

P0 Medialogi September 2011

Gruppe A306

Lav et spil i Game Maker



Nugget's Revenge

Indholdsfortegnelse

Synopsis	3
Design & Analyse.....	5
Krav og begrænsninger.....	5
Brainstorming.....	5
Tema	5
Målgruppe.....	6
Implementering	6
Historien i spillet	6
En 2D-platformer	6
Overgang mellem banerne og perspektiv-skift.....	12
Feedback og justering af sværhedsgrad.....	13
Rapid prototyping og placeholder-grafik	14
Workflow i Game Maker	15
Musik.....	15
Evaluering.....	17
Konklusion	17
Værktøjer	18
Referencer	19



Billede 1.0 Første bane i spillet *Nugget's Revenge*. Her styrer man kyllingen Nugget, der skal forsøge at fange den ondsksfulde bondemand

Synopsis

Denne rapport beskriver tilblivelsen af spillet *Nugget's Revenge*, som er udviklet i Game Maker af seks Medialogi-studerende fra Aalborg Universitet. Rapporten fortæller om tankerne bag spillets grundlæggende design, og hvordan spillet blev implementeret. *Nugget's Revenge* er et 2D platform-spil om en kylling ved navn Nugget, hvis familie bliver bortført af en ond bondemand. Det er Nuggets opgave at klare sig igennem tre baner for til sidst at kæmpe mod bondemanden og redde sin hønse-familie. Rapporten beskriver, hvordan spillets mekanik fungerer, og hvordan gameplayet, det visuelle og det auditive spiller sammen. I rapportens konkluderende afsnit kigges der på, hvad der kunne være gjort anderledes, og hvilke ting, der er blevet skåret væk fra spillet.

Gruppens underskrifter:

Gustav Dahl

Henrik Nørbygaard

Maja Kathrine Lundholm Larsen

Nadia Adolphsen

Peter Bøgh Andreasen

Tobias Winkel

Design & Analyse

I dette afsnit beskrives overvejelserne og processen, fra idé til det færdige spil *Nugget's Revenge*.

Krav og begrænsninger

På forhånd blev der opstillet et krav til spillet om, at det skal indeholde følgende elementer:

- Farmer
- 4 slags dyr: kvæg, grise, får og kyllinger
- Fjender af eget valg

Brainstorming

Det allerførste vi gjorde var at brainstorme os frem til mulige idéer til spillet. Alle kom med deres forslag, og derefter blev idéerne skrevet op på tavlen for at skabe et overblik over dem. Der var mange idéer - nogle mere realistiske at udføre end andre - men efter et par timer kom vi frem til to overordnede idéer.

Idé 1: Et *casual*-spil i stil med *Plants vs. Zombies* (Popcap, 2009)

Plants vs. Zombies er et såkaldt *tower defense*-spil, hvor man i stil med skak skal opstille forskellige enheder for at beskytte sin hjembase. Vi overvejede muligheden for at bruge *Plants vs. Zombies'* formular med lettilgængeligt gameplay og sjove figurer. Her skulle zombierne blot skiftes ud med dyr fra en bondegård og *power-ups* som en mejetærsker.

Idé 2: Et platform-spil i stil med *Super Mario Bros.* (Nintendo, 1985)

Den anden idé var et platform-spil, hvor det handler om at gå fra venstre del af skærmen mod højre, imens at man springer over diverse forhindringer og fjender. Spillet skulle omhandle en kylling ved navn Nugget, hvis familie blev kidnappet af en ond bondemand.

Siden ingen af de to idéer var synderlig originale i sig selv, valgte vi et lidt mere unikt koncept, hvor vi kombinerede de to: et platform-spil med gakkede figurer og onde zombie-dyr, samt action i form af skydevåben. I stedet for at bruge bondemanden som spillets hovedperson valgte vi at anskue det fra en anden vinkel og gøre kyllingen til protagonisten og bondemanden til antagonist. De resterende dyr blev valgt som spillets fjender, dog med et lettere modificeret "zombie-tilgang", hvor dyrene er blevet hjernevasket og genmodificeret af den onde bondemand.

