Rentegning af fjende nr. 3 for spillet.)

Samleobjekt

Til spillet ville gruppen gerne have et point system, derfor tænkte gruppen det ville være sjovt at have en flue som flyver med en mønt. Gruppen tænkte at frøen skal kunne fange fluen med sin tunge, og på den måde samle point ind. Fluen som er blevet lavet, er sort og har meget få detaljer. Gruppen er gået efter at den skal ligne at den kunne være med i en tegnefilm. Den har de naturlige farver som en flue, men er meget cartoonish.



(Figur 27: Rentegning af samlerobjekt for spillet.)

Øjnene på fluen er røde og ligger i kontrast i forhold til kroppen og det samme gør dens vinger. Siden fluer er meget små, har gruppen fået det til at ligne at fluen kæmper med at holde mønten oppe, som ville give mening siden den er så meget mindre end mønten. Selve karakteren har en rund silhuett, som gør at den ser venlig ud, dette gør at brugerne ved at

denne karakter ikke vil gøre spilleren ondt, og de så dermed ved at de kan stole på at få mønten fra fluen.

Animation

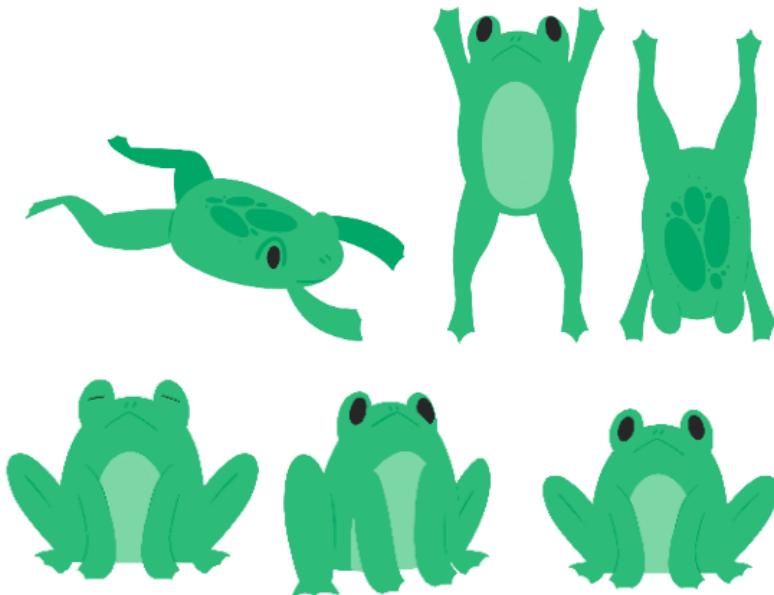
Når man laver animation kan man hurtigt ende op med at have flere tusindvis af tegninger for en simpel animation. Det kan tage flere timer at lave. Dog har gruppen ikke nok tid til at skulle lave masseproduktion af tegninger eller billeder, derfor bruger gruppen noget som hedder "two frame animation", som er en simpel måde at animere på hvor man både kan vise massere af bevægelse og liv.

Den måde gruppen bruger to billede animation er ved at tegne et billede, hvor det første billede kunne være et stille billede, og hvor det andet kan være af frøen som hopper. Her kunne man selvfølgelig have massere af billede imellem de to, som ville gøre det hel del glattere, dog valgte gruppen den simple form for animation. Til alle karakterene, bruger gruppen to billede animation.

Karakter

For at give frøen liv har gruppen valgt at den skulle animeres i dens bevægelse. Dette bliver gjort så frøen ikke bare sidder stille når man hoppe op, frem og tilbage, og giver spillet mere liv i sig og gør det mere spaendende for spilleren at spille. Derudover har frøen flere billeder, siden den skal kunne hoppe rundt på banen og se ud som en normal frø.

Selvom det kræver flere tegninger, kan man let genbruge billeder ved at spejlvende dem. Frøen og stedet spillet tager plads i er mere tegneserieagtig end realistisk. Karakteren har en rund silhuet, som bliver brugt til blød karaktere og venlige karaktere så frøen ser meget venlig og sød ud.



(Figur 28: Rentegning af animations billederne for frøen.)

Flue

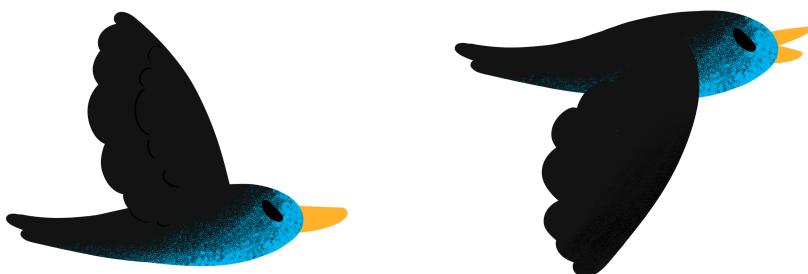
Den lille flue, som holder mønster, skal bruges til at spilleren skal kunne samle point. Den er mere cartoonish en frøen og er meget simpel da den ikke har en større rolle i spillet. hvor frøen her flere karaktertræk som prikker på ryggen som ikke er nødvendig men giver personlighed.



(Figur 29 og 30: Rentegning af animationsbilleder for point fluen.)

Fjende

For fuglens animation, bruger gruppen to billede animation, som der bliver gjort med frøen. Her skal man bare bruge to billeder for at vise fuglen flyver. Et billede af fuglen med vingerne nede og vingerne upper var det eneste man havde brug for.



(Figur 31 og 32: Rentegning af animationsbilleder for fjende nr. 1.)

Parallax

Spillets baggrund anvender en parallax effekt. En parallax effekt er hvor baggrunden er opdelt i flere lag hvor hver del af laget bevæger sig med forskellig hastighed. Et eksempel kunne være en baggrund hvor det forreste lag bevæger sig hurtiger end det bagerste lag. Parallax effektens formål er at simulere dybde i et billede ved at bevæge forgrunden separat fra baggrunden. På baggrund af dette er spillets baggrund konstrueret af en række elementer såsom bygninger og bjerge frem for et fast stillbillede.

Endeligt design

Fjender

Platform spil plejer relativt at være rimelig nemme, derfor plager disse spill have forskellige former for forhindringer, som skal gøre spillet svært. Dette kan gøres på mange måde, med platforme, som enten giver skade eller langsomt begynder at forsvinde, eller fjender som flyver eller falder gennem luften mens man hopper fra platform til platform. Til platform spillet har gruppen derfor valgt at lave flere fjender for at gøre spillet sværere og mere interessant at spille.

Gruppen har skullet vælge flere fjender siden at frøen kommer længere op mod atmosfæren, og det ville ikke give mening at have en fugl hvor for eksempel fly flyver, og dette er så grundet til gruppen har valgt en UFO og en flyver. Designet for frøen, flyet og UFOen er valgt fordi de også er tegnet således i tegneserier, og spillet har et vis tema at skulle følge.



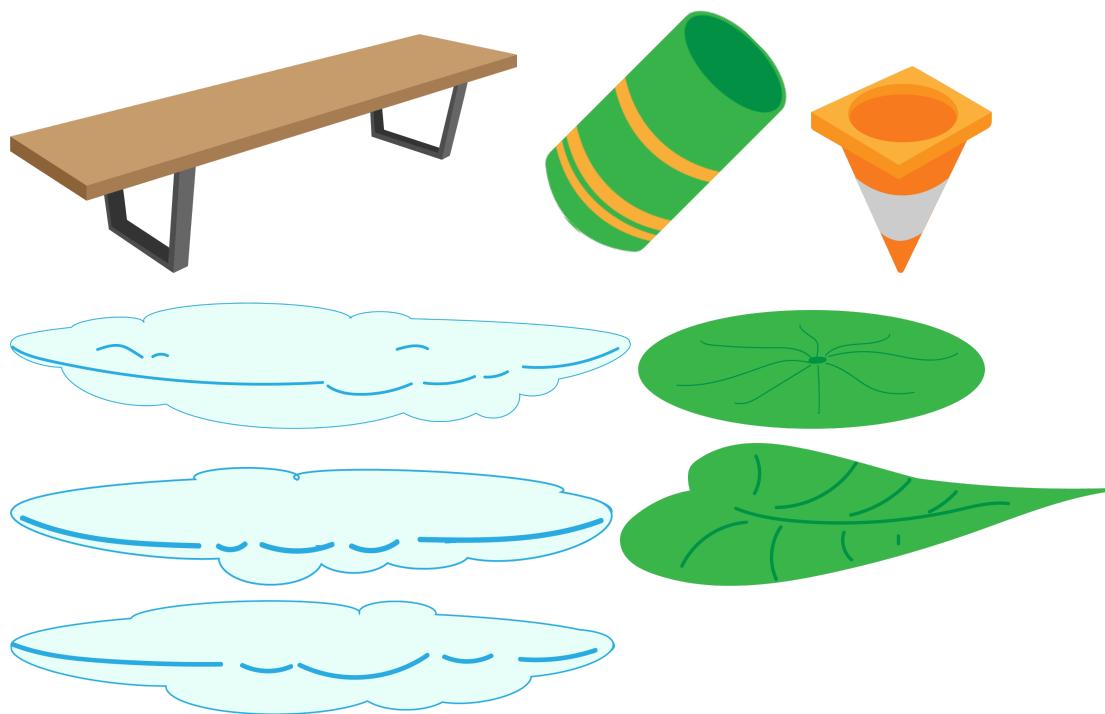
(Figur 33: Rentegning af fjender nr. 1, 2 og 3 for spillet.)