Tema

Fra starten blev gruppen enige om, at spillet skulle have en lettere tegneserie-agtig stil, men at det samtidig gradvist skulle udvikle sig til et mere dystert univers, desto længere man nærmer sig bondemandens hemmelige militærbase. Der ses altså en udvikling fra det idylliske bondegårds-miljø til de mere skumle undergrunds-miljøer, hvor den onde bondemand har til huse.

Målgruppe

Formålet med spillet var at lave en platform-titel, som er nem at samle op, men også tilbyder en vis grad af udfordring i stil med gamle arkadespil fra 1980'erne. Gameplayet er letforståeligt, og man styrer hovedfiguren ved hjælp af fire knapper på tastaturet, hvilket gør det nemt for alle at spille. Spillet henvender sig mest til den yngre målgruppe, men burde sagtens kunne spilles af alle, uanset tidligere erfaringer med spil-mediet. Spillet giver dog associationer til klassikere som *Super Mario Bros.* (Nintendo, 1985) og *Mega Man* (Capcom, 1987), hvilket giver ældre spillere en nostalgisk følelse.

Hvis spillet skulle udgives kommercielt, ville distributionskanalen sandsynligvis være spil-portaler på internettet, hvor forbrugeren enten gratis eller for et beskedent beløb kunne prøve spillet.

Implementering

Historien i spillet

Når spillet startes op, mødes man af en startmenu og en knap med teksten "Play". Herefter begynder en kort intro-sekvens bestående af tre billeder holdt i en tegneserie-stil (billede 1.1). Her fremføres spillets historie om en ond bondemand, der fanger hovedpersonen Nuggets familie. Nugget undslipper bondemanden og det er nu op til ham at redde sin hønsefamilie. For at skabe en kontrast i Nuggets figur fremstilles kyllingen sød og uskyldig, men samtidig går han rundt med et stort maskingevær og rødt bandana-tørklæde om hovedet.



Billede 1.1 Intro-sekvens til *Nugget's Revenge*.

En 2D-platformer

Da Game Maker hovedsagligt er en *game engine* brugt til at lave spil i to dimensioner, valgte vi at lave et 2D-spil i genren *platformer*. Denne genre kan spores helt tilbage til de første video- og arkadespil som *Super Mario Bros.* (Nintendo Entertainment System, 1985) og *Sonic The Hedgehog* (Sega Mega Drive, 1991). Der er mange fordele ved at lave spil i denne genre, både for spilleren og spiludvikleren. Først og fremmest er 2D-platformers en kendt genre, som mange mennesker har prøvet før. Platform-spil

handler om at bevæge en figur rundt på skærmen og hoppe over diverse forhindringer. Platform-spil er nemme at gå til og kræver kun meget lidt tilvænning for at forstå spillets mekanikker. Visuelt er 2D-spil også nemmere at overskue i forhold til avancerede 3D-computerspil.

Derudover er platform-spil forholdsvis lette at udvikle, især med et værktøj som Game Maker, der gør det nemt at arbejde med 2D-elementer. Selve mekanikken og programmeringen bag er enkel at implementere, hvilket giver mulighed for at fokusere mere på banedesign, forhindringer og fjendernes opførsel.

Spillets perspektiv er holdt i et klassisk *side-scroller* perspektiv, hvor man ser spillet fra siden. Målet er at bevæge sig mod højre, og kameraet vil gradvist bevæge sig ("scrolle"), og følge Nugget, så han aldrig kommer uden for skærmens synsfelt.

Nugget styres med piltasterne: Højre og venstre piltast bevæger Nugget frem/tilbage, og op-tasten bruges til at hoppe. Denne knap er valgt, således at mellemrums-tasten er fri til de senere baner, hvor Nugget får et skydevåben. Nugget har fra starten tre liv, og han mister et liv, hver gang han støder ind i en fjende, et projektil eller anden fjendtlig forhindring.