Platforms

Spillet foregår inde i byens lokale park, herfra kommer alle de objekter som frøen skal hoppe oppe til. Gruppen har valgt simple objekter som man forventer at se i en by park, som bænke, affaldsspande og store blade, ud fra hvor byen ligger. Dog har gruppen også brugt nogle af de mest kendte ting man se i en by, som en konstruktion kegle.

Som sagt når man kommer længere op begynder platform objekterne at blive skyer. Det viser visuelt at spilleren begynder at komme længere op. Dette er en anden måde at vise spilleren de gør det godt, i stedet for at bruge en counter. Der er blevet lavet tre former for skyer, for at der er variation i platformerne. Skyerne er afrundet for at lige de er bløde som skyer ofte bliver antagelse for at være. Farven for sky platformene er lyseblå, endnu et element som giver til det bløde udseende.

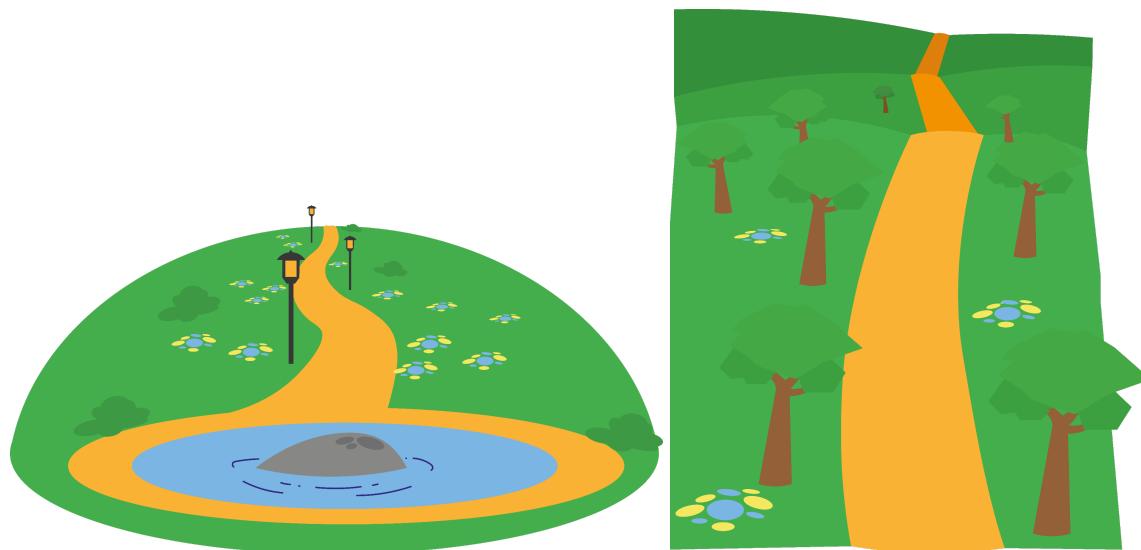
Den sidste sidste platform som er blevet lavet er en åkande, som ofte bliver set sammen med en frø. Det var oplagt at lave og derfor blev lavet.



(Figur 34: Rentegning af alle platforme som spillet gør brug af.)

Forgrund & mellemgrund

Fore- og mellemgrund er her for at give dybde i spillet, det skal også hjælpe med at skjule enderne på højhusene. Forgrunden, som er søen med den store sten, bliver brugt som startpunkt for spillet. Her skal frøen starte med at hoppe fra og så til en af de mange platforme. Mellemgrunden ligger bag forgrunden og skal hjælpe skjule enderne af højhusene som forgrunden ikke kan nå. Så begge billede har et formål i at gøre spillet pænere.

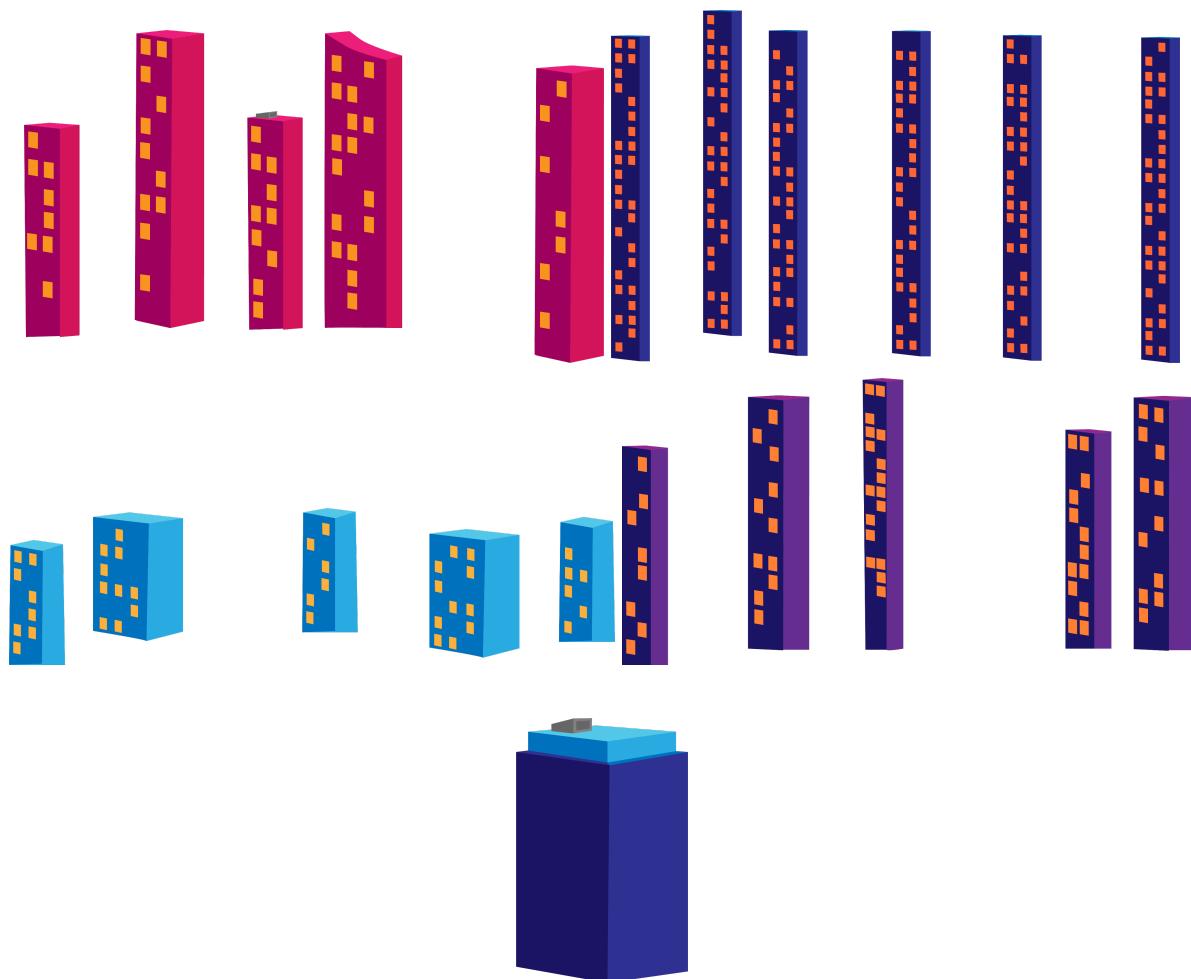


(Figur 35 og 36: forgrund og mellemgrund for spillet, bruges til at kunne skabe dybde i spillet.)

Baggrund

Spillet foregår i en park som ligger inde i en storby. Så til baggrunden har man skulle lave en masse højhuse som skal være storbyen. For at skabe perspektiv og for at vise byen er stor, har man skulle lave flere lag til baggrunden. Dette gøres med størrelsesforskæl og farver. De første bygninger for byen her er lysere farve og er tykkere end de bagerste skyskrabere. For at skabe liv til byen har gruppen skabt variation imellem bygningerne, samt have lys i bygningerne. Variationen imellem bygningerne består af to bygningerne som har en anden form end de andre. Det gør at baggrunden ikke bliver kedelig men ikke tager for meget opmærksomhed væk fra karakteren.

Når bygningerne endelig er i spillet, så ville gruppen gerne have at de skal kunne bevæge sig individuelt. Dette ville gøre at baggrunden ikke er stiv og kedelig. Dette er ikke nødvendigt men det er en fed detalje.



(Figur 37: Rentegninger af de forskellige højhuse og bygninger som udgør den fulde baggrund.)

Forventning af hvordan spillet kommer til at se ud:

Inden man sætter alle sine assets og baggrund ind i spillet, så kan man lave et eksempel på hvordan det kan forventes at se ud. Både byen og platformerne passer godt sammen, de blænder ikke ind med hinanden og det er nemt at skelne mellem dem. Når man laver sådan et eksempel kan det enten betyde at man skal ændre sine assets eller at det er perfekt, og for os kan man se de passede perfekt med hinanden.



(Figur 38 og 39: Udklip af de forskellige elementer til spillet sat op således så det visualiserer hvordan spillet ser ud.)

Man har indsat frøen i et af figurerne for at se om der er nogle stærke kontraster og hvor meget frøen blander ind i mængden. Disse eksempler er med til at checke ren tegningerne for farveharmoni og kontraster.

Gameplay

Spillets gameplay kan beskrives ud fra Scott Rodgers "The three C's" som man kan læse om i bogen "Level up!"