Grafikken er holdt i lyse og klare farver for at give spillet et tegneserie-look. Dette fremstår en smule kontrapunktisk set i forhold til historien om bondemanden, der lemlæster og hjernevasker sine bondegårdsdyr. I takt med at spillet skrider frem, bliver det visuelle såvel som det auditive mere dystert.

Spillets baner

Nugget's Revenge indeholder tre baner, samt en *boss-bane* til sidst, hvor den endelige kamp står mellem Nugget og den onde bondemand. Alle banerne har et særskilt udseende/tema og har til formål gradvist at lære spilleren, hvordan spillet fungerer. Den samlede estimerede spilletid er omkring 6 minutter.

Bane 1

Den første bane fungerer som en lille *tutorial*: en bane der skal belære spilleren. Først og fremmest bliver styringen introduceret med en lille tekst, der står på græsset, som fortæller, at piltasterne styrer Nugget. Derudover bliver *settingen* sat, altså hvor spillet foregår, og hvad spillerens mål er. Nugget befinder sig udenfor i idylliske bondegårds-omgivelser med stor åben himmel, sol og skyer. Lystig banjo-musik spiller i baggrunden, og tilsyneladende ånder alt ro og fred. Straks efter at spillet begynder, ser man den onde bondemand bevæge sig mod højre. Dette skaber incitament for spilleren til at følge efter ved at trykke på højre piltast. Dette er spillerens eneste mulighed, for det er ikke muligt at gå mod venstre, da man her støder ind i en usynlig væg.

Idet spilleren bevæger sig mod højre, bevæger skærmen med, og man mødes af nogle høstakker, som er spillets første forhindringer. Her lærer man, at man skal bruge op-tasten til at hoppe over stakkerne, og kort efter bliver man mødt af spillets første fjender: får med motorsave (se billede 1.2). Fårene er

placeret således, at de er utrolig nemme at hoppe over, men jo længere man nærmer sig, desto højere afspilles en farlig motorsavs-lyd. Denne feedback er vigtig for at indikere, at spilleren skal forsøge at undgå fårene. Banen slutter kort efter, hvor man mødes af en lade, som Nugget skal begive sig ind i for at komme videre til næste bane.



Billede 1.2 Starten af første bane i *Nugget's Revenge*. Giver spilleren et incitament/mål (at fange bondemanden) og er samtidig en god "kravlegård" til at lære spillet at kende.

Bane 2

Inde i laden bliver der sat flere kontraster: Musikken intensiveres og fremstilles mere elektronisk-mekanisk frem for det glade banjo-musik, der var i bane 1; sværhedsgraden forøges, og spilleren bliver mødt af flere typer fjender; og man ser de første spor af bondemandens onde værk i form af lemlæstede og døde høns, som ligger rundt omkring i banen.

Ved at lade bane 1 være en slags "kravlegård", kan vi forudsætte, at spilleren nogenlunde har lært spillets basale mekanik at kende. Derfor kan vi tillade os at øge udfordringerne, hvilket gøres ved at introducere nye typer fjender, som bevæger sig rundt i banen. Her mødes Nugget af ondsindede køer, der patruljerer starten af banen, samt farlige savklinger (se billede 1.3) og en kæmpeedderkop. Derudover støder spilleren også på ekstraliv i form af *pickups*, som man samler op for at forøge sit antal af livsforsøg (som fra starten af tre). Livene symboliseres i form af små ænder, der bliver vist oppe i venstre hjørne af skærmen.



Billede 1.3 Savklingerne i bane 2 udgør en større udfordring, da nogle af disse bevæger sig frem og tilbage i en utilregnelig rytme. Hvis Nugget rører en klinge, mister han et liv. Heldigvis er der placeret et ekstraliv på banen (en gummiand), som han kan samle op.

Bane 3

Denne bane foregår under jorden i et hemmeligt våbenkammer, som bondemanden har bygget. Banen er væsentligt længere og sværere end de to foregående, men denne gang bliver Nugget i stand til at svare igen mod spillets fjende, da han finder et maskingevær. Ved at trykke på mellemrumstasten affyres projektiler, hvilket er nødvendigt, da Nugget konfronteres med en soldat-gris, som skyder om sig med sin bazooka-raket (billede 1.4).