De 3 C'er

Scott Roger definere hvad han kalder for de 3 c'er (The Three C's) i sin bog Level Up!. De 3 C'er beskriver 3 ting der dømmes som at være grundlæggende for udviklingen af et spil. C'erne står for Character, Camera & Controls og hvert C bliver uddybet i bogen.

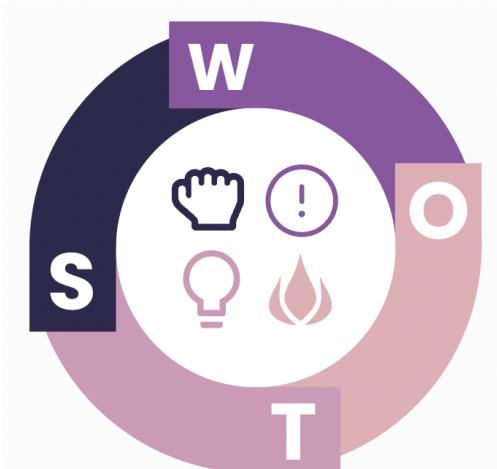
Projektet bruger de 3 C'er til at opnå det bedst mulige spil. Projektet har i større grad haft et fokus på Character delen af de 3 C'er men alle C'er er taget til betragtning i forhold til spillets udvikling. Det var fra start bestemt at den spilbare karakterer (frøen) skulle føles godt at bevæge og hoppe rundt med. Det skal altså betyde at frøen skal have den helt rigtige vægt, hoppehøjde og bevægelseshastighed.

Det næste C er "Camera", og beskriver hvordan et spil skal gøre brug af kameraet og hvad der henholdsvis skal være inde og ude for billedrammen. Spillets kamera har som udgangspunkt fokus på at følge spilleren, så hvem end der spiller altid ved hvor spilleren befinner sig. Spillets kamera har til formål, at gøre det muligt for spilleren at se hvad det næste skridt er uden at lade spilleren se for langt frem. Kameraet er af den grund afgørende for spillets sværhedsgrad. Det er af den grund vigtigt at finde den rigtige kamera indstilling.

Det sidste C drejer sig om spillets controls. Her beskrives controls som noget der skal være udviklet med den hensigt om at det skal kunne blive spillet af mennesker. Spillet skal altså have controls der både er intuitive og brugervenlige. Gruppens spil har af den grund controls der tilrettelægger sig efter industristandarden for bevægelse WASD. På baggrund af den fortaget brugertest har gruppen også implementeret en guide der beskriver hvad spillets controls.

SWOT-analyse

Der er blevet foretaget en delvis SWOT analyse, hvor delvis indebærer at SWOT analysen typisk er rettet imod virksomheder. Denne SWOT analyse fokusere altså på selve spillet og dets Strengths, Weaknesses, Opportunities og Threats, frem for spil udgiveren/producentens.



(Figur 40: Billede af en digital illustration af SWOT-analysen.)

Strengths:

Gruppens stærke sider ligger i at spillet som er blevet udviklet har to former for gameplay, i form af frøen og frøens tunge. Det alternative gameplay ses også igennem de forskellige fjender og den interessante hovedkarakter, som spillet har. Den anden styrke gruppen har fra spillet, er den unikke stil. Stilen er meget sjov og farverig og er en slags som ikke kan ses andre steder end i spillet.

Weaknesses:

Markedet er uendeligt stort for spiludvikling og som et udviklingsteam er gruppen ikke så stor igen. Gruppen har et totalt medlem af fire personer, og ude i den virkelig verden skulle man nok være flere for at det giver mening, og på det punkt trække det gruppen ned.

Opportunities:

Siden spillet er i sin prototype og har ikke haft særlig stor mulighed for marketing, har spillet ikke mange muligheder.

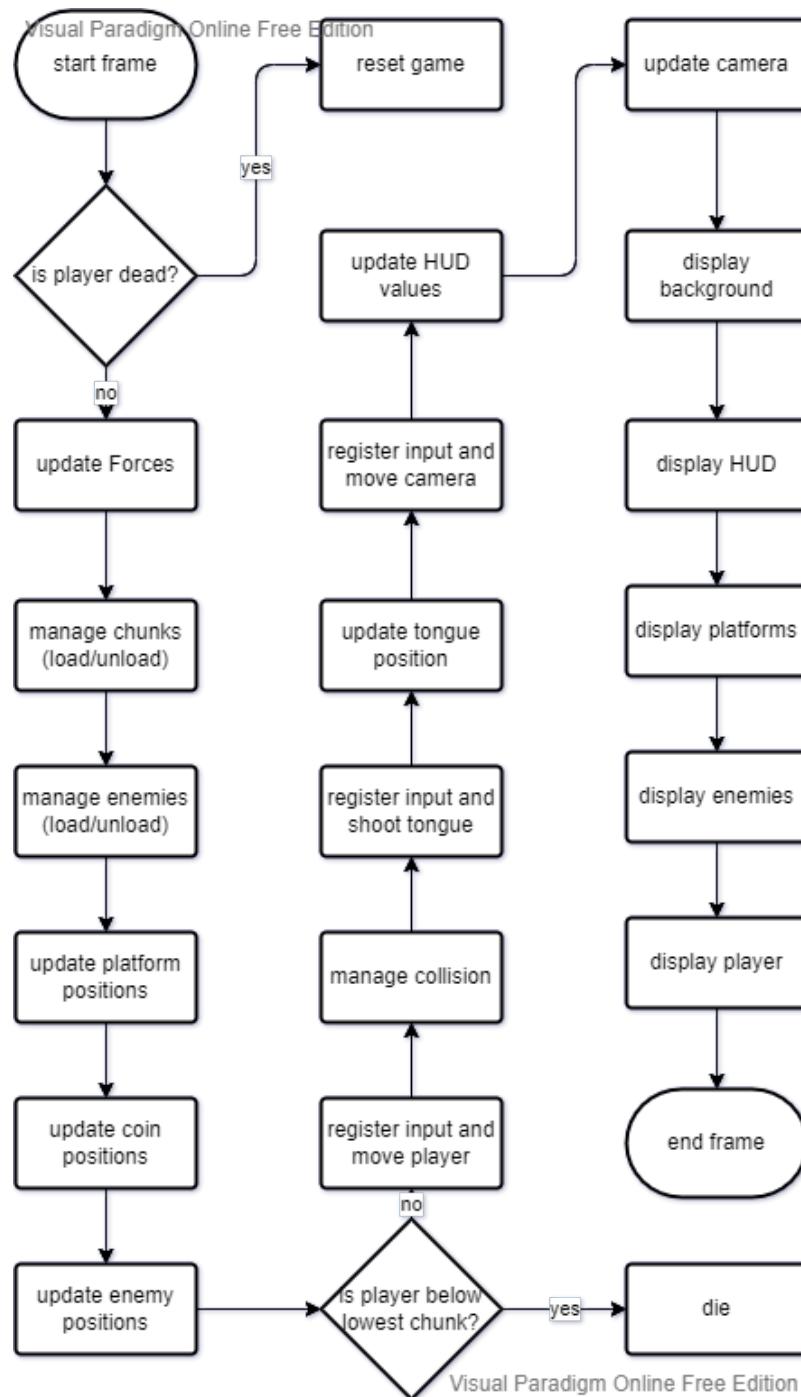
Threats:

En udfordring som ligger for gruppen og spillet, er det ocean af andre spil som minder om det som gruppen har lavet. Den spillertype som spillet ligger i, findes der massere af andre og måske også sjove spil og derfor er der en stor udfordring for gruppen at man skal hele tiden kunne give noget ny eller bedre for at gruppen kunne ligge øverst på listen.

Forklaring af programmet

Overordnet flowchart

Et flowchart over den overordnede process, som programmet går igennem hvert frame. Dette kan bruges til at forklare en kunde, hvordan koden fungerer, uden at de behøver at forstå koden bag.



(Figur 41: Illustration af spillets flow chart.)

Main-klassen

Programmet er lavet ud fra en Main-klasse, som fungere som ligesom den globale klasse, når man arbejder i Processings IDE. Denne klasse indeholder alle objekterne, bortset fra de statiske, og sørger for at opdatere dem hvert frame.

Loading af assets

Når spillet starter, vil programmet som det første, loade alle billederne. Eftersom der ikke er så mange billede-assets, vil det være bedst at loade dem i starten, for at undgå lag midt i spillet. Spillet er også lavet til at springe dette skridt over, når man spiller forfra efter man dør.

World-klassen og chunk-system

Til at holde styr på alle spillets platforme, mønter og fjender bliver der brugt en World-klasse. Eftersom spillet skal blive ved for evigt, så skal spillet være i stand til at generere nye platforme, mønter og fjender for evigt. Derfor skal der også være mulighed for kun at loade de objekter, som er relativt tætte på spilleren,, sådan at programmets hukommelsesforbrug ikke bliver uendelig højt. Derfor bruger World-klassen et chunk-system, hvor hele verdenen bliver delt op i mindre chunks, som hver indeholder nogle objekter. På den måde, kan programmet loade enkelte chunks af gangen og un-loade de chunks, der ligger længere væk.

Chunk-systemet virker på den måde, at der er blevet sat et maksimum antal chunks, som kan være loadet på én gang. Dermed kan programmet beregne, hvilket chunk spilleren befinner sig i, og hvis spilleren befinner sig over midten af de loadede chunks, vil den loade nye chunks over og un-loade de gamle chunks under, indtil spilleren igen befinner sig i midten. Spillet vil ikke generere nye chunks under, hvis spilleren bevæger sig nedad. Dette er, fordi det er meningen, at spilleren skal dø, når de falder for langt ned.

```
//Loads and unloads chunks
private void manageChunks(float playerAltitude) {
    int playerCurrentChunk = getIndexFromAltitude(playerAltitude);
    while(playerCurrentChunk + chunksToLoad / 2 > currentTopChunk) {
        if(chunksLoaded >= chunksToLoad)
            removeChunk();
        newChunk(currentTopChunk);
    }
}
```

(Figur 42: Skærmbillede af programmets kode for chunk-system.)