Generelt er den tredje bane en sammenblanding af de to tidligere. Nugget skal bl.a. forcere søer af giftig syrer og hoppe på lamper, som hænger over en faretruende gruppe af savklinger.



Billede 1.4 Bane 3 er mere dystert og udfordrende. Her har Nugget netop samlet et våben op, som skal bruges til at besejre en ond gris, der står i vejen

Bane 4 - Boss Battle

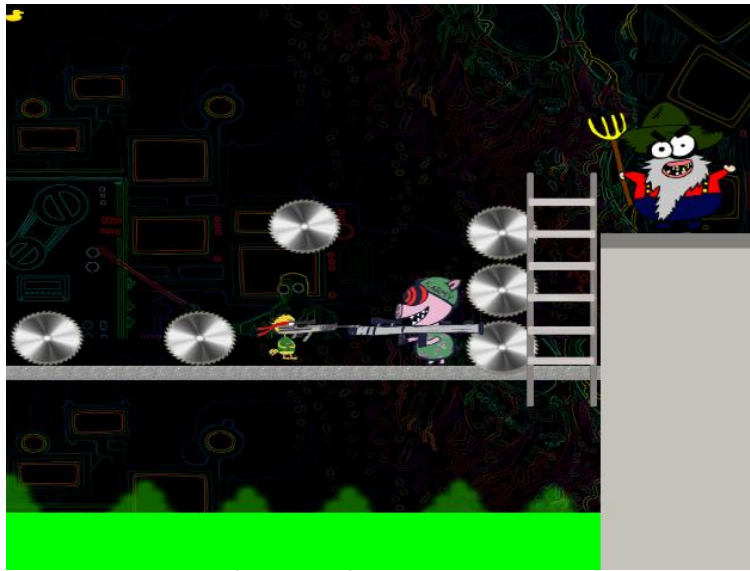
Med inspiration fra Nintendos *Super Mario Bros.* (se billede 1.5) slutter spillet af med en endegyldig kamp mellem kyllingen Nugget og den onde bondemand. Her skal spilleren bruge samtlige færdigheder, han har lært op til nu for at besejre bondemanden og hans hær af onde dyr.

I modsætning til alle andre baner, er skærmen i boss-kampen fastlåst. Her har bondemande positioneret sig i højre del af banen bag, hvor han slynger forskellige fjender og forhindringer ud. Det starter meget nemt med nogle enkelte får, men hurtigt forøges tempoet, og pludselig befinder der sig adskillige fjender og savklinger på skærmen, som Nugget skal undgå (se billede 1.6). Banen er forholdsvis svær og kræver ofte et par forsøg. Der er lagt op til mønster-genkendelse, idet fjenderne følger en fastsat rytme. Formår spilleren at undvige alle forhindringerne, vil han til sidst møde bondemanden i en ultimativ kamp på liv og død.

Det viser sig dog hurtigt, at bondemanden egentlig ikke udgør nogen stor trussel uden hans hær af ondsksfulde dyr, hvorfor Nugget hurtigt besejrer ham. Efter at bondemanden er blevet nedkæmpet, begynder skærbilledet at ryste, og en giftig syreflod stiger langsomt op. Nugget skal hurtigst muligt klatre op ad stigen i højre del af banen for omsider at kunne genforenes med sin familie.



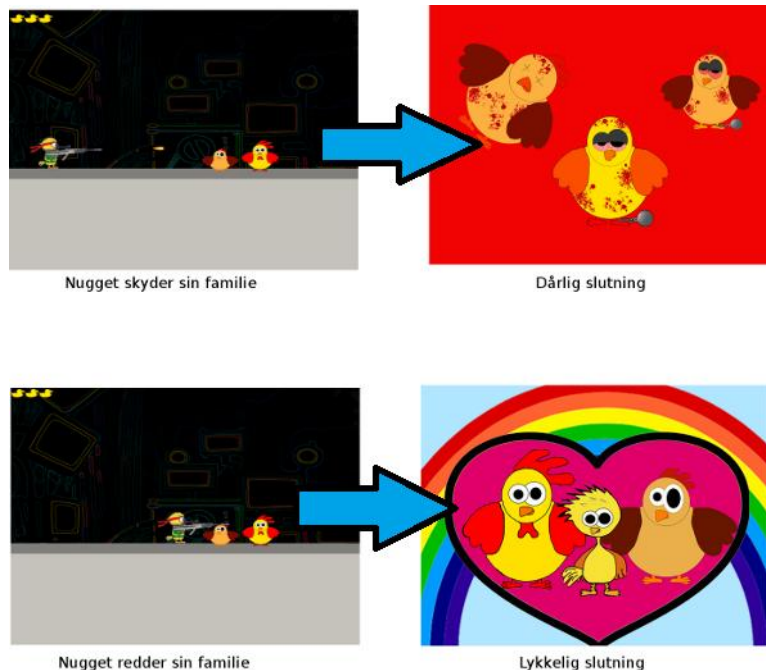
Billede 1.5 Her ses den sidste boss-kamp i 1985-spillet *Super Mario Bros.* til Nintendo Entertainment System (Nintendo).



Billede 1.6 Spilleren bliver virkelig sat på prøve, når der skal hoppes over savklinger og nedlægges fjender. Musikken er her intens og stressende, og banen kræver ofte adskillige forsøg.

Epilog - To forskellige slutninger

Som en lille ekstrating får spilleren til sidst et valg, da han genforenes med sin hønse-familie. Hvis spilleren blot går hen til hønsene, slutter spillet med regnbuefarver og glad musik; men hvis spilleren i stedet “kommer til” at trykke på aftrækkeren og skyder hønsene, vil man blive mødt med en trist slutning, hvor hønsene ligger døde. På den måde sættes spillerens tanker og handlinger lidt i perspektiv: Nogle gange handler vi, før vi tænker, og dette kan have fatale konsekvenser. Måske var det ikke meningen, at vi ville skyde, men fordi vi op til nu har skudt på alt, der bevæger sig, er det intuitivt at trykke på mellemrumstasten. Dette er gjort for at give spillet et lidt interessant twist, om end det i sidste ende ikke bidrager til selve gameplayet. Sekvensen er bevidst holdt implicit, så spilleren selv kan tolke, hvad han vil.

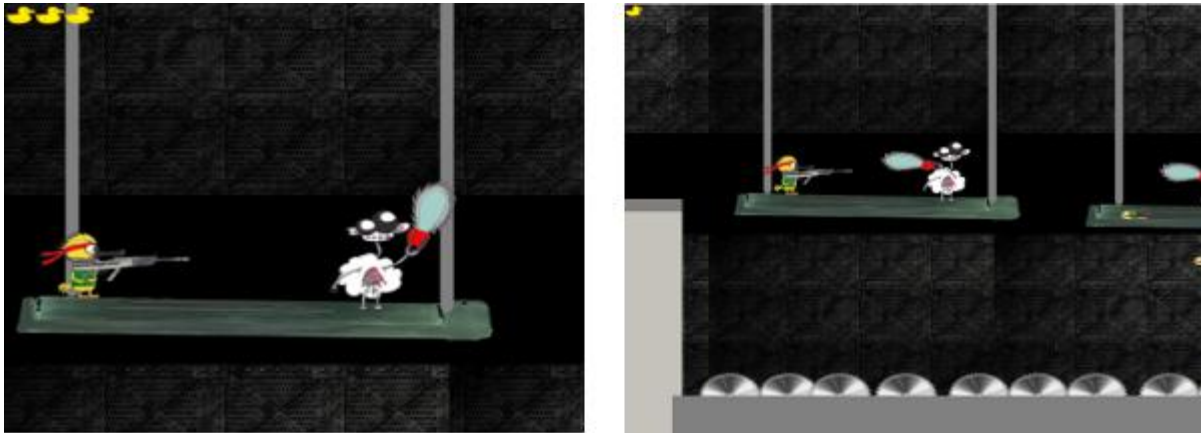


Billede 1.7 To forskellige slutninger: Vælger spilleren at redde eller dræbe sin egen familie?