Platform-, Coin- og Enemy-klassen

Når det kommer til platforme og mønter, så indeholder hver chunk en liste over chunkens platforme og mønter. Eftersom hver chunk har sit eget område, kan programmet bare

genere positioner inden for dette område, til platformene. Antallet af platforme er defineret ud fra chunkens højde, sådan at gruppen har mulighed for at bestemme, hvor mange platforme der skal være, ved forskellige højder. Mønsterne er genereret efter tilfældighed. Hver platform har en angivet sandsynlighed for at have en mønt over sig. Genereringen af fjender sker separat for chunkene og bruger også tilfældighed. Spillet har en angivet sandsynlighed for genereringen af fjender per sekund, som sørger for at mængden af fjender er tilfældig. For at undgå, at bruge for mange ressourcer til fjenderne, vil de blive fjernet, når de bevæger sig ud af skærmens område.

Typen af platforme og fjender

Spillet bruger et type-system til generere platformene og fjenderne. Ud fra hvert billede, er der blevet lavet en tilhørende type. CLOUD-, BENCH- og LEAF-typen er eksempler på platforme typer. Ligeledes er der BIRD-, PLANE- og UFO-typen som fjender. Når der skal genereres platforme og fjender, så bliver objektets højdes brugt til at bedømme, hvilken type der skal bruges. For at skabe variation blandt platformene er programmet også lavet til at håndtere flere typer til hver chunk, hvor programmet til hver platform vælger en af typerne tilfældigt.

Spiller-inputs og ButtonManager-klassen

Til at holde styr på spillerens inputs har programmet en ButtonManager-klasse, som kan sammenkoble knapper med forskellige Bindings, og som kan via boolean-metoder kan fortælle, om en given Binding bliver trykket. Denne klasse bruges primært, fordi Processing kun kan gemme på den knap, der sidst er blevet trykket på, og det vil sige, at man med Processings keyPressed-metode ikke kan bruge flere knapper på en gang. ButtonManager bruger derfor et hashMap, der sammenkobler hver knap med en boolean, der angiver om en knap er trykket på eller ej. På den måde bliver det nemt at tjekke, om der eksempelvis bliver trykket på en af hoppe-knapperne.

Bevægelse og fysik

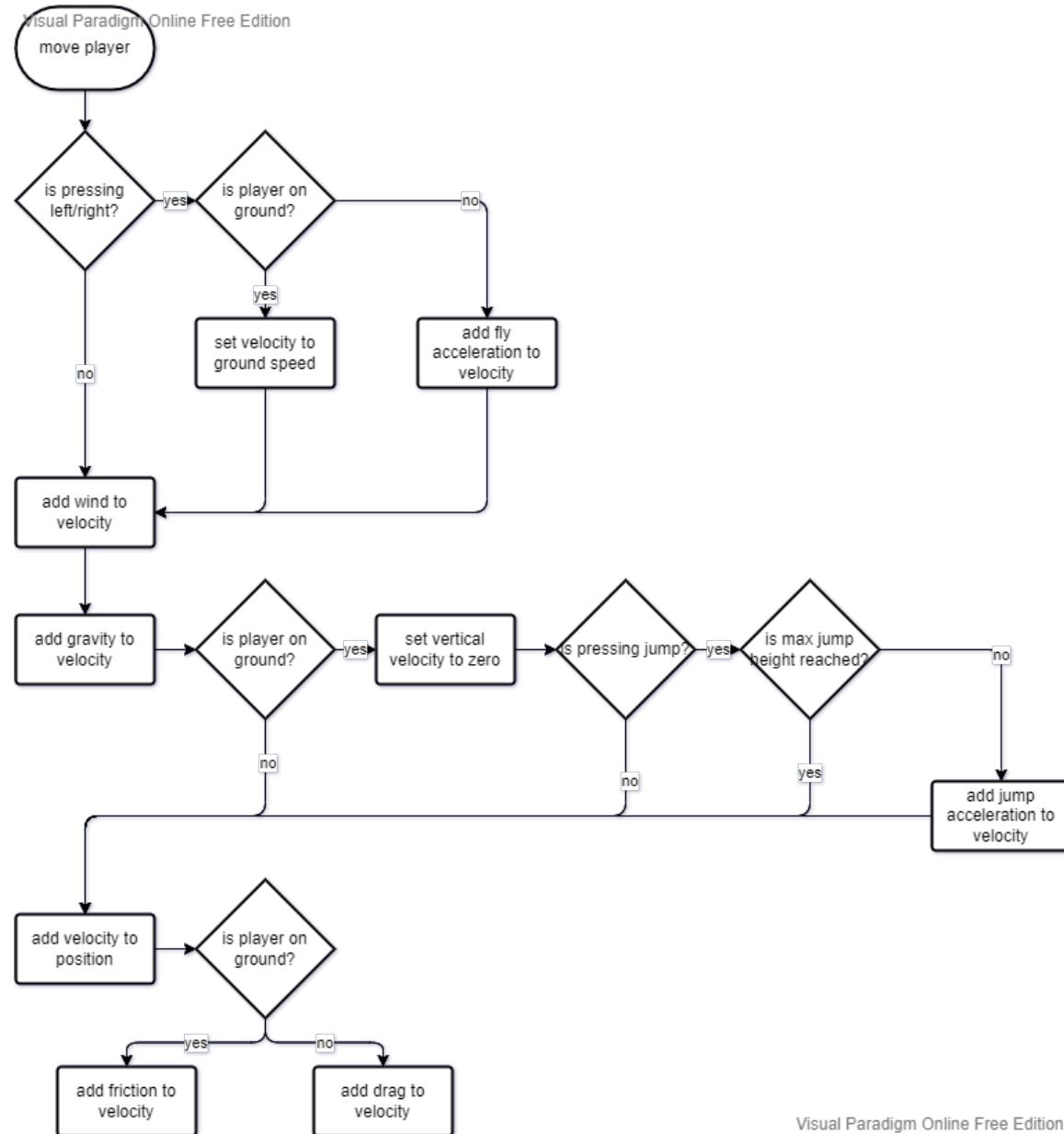
Bevægelse i spillet er baseret brugen af 2D-vektorer til at beskrive objekternes position og hastighed i både x- og y-aksen. I nogle tilfælde bliver der også brugt acceleration, og i andre bliver der brugt konstante hastigheder. Hvert frame vil programmet for hvert objekt udregne, hvor meget der skal lægges til eller trækkes fra objektets hastighed, og derefter vil hastigheden blive lagt til objektets position. Størrelsen på bevægelserne afhænger af, hvor lang tid spillet bruger på hvert frame. Det vil sige, at når spillet kører på en computer med høj FPS, vil alle bevægelserne ligeledes bliver mindre, sådan at alle objekternes bevægelser er den samme i forhold til tiden i virkeligheden. Derfor vil bevægelserne bliver større, når der er en lav FPS, så spillet ikke bevæger sig langsommere. Her bliver der derfor brugt en værdi, kaldet deltaTime, som er findes ved at finde reciproksen til Processings frameRate-variabel. deltaTime er altså tiden det tager at vise hvert frame.

Spilleren og Frog-klassen

Frøens bevægelse afhænger af spillerens input og verdenens platforme. Hvert frame vil spillerens hastighed bliver opdateret i forhold til ButtonManager-klassen, sådan at, hvis man

trykker mod højre eller venstre, bevæger man sig til siden, og, hvis man trykker på hop, bevæger man sig opad. Når man er på en platform, er højre-/venstre-bevægelsen baseret på en konstant hastighed. Når man er i luften bliver der brugt en acceleration, som så bliver lagt til hastigheden i den retning, som man trykker.

I forhold til at hoppe bliver der også brugt acceleration, men her er der blevet sat en tidsbegrænsning på, hvor længe et hop kan vare. Dette gør, at spilleren kan holde hoppe-knappen inde og hoppe højere, end hvis de kun havde trykket på knappen et kort øjeblik, uden at lade spilleren flyve uendeligt højt. I forhold til tungten, bruger programmet spillerens muse-position, til at beregne retningen. Den maksimale længde af tungten samt dens hastighed er blevet angivet som variabler.