Overgang mellem banerne og perspektiv-skift

For at skabe en mere holistisk helhedsoplevelse er overgangen mellem hver bane designet således, at man altid ved, hvor man bevæger sig hen. Eksempelvis går man i bane 1 fra et udendørs miljø hen mod en lade. I bane 2 befinder spilleren sig inde i laden, men kan stadig se den dør, som han trådte igennem i forrige bane. På samme måde er der i slutningen af bane 2 en stige, som man kravler nedad, og i bane 3 ser man Nugget komme ned ad selvsamme stige. Game Maker har indbygget en række overgangseffekter, når der skiftes baner, hvor eksempelvis skærbilledet bliver skubbet fra venstre mod højre. Disse er med fordel brugt i spillets baneskift.

Kameraperspektivet i *Nugget's Revenge* er forsøgt holdt konsistent hele vejen igennem, hvor kameraet konstant er fokuseret på den lille kylling. Dog opdagede vi i playtestning, at der i slutningen af bane 3 var lidt problemer med at danne sig et overblik, fordi banen her bruger lidt mere vertikalt plads end de andre baner. Derfor valgte vi, at når spilleren kommer til et bestemt punkt på banen, bliver kameraet zoomet en smule ud. Løsningen er ikke helt elegant, fordi skiftet virker meget brat, men den er klart bedre, end hvis kameraet fortsatte med nærbillede (se billede 1.8).



Billede 1.8 I den første prototype (venstre) var kameraet meget nær, hvilket gjorde det svært at danne sig et overblik. Derfor zoomer kameraet lidt ud (højre) i den færdige version, når Nugget kommer til et bestemt punkt.

Feedback og justering af sværhedsgrad

Det er vigtigt, at spilleren konstant får feedback på de ting, han udfører. Her har især lydeffekterne været i fokus, fordi lyd er rigtig godt til at kommunikere, om man gør noget rigtigt eller forkert. Et eksempel herpå er når man bliver skadet af en fjende, eller når man selv dræber en fjende. Hver situation har sin egen lyd; det er vigtigt, at lydeffekterne er distinkte fra hinanden, så spilleren aldrig er i tvivl om, hvad der sker.

Der skal tages hensyn til nogle kriterier, når effekterne bliver designet, da de har stor indflydelse på hvordan spillet opleves af brugeren. De lydeffekter, som gentager sig mange gange, skal ikke blive irriterende at høre på efter kort tid - eksempelvis den lyd, der afspilles, når man hopper. De effekter, man kun hører enkelte gange i spillet, fx dem fra fjenderne, er mere markante og højere. Fårene har eksempelvis fået en brøllende motorsavs-lyd, som intensiveres, jo tættere på Nugget befinder sig. På den måde bliver spilleren advaret om, at fårene er farlige. Sammen med en simpel animation, hvor motorsaven svinges over hovedet, gør lyden fårene mere levende i deres fremtoning.

Det har fra start været tiltænkt, at spillet skal være en udfordring at komme igennem. Selvom den første bane er meget nem, bliver sværhedsgraden gradvist forøget. For ikke at gøre det for svært er der placeret en række ekstraliv i banerne. Derudover er spillet udformet således, at hvis Nugget dør, skal han ikke altid starte helt fra begyndelsen af banen. Såkaldte *checkpoints* er sat ind i de to sidste baner, hvilket gør, at hvis spilleren laver en fejl, skal han ikke begynde helt forfra. Det samme gør sig gældende i den sidste boss-bane: Selvom man mister alle sine liv, begynder spillet ikke fra bane 1, men i stedet genstarter boss-kampen.