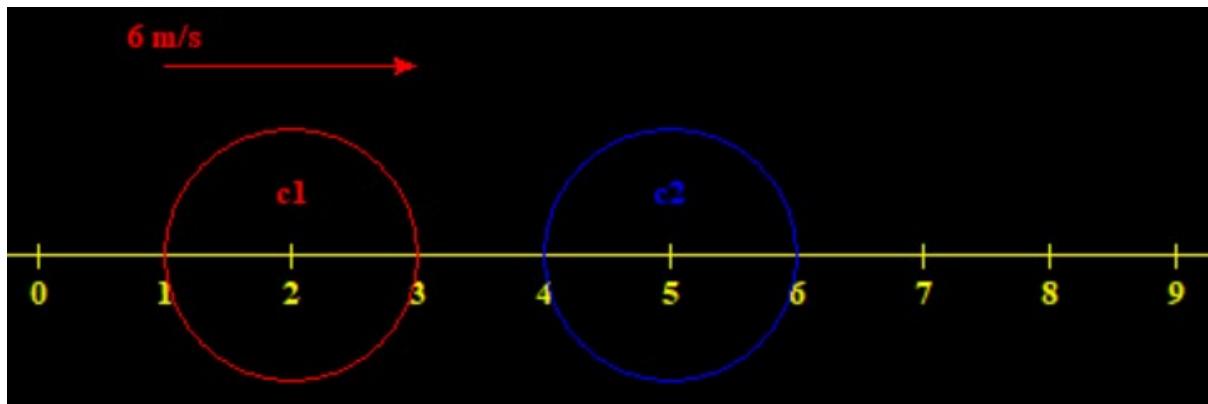


Visual Paradigm Online Free Edition

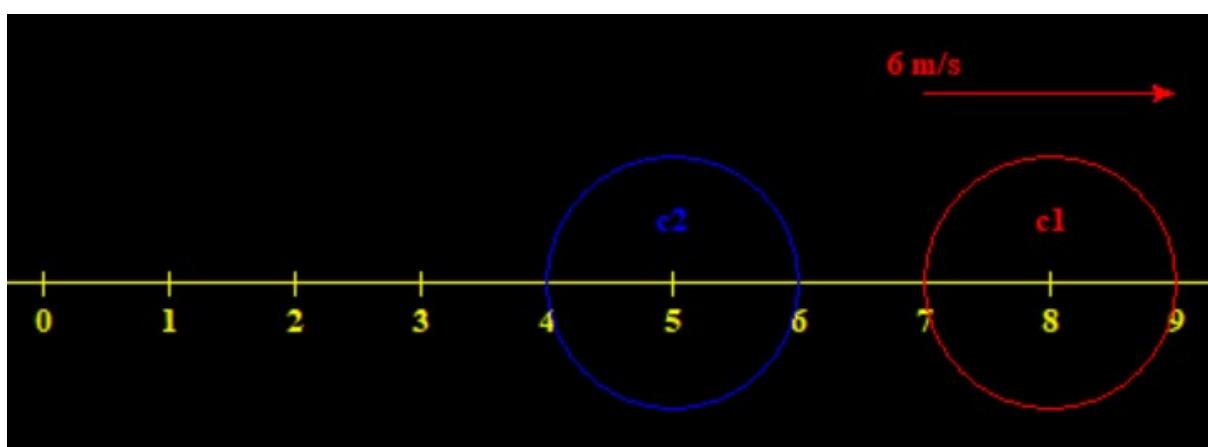
Kollision

Hver frame vil frøen også kontrollere, om den vil kollider med nogle af platformene. Denne kontrollering bliver ikke kørt, når frøen bevæger sig opad eller trykker på ned-knappen. Dette er for at tillade spilleren at bevæge sig gennem platformene, når man bevæger sig opad, men ikke nedad. Derudover tillader det også spilleren at falde ned gennem platformene via ned-knappen, hvis man vil.

Til at forbedre kollisions-kontrolle bliver der også brugt interpolation til at sikre at spilleren ikke kan bevæge sig gennem platformene ved høje hastigheder. Dette har nemlig været et problem, da spillet bliver påvirket af det, som i det henviste StackOverflow-opslag, bliver kaldt for tunnelling.⁶ Dette sker, når et objekts hastighed er så høj, at det i et frame kan befinde sig på den ene side af en mur og i det andet frame befinde sig på den anden side af muren, sådan at spillet ikke har mulighed til at registrere kollisionen. Derfor bliver der også taget højde for spillerens hastighed. Det vil sige, at platformens og frøens størrelse plus frøens hastighed skal være mindre end afstanden mellem frøen og platformen, for før at der bliver registreret en kollision.



(Figur 43: Frame 1 af eksempel på tunnelling)⁶



(Figur 44: Frame 2 af eksempel på tunnelling.)

```
private Platform getClosestCollidingPlatform(List<Platform> platforms) {
```

⁶ <https://stackoverflow.com/questions/61878376/linear-interpolation-and-object-collision>

```

return closePlatforms.stream()
    //Filter the platforms that share x-coordinate with the frog
    .filter(p -> Math.abs(p.position().x - position.x) < p.size().x / 2)
    //Map the platforms to a PlatformAndDistance object to keep track of the
    distance and object
    .map(p -> new PlatformAndDistance(p, p.position().y - position.y))
    //Filter the platforms whose distance is less than the frog's height plus
    its y-velocity (interpolation)
    .filter(pd -> pd.distance >= 0 && pd.distance <= velocity.y + size.y)
    //Get the closest platform
    .min((a,b) -> Float.compare(a.distance, b.distance))
    //If none is found, return null
    .orElse(null);
}

```

(Figur 45: Skærbillede af programmets kode for kollision.)

Tyngdekraft, vind og vindmodstand/friktion

Spillet bruger en Forces-klasse, som har et par statiske funktioner til at beregne tyngdekraften og vinden ved forskellige højder samt vindmodstanden/friktion. Disse værdier bliver lagt til de forskellige objekter hver frame for at simulere fysikken i verdenen. Værdierne er baseret på objektets højde, sådan at spillet eksempelvis kan have en lavere tyngdekraft, når man kommer op i rummet.

Camera-klassen

Til at holde styr på at kalde alle display-metoderne bruger programmet en statisk Camera-klasse. Camera-systemet fungerer på den måde, at alle objekters positioner angiver, hvor de er i spille-verdenens. Kameraet har derfor sin egen position, som bliver trukket fra alle objekternes positioner, når de skal tegnes. På den måde vil alle objekterne bliver bevæge sig rundt på skærmen i forhold til kameraet position. Dette tillader gruppen at angive objekternes positioner ud fra spille-verdenens koordinatsystem i stedet for skærmen/vinduets koordinater. Derudover er kameraet lavet til at følge spilleren. Hver frame vil klassen beregne afstanden mellem frøen og kameraet, og hvis afstanden er for stor, vil kameraet bevæge sig mod frøen.

Baggrunden

Kamera-systemet gør det også nemt at lave parallax-effekten, da kameraets position blot kan ganges eller divideres, sådan at baggrunds-elementerne henholdsvis kan bevæge sig hurtigere eller langsommere end de andre objekter. Derudover bliver den bagerste baggrund fyldt ud med en gradient, hvis farver vil afhænge af kameraets højde. Gradienten er lavet ved at tegne en masse horisontale linjer, som gradvist går mellem to farver.

SQLite database

Til at gemme spillerens score bruger programmet en SQLite database, som har en enkel tabel kaldet saves. Hver 15. sekund gemmer/opdaterer programmet spillerens højde og antal mønster i databasen. Derudover gemmer den også, hvorvidt spilleren er i live med denne score. Når spilleren åbner spillet vil programmet lede efter en score, hvor som er markeret som stadig i live. På den måde kan programmet gøre, så spilleren starter med denne score. Når spilleren på noget tidspunkt dør, bliver dette øjeblikkeligt opdateret i databasen, så spilleren ikke længere kan bruge denne score, hvis de åbner spillet igen.

IT-sikkerhedspolitik

Det er også vigtigt at vurdere ting såsom databeskyttelse og sikkerhed på internettet.

GDPR (Databeskyttelsesforordningen)

Da vores spil prototype ikke P.T. inkludere nogen form for data logging udover at gemme highscoren lokalt, så er GDPR ikke en af de store problemer. Dog kunne det tænkes at en færdig version ville inkludere et online scoreboard hvor spillere ville kunne sammenligne deres score med andre. For at kunne fremstille sådan et system skal der naturligvis gøres brug af en form for database der kunne gemme på spillerens score samt kontooplysninger. Yderligere kunne man også gøre brug af en API for en service såsom facebook for at gemme eller dele sine score med sine venner.

Ophavsret

Når spillet skulle laves ville gruppen gerne have at alle assets som skulle bruges var noget man selv havde lavet, og så gruppen ikke skulle tænke på ophavsret. Dette betyder for gruppen at man ikke skulle bekymre sig om hvilke assets man brugte siden de alle er blevet lavet af gruppens medlemmer. Hvis dette ikke var scenarie skulle gruppen finde side hvorfra gruppen skulle havde kunnet taget billede til spillet, og hvem man skulle inkudere eller betale for at skulle bruge disse billeder. Sider man kunne havde brugt hedder "shutterstock" som tager billede og udgiver dem som alle kan bruger som de lyster. Dog igen har gruppen lavet alle deres egne assets og derfor ikke har skulle tænke på ophavsret.

Brugertest

Efter som mange af elementerne fra spillet er færdig, så kan gruppen gå i gang med at lave en brugertest. Når man laver en brugertest, tester man spiller for at se om andre har den samme opfattelse af den ting du tester som personer der laver produktet har. Her er der vigtigste at spillet er så at se om spillet er så brugervenligt som gruppen mener det er. Ting som dette tester gruppen ved at have spørgsmål før og efter testen, for at sikre os at kan få feedback som kan bruges til optimere spillet, eller lave flere ting som kan gøre spiller sjovere.

Selv testen er blevet lavet ud fra en kvalitativt metode, gruppen valgte at lave et interview. Dette gør gruppen siden det er oplagt at interview brugerne og derefter bruge dem som testkaniner for spillet. Den kvalitativt metode er i dette scenerie langt bedre end den kvantitativ metode, da gruppen kan få brugerne til rent faktisk at prøve spillet ud i person. Gruppen kan guide personen til at spille spillet på den måde så gruppen kan se om det er brugervenligt nok, og så de opfylder at gruppen mener alle vil kunne spille det. Den kvalitative metode er også bedre siden man kan få mere specifikke svar på hvad der skal gøres bedre og hvad der fungere. Dette vil man ikke havde kunnet fået i et spørgeskema siden man ville have mere specifikke svar der.

Når man så har lavet interviewene og brugertesten, så kan man gå igennem svarene og se hvad der fungere og hvad der ikke gør. Har kan gruppen også se om de funktioner man har i spillet er lige til, og at man ikke skal lede efter dem. Gruppen tjekker også for om karakteren er nem at spille og om den føles rar at spille, som hvis den var for "tung" så skulle den nok hoppe højere, eller hvis der var en funktion som var svær at finde skulle man nok gøre noget for at gøre den mere mærkbar.

Feedback

Testen er blevet lavet og gruppen ville i alt teste 3-4 personer, dette gør man da gruppen vil have en bred spector af mennesker med forskellige viden inden for videospil. Gruppen ville for eksempel gerne have nogle som ikke spillede så meget og nogle som har spillet i flere år. Dette vil gøre at man får feedback fra forskellige mennesker som ligger under målgruppen i at man ikke behøver at spille så tit for at spille spillet.

Det feedback man har fået som er relevant for os, er den som foregår efter testen, dette er den data som kan bruges til at gøre spiloplevelsen bedre.

Til det første spørgsmål "*Hvordan føles spillets karakter at spille?*" fik man at vide at karakteren var samlet set, meget fornøjelige at styre. Den føltes, for nogen, "*ikke for tung*", "*kontrol mæssigt var den meget god*", og "*den hoppe godt*". Dog for andre var karakteren lidt langsom da den gik rundt på jorden og en "tunge feature" var svær at finde. Dette kan man bruge til at bare hurtigt gøre frøen en tand hurtiger og finde en måde hvorpå "tunge funktionen" er nemmer at se. dette kunne man for eksempel gøre med et sigte på musen, siden man trykker for at skyde med tungen. Disse få ting ville ikke være særligt svære at optimere.

Det andet spørgsmål til efter testen var "*Hvordan var sværhedsgraden?*", dette spørg gruppen siden spillet skulle kunne spilles af en bred gruppe af mennesker. Her fik gruppen

meget de samme svar “*ikke særligt svært, siden man ikke kan dø*” eller “*ikke særligt svært, siden der er for mange platforme*”. I det stadie hvor spillet var havde man ikke færdiggjort funktion i at kunne dø, men ud over det kan gruppen konkludere at spillet var nemt at spillet, som lige præcis var gruppen gik efter, men dog kunne gøres en smule svære.

Tredje spørgsmål lyder sådan her “*Er spillet sjovt?*”, her vil man egentlig bare have feedback for om nogen overhoved ville spille spillet, og hvis nej, hvorfor ik. Testpersonerne svarede alle sammen ja, men kom med en kommentar som kunne gøre det sjovere. Her fik gruppen “*ja, til en vis grad*” og “*ja, man måske ikke i så lang tid, det er en time killer*”. Spillet er sjovt men måske kunne det have nogle features som kunne gøre det sjover, siden det er meget det sammen, som sagt en “*time killer*”. Det er godt feedback at få, man kan bruge det “men” til at tænke på flere ting for at give brugerden det sjovere spil som det nok ville spille i længere tid.

Til det fjerde spørgsmål spørger gruppen brugerden “*Minder spillet dig om nogle andre spil, og ja hvilke er bedre?*”, i dette spørgsmål vil gruppen høre om hvordan man eventuelt kunne konkurrerer med andre spil og hvordan man potentielt kunne gøres bedre. Her fik gruppen kun et relevant svar som “*color switch, det er bedre fordi man kan tage en pause*” man kunne have samme funktion i spil dog går det hele ud på hvor højt du kan nå, og hvis man kunne pause spillet ville det ikke have sammen effekt. Gruppen fik også at vide at spillet mindede om “*doodle jump*” som var en af de spil som gruppen brugte til reference. Det var godt at vide de mindede om hinanden, dog fik man ikke at vide om gruppens var bedre. Feedbacken her kan bruges til at gøres spillet bedre end de spil som gruppen konkurrerer med, og desværre fik gruppen ikke noget feedback, i dette spørgsmål, som var så hjælpsomt.

Til sidst ville gruppen spørge om testpersonerne havde noget yderlige feedback, som ikke blev spurgt om i det tidligere spørgsmål. Her fik man at vide hvordan de mente spillet kunne gøres bedre, alle brugerden som med gode eksempler. Som at man måske skulle bruger mønsterne til at købe “*accessories*” eller andre “*skins*”, som kunne motivere spilleren til at spille videre og i længere tid. Et andet feedback som gå under kategorien “*bruger længere tid i spillet*” var ved at man kunne lave forskellige baner, så man ikke hele tiden spille den sammenbane.

Den samme person kom også med at forslag til en ny funktion, som var at nogle forskellige platforme, som keglen, kunne droppe efter man har stået på den i et stykke tid. Dette vil gøre at spilleren skal tænke over hvor de hoppe og hvor lang tid de står på forskellige platforme, dette ville også øge sværhedsgraden en smule.

Til sidst fik gruppen feedback til karakteren og dens kontrolls. Her fik man at vide at kamera kontrollerne var forvirrende og mange troede man skulle hoppe på “*w*” da man skal gøre det på “*space*”. Dette kan kan hurtigt ændres ved at man vise spil kontrollerne så spilleren ikke forvirres.

Konklusion på feedback

Der kan konkluderes at størstedelen af feedbacken som gruppen fik af brugertesten kan bruges til at endvidere forbedre programmet, og som også kan gøre spillet bedre og endnu mere brugervenligt for målgruppen. De største og hyppigste klager for brugertesten var:

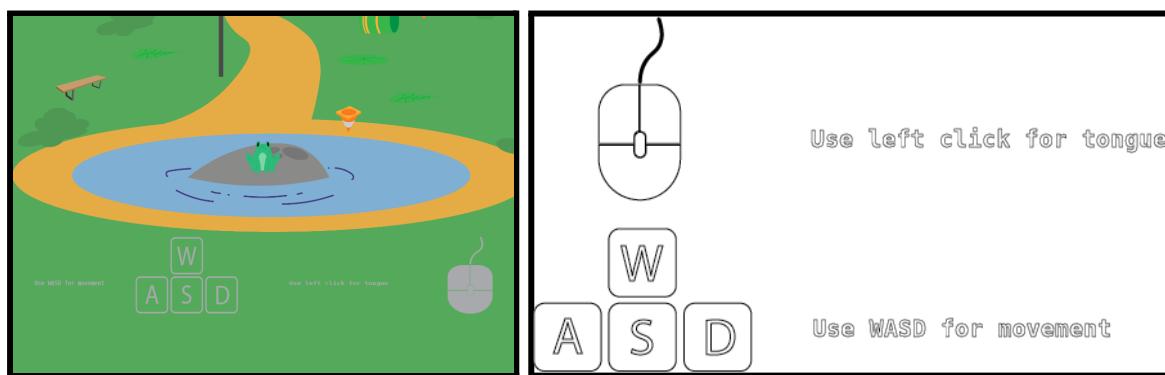
1. Vise spillerens kontroller (hvordan man kan bevæge, hoppe, skyde tungé, osv.).
2. Sørge for at karakteren kan dø.
3. Gøre spillet sværere (færre platforme der kan risikere at falde, stærkere vind).
4. Flere ting at kunne gøre i spillet (flere baner, butik til at bruge samleobjekter).

Column1	Spørgsmål efter test	Testperson 1	Testperson 2	Testperson 3
1	Hvordan føles spillets karakter at spille?	Karakteren er ret nice, at hoppe er lige som man forventer	Kontrols messigt er det meget godt. Man er måske lidt langsom når man går på jorden. Vægten af karakteren er god	Han hopper meget højt. Højre venstre er meget fint. Ved ikke helt med tungefunktionen (vidste ikke at den var der)
2	Hvordan var sværhedsgraden?	Det var ikke særligt svært da man ikke kan dø endnu	Det ville være godt hvis man faktisk kunne dø	Vireker ikke særligt svært fordi der er mange platforme. Måske sterkere tidliger
3	Er spillet sjovt?	ja til en hvis grad. Det har potentiale	Jo det er meget sjovt. Tunge funktioner er meget intressant og forventer at man kan lave mange sjove ting ud af den. Grafikken er også god og har sin egen stil selvom de er simple	Altså ja. Men måske ikke i så lang tid. Det er lidt af en "timekiller"
4	Minder spillet dig om nogle andre spil og er det bedre end de spil du tænker på?	Ja, color switch. Det er bedre fordi man kan tage en pause	Doodlejump. Men det er også noget der før i tiden var meget in.	Minder om "happy pudding"
5	Andet	Kamarets controls er forvirrende. Trode hop var på "w". Dræb fjender med tungé. Skyerne er i rummet	Lav nogle platforme der dropper når man står på dem. Det kunne være fedt hvis de var designet baner (en mini campaing)	Gør så du kan købe noget med mønsterne så man får mere motivation såsom andre frør eller andre baner.

(Figur 46: Resultater fra brugertesten indsat i en tabel.)

Forbedring af program ift. brugertest

Generelt feedback fra brugertesten var at man ikke helt vidste hvad spillets controls var. På baggrund af dette problem er en række pictogrammer blevet lavet for bedre at kunne forklare hvad spillets controls var.



(Figur 47: Forbedring af program med feedback fra brugertesten.)