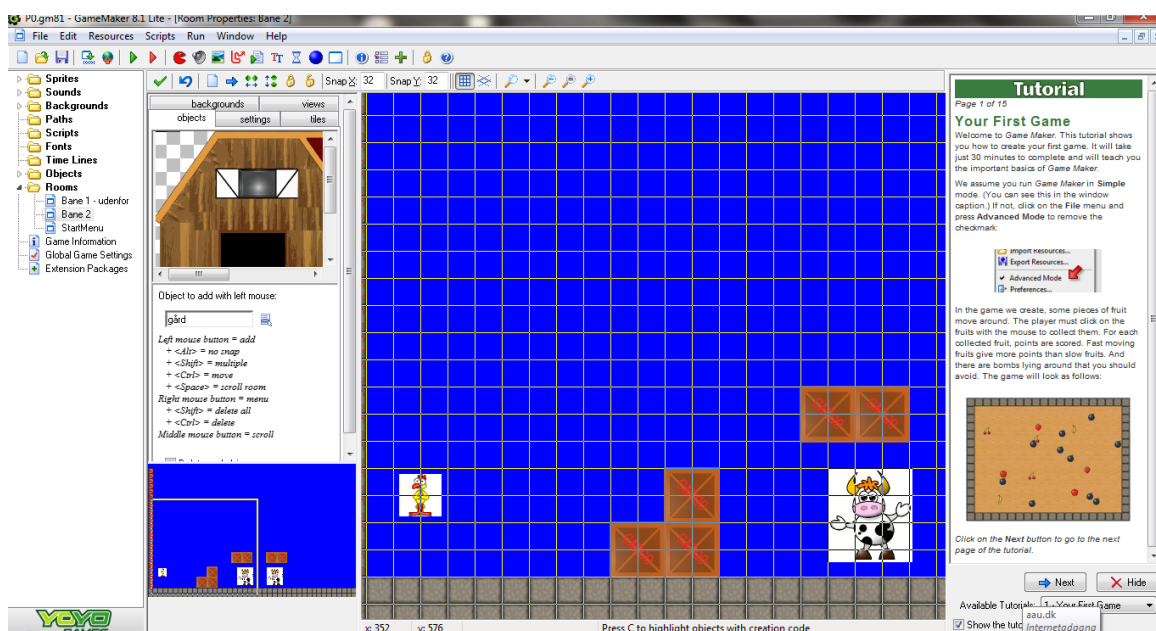
En anden ting, som har været vigtig i forbindelse med sværhedsgraden, er, at når Nugget dør og starter på banen igen, bliver han udødelig i nogle få sekunder. Dette har især været nødvendigt i boss-kampen,

da man ellers vil risikere at dø gentagne gange, fordi man starter op tæt på en savklinge eller andre fjender. For at kommunikere dette til spilleren, blinker Nugget, når han er udødelig.

Rapid prototyping og placeholder-grafik

Før vi begyndte at arbejde i Game Maker, brugte vi papir og blyant til at komme med idéer til spillets baner. Det er langt hurtigere at lave *level design* på papir, da man her nemt kan viske ud og ændre banens udformning. Efter at have designet spillets baner udviklede vi en simpel prototype for at teste, om spillet fungerede, var udfordrende og vigtigst af alt: sjovt. Her var ikke lagt væk på spillets udseende, men udelukkende på dets mekanik: Hvordan Nugget bevæger sig, hvor højt han kan hoppe, hvor meget liv fjenderne har, banernes design, etc. I stedet for at bruge lang tid på at tegne grafikken selv, fandt vi nogle midlertidige billeder fra nettet, som senere hen er blevet erstattet af tegninger, vi selv har lavet. Denne løsning gav os en god fornemmelse af størrelsesforholdet (Nugget er lille i forhold til spillets fjender) og Game Makers indbyggede level editor.