Nogle af testpersoner kom også med forslag til ekstra indhold og features, som kunne være gode at tilføje, men som på grund af gruppens begrænsede tid og ressourcer ikke kan tilføjes.

Evaluering

Opgavekrav

Her vurderes det hvorvidt opgavens krav er blevet opfyldt af projektet. Dette kan ses som en god indikator for hvordan projektet er gået. De omtalte krav ses forneden:

Der skal anvendes en database eller anden form for datasikring.

Til programmet for eksamsprojektet bliver der anvendt en SQLite database til at kunne håndtere brugernes score, og holde styr på hvordan man kan se dem rangeret. Gruppen har overvejet datasikring og gemt filen i en sikker mappe, hvor man ikke kan lige tilgå den hvis man er uerfaren med hacking.

Selve produktet skal overvejende være udviklet med et højniveau sprog, vi anbefaler Processing, C#, Java, Python eller C++. Vi forbeholder os retten til at godkende udviklingsmiljø og sprog, hvilket derfor skal være tydeligt angivet i jeres projektbeskrivelse.

Kodesproget for programmet er udviklet i Java/Processing, og er blevet godkendt af begge vejledere. Det er også nævnt i "tekniske valg" i rapporten, at der er blevet gjort overvejelse af kodesprog.

Vi forventer et højt niveau inden for både valg og udførsel af projektstyring igennem hele projektet, herunder en særlig kritisk indstilling til tidshorisonter (for optimistiske/for pessimistiske) og projektomfang.

Projektstyringen er gået generelt meget godt med undtagelse af den sidste iteration der omhandlede en grafisk forbedring af programmet som gruppen ikke kunne nå at lave. Den grafiske forbedring var ment som at tilføje flere elementer til spillet og andre måder at spille spillet på, men var ikke vigtig for at kunne have en klar prototype, og af den grund var det heller ikke den vigtigste del af projektet.

Der skal udvikles en prototype. Prototypen skal testes og danne grundlag for den endelige udvikling af produktet.

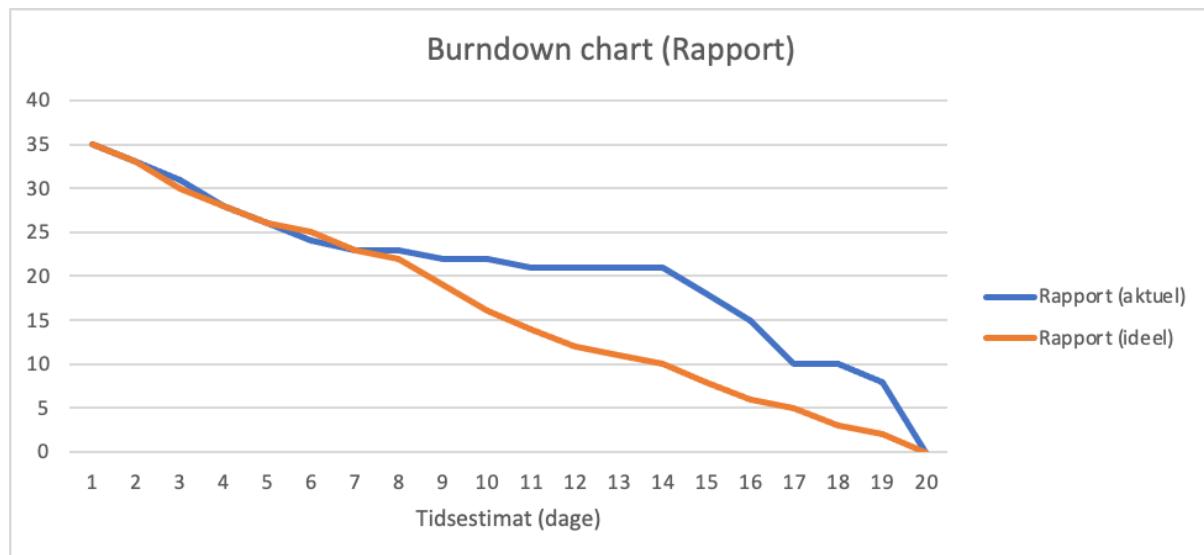
Der er blevet udviklet en færdig klar prototype som også er blevet testet og revideret på baggrund af brugertests og brugerundersøgelser, samt adskillige test resultater. Der er testet for om programmet opfylder alle krav for eksamsprojektet.

Desuden skal det overvejes hvordan I håndterer: sikkerhed, GDPR, interaktionsdesign og brugertests.

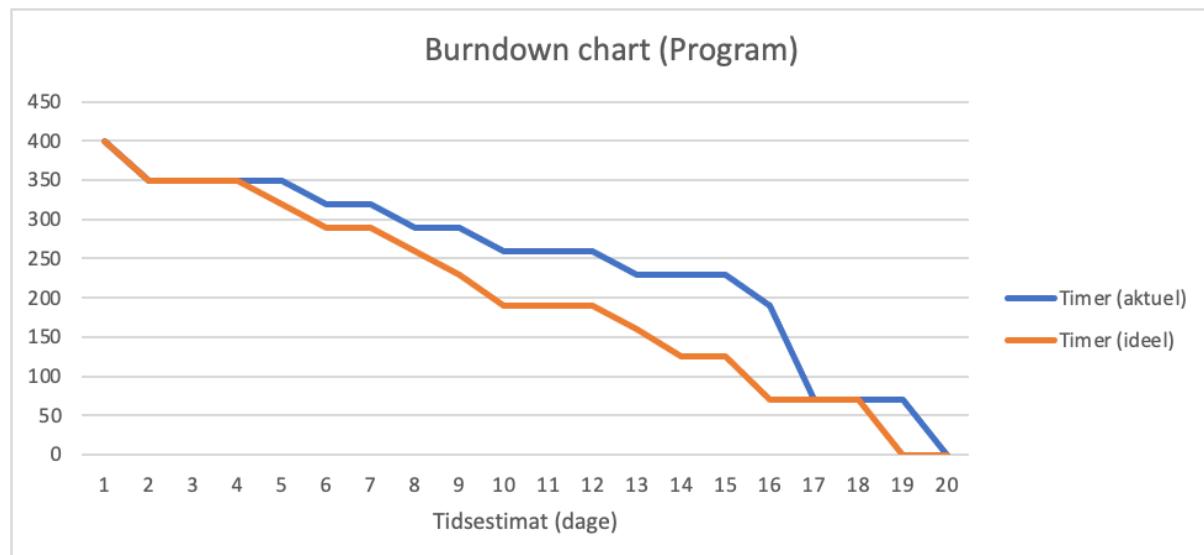
GDPR er blevet stærkt overvejet, men fylder ikke en særlig relevans for prototypen da meget er privat og ikke taget fra offentligheden. Brugertest har været essentielle for programmet, da det hjalp med at forbedre programmet og fikse fejl.

Burndown chart

Som man kan se på gruppens to burndown charts nåede gruppen både iterationer og sideantal. Man kan se at gruppen har haft være en smule bagud ift. iterationer og sideantal, som kan i forskellen på den blå og orange graf på begge burndown charts. Hvis man sammenligner graferne, har gruppen været rimelig dårlig til at skrive oftere på rapporten, hvor man har haft skulle skrive længere og mere de sidste på dage, som ikke er ideelt for et eksamensprojekt.



Til programmet har brundownchartet ligget meget tættere, dog en smule over grafen for den ideelle tids estimation. Programmet blev dog færdigt til tiden.



(Figur 48 og 49: Burndown chart for både programmet og rapporten. Disse charts er dem der er blevet foretaget til sidst i projektet.)

Konklusion

Gruppen startede projektet med at finde en løsning til problemstillingen

"Hvordan kan man lave et computerspil der kan sørge for at spilleren er både aktiv, tænker strategisk, og får spilleren til at planlægge sit næste træk, men samtidigt også er begyndervenligt og tilgængeligt for alle i aldersgruppen fra 6 årlige til og med 86 årlige?"

Gruppen har analyseret sin målgruppe ved at undersøge spillertyper og personae. Man har formuleret adskillige User Stories, tasks og iterationer for programmet til at kunne designe det bedste program ift. målgruppen. Gruppen fik derefter evalueret sin arbejdsproces og program, og fandt ud af nye metoder til at forbedre på begge ting.

Til slut kan gruppen konkludere at gruppen succesfuldt har lavet et fuldendt projekt, udformet i et spil, der opfylder en blanding af case 1 og 2. Gruppen ville gerne have at spillet var brugervenligt og kunne spilles af alle, dette har gruppen opnået ved brug af feedback fra brugertesten.

Gruppen har ultimativt lavet et spil, hvor hvert gruppemedlem har givet deres bedste for at kunne lave et godt DDU eksamsprojekt. Dette er bl.a. gjort muligt med god projektstyring som har holdt gruppen i gang konstant.

Logbog

En vigtig ting at holde styr på for at bl.a holde styr på arbejdsprocessen og kunne opdatere ens burndown charts er ved at lave logbøger over sin proces. En logbog er meget god at bruge for også at optimere arbejdsprocessen, f.eks. hvad har man arbejdet på, hvad skal man arbejde videre på, og endvidere om man kan forudse hvordan arbejdsprocessen for det specifikke emne vil ende.

Tirsdag d. 22. februar

I dag er starten af vores projekt. Vi er lige begyndt og skal have udvalgt vores case til projektet. Vi skal også i gang med vores projektbeskrivelse da den skal afleveres snart. Vi har valgt at tage en blanding af case 1 og 2, og lave vores eget spiludviklingsprojekt. Vi er godt i gang med projektbeskrivelsen og har lavet mange tekniske valg.

Onsdag d. 23. februar

Dagen efter vi begyndte fik vi lavet meget af projektbeskrivelsen. Vi fik lavet krav- og design-specifikation til projektbeskrivelsen og vi havde lavet en brainstorm for at finde på et computerspil at lave, og endte op med at vi ville programmere et 2D platform spil. Vi er næsten klar til at aflevere projektbeskrivelsen, men vi mangler at lave de endelige skitser og forklare hvordan vores program bruger kravene for projektet.

Fredag d. 25. Februar

Vi fik afleveret vores projektbeskrivelse i går, og er gået i gang med vores projektstyring aflevering. Dog har alle i gruppen også et programmeringsprojekt som vi snart skal have afleveret (27 februar), så vi fik ikke fået at lave meget i dag udover vores user stories og nogle af vores tasks.

Onsdag d. 2 marts

Vi har arbejdet videre på vores projektstyring og er ikke under nær så meget stress som da vi ikke var færdige med vores programmeringsprojekt. Vi er kommet godt i gang igen, men mangler stadig at lave en del. Vi har planlagt at lave resten som vi ikke når i dag i hverdagen og/eller i weekenden så vi ikke stresser ud over vores SOP (Studieområde Projekt).

Tirsdag d. 22 marts

Vi er endelige færdige med vores SOP, den var særlig stressende og svær at arbejde med. Vi blev færdige med vores projektstyring aflevering, og er fuldt ud begyndt på at arbejde på vores program og resten af vores rapport. Vi har en masse elevtimer sat til vores program, men også til rapporten da vi skal have skrevet mindst 30 normalsider.

Onsdag d. 23 marts

Vi har ikke været så fokuseret i dag, så vi har ikke fået særligt meget. Vi er begyndt på at lave en prototype af vores spil, men det er stadig i sine tidlige stadier, så vi kan ikke lige bedømme vores spil endnu. Vi har dog Marcus som er virkelig god til at kode, så det burde ikke være noget problem. Vi er blevet fortalt om vejledninger i næste uge, så vi skal forberede os på det.

Onsdag d. 30 marts

Vi skal først fremlægge på fredag, så vi har stadig tid til at kigge på vores rapport og arbejde videre uden at bekymre os. Marcus har optimeret prototypen, og Julie har fået lavet nogle gode digitale assets til spillets karakterer. Rapporten er også godt i gang, men vi mangler stadig 25 normalsider for at ramme mindstekravet.

Fredag d. 1 april

Vi har haft vores vejledningsmøde i dag og har opdaget at vores iterationers arbejdstimer ikke er helt korrekte, så vi skal have arbejdet på at ændre det. Vi blev også foreslægt at vi skal have to forskellige burndown charts - en til programmet, og en til rapporten. Vi indså også at vi er lidt bagud med begge, så vi planlagde en strategi om at inddelte gruppen for at bedre kunne håndtere de forskellige dele af projektet.

Tirsdag d. 5 april

Vi er godt i gang med vores program/rapport. Marcus sidder og koger sin hjerne over hvilke funktioner og linjer kode der skal enten tilføjes, optimeres eller ændres på. Julie er godt i gang med digitale assets såsom diverse baggrunds- og forgrundsdesign for spillet og dertilhørende farvepaletter. Jeg er selv igang med rapporten, og prøver at inkludere hvad der er nødvendigt at have. Alexander er desværre syg, så han arbejder hjemmefra også på rapporten.

Tirsdag d. 19 april

Vi er langsomt begyndt igen efter påskeferien. Mange af gruppemedlemmerne er lidt trætte efter en hel uge med påskefrokoster og påskeæg, så man fik ikke lavet særligt meget denne gang. Vi er på tidspunktet bagud med vores ideelle tidsestimater, men forventer at arbejde 110% og mere mod den sidste uge for at færdiggøre projektet.

Onsdag d. 20 april

Et af gruppemedlemmerne er fraværende, men vi har skrevet beskeder til hinanden online og har planlagt at arbejde på sin del af projektet. Resten af gruppen er på skolen og arbejder videre på programmet og rapporten. Programmet skal arbejdes mere på så det er klart til at vises til det næste vejledningsmøde.

Onsdag d. 27 april

Gruppen har fortsat arbejdet videre på hver deres del af projektet, men man kan godt føle at man arbejder mindre effektivt end som i sidste uge. Grunden til det er fordi der snart skal afleveres Kommunikation/IT eksamensprojekt, så gruppen har pres på to afleveringer.

Fredag d. 29 april

Et af gruppemedlemmerne var allerede hjemme pga. sygdom, og en anden skulle tage hjem pga. problemer med sin bærbare computer, så der var ikke arbejdet så meget som der originalt skulle have været.

Tirsdag d. 3 maj

Dagen efter aflevering af KIT eksamensprojekt. Mange af gruppemedlemmerne er lidt trætte fra at være sent oppe forleden nat, og som et resultat af dette har størstedelen af gruppen ikke særligt meget energi. Ved at bedømme de andre logbøger og hvor langt gruppen er ift. deres burndown chart, og har vurderet at raten af effektivitet er ikke høj nok inden deadline. Der skal derfor arbejdes en del mere koncentreret i den kommende teknik uge.

Onsdag d. 4 maj

Gruppen har arbejdet særligt godt på programmet, mange af spillets mekanismer er nu blevet kodet og er spilbare. Der er stadig nogle iterationer/tasks der mangler ift. programmet, og rapporten mangler stadig en masse normalsider, men gruppen er selvsikre i at de vil nå det inden deadline i næste uge.

Torsdag d. 5 maj

Programmets assets såsom hovedkarakteren (frøen) og de fastlåste objekter (platformene) er nu blevet implementeret og giver spillet en bedre følelse når man spiller det. Yderligere er frøens våben, som blev besluttet skulle være en tunge fra en frø, nu blevet implementeret i spillet så man kan opsamle samleobjekter i spillet og have et formål med spillet. Eftersom at vi har været bagud med programmet, så har vi som slut på første iteration været nødt til at lave nogle ændringer på iterations-listen, for at sikre, at vi får lavet taskene på de rigtige tidspunkter.

Fredag d. 6 maj

Gruppen har i dag holdt deres andet statusmøde som alt i alt gik godt. Dog skal gruppen gøre mere ud af deres projektstyring med fokus på iterationerne. På baggrund af den feedback har gruppen optimeret deres Trello så den har et større fokus på de enkelte iterationer. Yderligere har gruppen også fortaget en brugertest af deres demo som alt i alt blev modtaget positivt. Dog var der forvirring over spillets "controls" med hvilken knapper der henholdsvis gjorde hvad. Yderligere blev nogle bugs identificeret såsom f.eks. at platformene kan generere udenfor frøens rækkevidde, dvs. at frøen/spilleren ikke kunne hoppe op på platformen da den blevet genereret i en højde frøen ikke kunne hoppe til. Mødet gjorde os desuden også opmærksomme på at vores tasks ikke var i helt overensstemmelse med indhold og tidsestimering. På baggrund af det reviderede vi vores tasks og tidsestimater i forhold til iterationerne.

Mandag d. 9 maj

Mange af gruppemedlemmerne blev hjemme i dag da der ikke var nogen speciel grund til at møde op, og fordi to havde nogle vigtige planer de skulle nå. Der blev derfor foretaget et Discord-opkald, hvor i gruppen diskuterede de næste trin for programmet og rapporten, såsom hvilke fjender der skal implementeres, hvordan det går med projektstyringen, og hvad der skal arbejdes på som det næste. Eftersom database-tasken har taget længere end forventet, så er den blevet rykket ned til iteration 4.

Tirsdag d. 10 maj

To af gruppemedlemmerne er gået kolde, og var ikke til stede i dag til den anden brugertest. Brugertesten afslørede at der skulle arbejdes mere på alt med programmet. Enten var noget ikke spændende, andre gange for spændende. Derudover arbejdede gruppen videre på

rapporten og programmet. Deadlinen er mindre end en uge til, så gruppen giver det sidste skub og arbejder hårdt.

Onsdag d. 11 maj

Sidste dag med timer. Gruppen arbejder særligt hårdt, men ved at der skal arbejdes uden for skoletimerne hvis man skal kunne nå at færdiggøre rapporten og programmet. Gruppemedlemmerne er lidt nervøse, men prøver at være rolige og arbejde fokuseret.

Litteraturliste

1. Scott Rogers - "Level Up!"
"Level Up!", <https://luisnavarrete.com/design/pdf/Level.pdf>
2. Somnium. "Bartles Spillertyper I Rollespil – et Tankeeksperiment – Somnium's Blog." Somnium's Blog, www.somnium.dk/blog/rollespil-bartles-spillertyper/
3. Chapman, Cameron. "Use Your Inspiration – a Guide to Mood Boards." Toptal Design Blog, 16 februar 2021:
www.toptal.com/designers/visual-identity/guide-to-mood-boards
4. Janaki Kumar, et al. "Bartle's Player Types for Gamification." The Interaction Design Foundation, UX courses, 6 Jan. 2020:
www.interaction-design.org/literature/article/bartle-s-player-types-for-gamification
5. "Doodle Jump." Wikipedia, 16 Jan. 2021:
https://en.wikipedia.org/wiki/Doodle_Jump
6. Justin. "C++ - Linear Interpolation and Object Collision." Stack Overflow:
<https://stackoverflow.com/questions/61878376/linear-interpolation-and-object-collision>
7. HubSpot. "Make My Persona." Hubspot.com, 2019:
www.hubspot.com/make-my-persona