Efterfølgende har vores tilgang været mest iterativ, hvor vi konstant har forbedret og rettet fejl/mangler, mens de er dukket op. Dette viste sig at være gavnligt, fordi vores krav til spillet hele tiden ændrede sig, og vi måtte gradvist fjerne flere og flere features for at holde den respektive tidsgrænse. Set i lyset af at vi samtidig lærte at bruge Game Maker, mens vi udviklede spillet, var den iterative tilgang et bedre valg end eksempelvis vandfalds-modellen, som handler meget om bevæge sig igennem en række faser uden at se sig tilbage. Med førnævnte prototype og den iterative model var vi i stand til at have en langt mere dynamisk tilgang til udviklingen af *Nugget's Revenge*, som gjorde, at vi var i stand til at holde os inden for et realistisk *scope*.



Billede 1.9 En tidlig prototype med placeholder-grafik for at give en idé om, hvordan spilles føltes.

Workflow i Game Maker

Game Maker er som standard inddelt i adskillige mapper som “sprites”, “objects”, sounds”, “rooms”, mv. Dette skaber grobund for en strukturering af filer, men det kræver ligeledes, at man er lidt opmærksom selv. For har man ikke nogle konventioner om filnavne og -typer, kan det hurtigt bliver uoverskueligt. Derudover tillader den gratis version af Game Maker 8 ikke, at flere personer arbejder på et projekt samtidig. Derfor er alle asset-filerne (grafik, lyd, scripts, osv.) i sidste ende nødt til at være samlet på én computer.

En anden ting, man skal være opmærksom på, er, at Game Maker pakker alle éns filer sammen til en enkelt fil af formatet .gm81. Dette betyder, at assets skal gemmes manuelt et eksternt sted i modsætning til andre game engines, hvor filerne automatisk kopieres over i en projektmappe (eksempelvis Unity 3D). På den anden side er Game Maker rigtig god til at tage backups, hver gang man laver en *build* af spillet.

Man skal også være bevidst om, hvilke filformater, man arbejder med. Eksempelvis er der forskel på, om man bruger lydformatet MP3 eller WAV. Game Maker håndterer nemlig filerne forskelligt, hvilket gør, at man godt kan spille et MP3-klip og WAV-klip oven i hinanden; men prøver man at afspille to MP3-filer samtidig, fungerer det ikke.

En anden ting, som man skal være opmærksom på, når man arbejder i Game Maker, er billedernes størrelse. Selvom Game Maker indeholder en *image resizer*, er det bedst at skalere sine billeder i et eksternt program. Det er nemlig ikke muligt at se størrelsen på billedet i pixels, når man vil skalere, men kun i procent, hvilket gør det svært at vurdere, hvor store billederne skal være.

Workflowet i Game Maker er generelt flydende, men på grund af dets drag-and-drop struktur også nogle gange lidt trægt, fordi der skal klikkes igennem mange menupunkter. Game Maker lægger ikke op til objekt-orienteret programmering, da hverken spilmotoren eller dets respektive programmeringssprog GML (som er tæt beslægtet med C) tillader brugen af classes, inheritance eller encapsulation. I stedet er alle objekter automatisk *instantiated*, hvilket gør det svært at arbejde med forskellige udgaver af samme objekt. I stedet er man nødsaget til at duplikere opførsler fra objekt til objekt, hvilket hurtigt kan blive meget uoverskueligt. Et eksempel herpå er grisen i *Nugget's Revenge*, som opfører sig forskelligt i bane 2 og i boss-banen. I stedet for blot at ændre nogle få variabler i de enkelte objekter, er man tvunget til at oprette et helt nyt objekt, hvis man ønsker en anderledes opførsel.

Musik

Musikken har i spillet været en væsentlig faktor i forhold til at skabe de rette stemninger i de forskellige miljøer. Den er med til at skabe en helhed samt en genkendelighed for spilleren. Alle musikalske indtræfninger i spillet er produceret i gruppen. Der er blevet lagt vægt på at skabe kontraster imellem den musik, som foregår i selve gameplayet og den musik, som bliver afspillet i spillets indledende historie-sekvenser. Grunden til denne kontrast er for at skelne de to forskellige dele af spillet og for at

give det mere alsidighed og dybde i forhold til et mere sofistikeret lydunivers. I historie-faserne har vi valgt at bruge akustisk lyd, hvor vi modsætningen til selve gameplayet har valgt at bruge *retro arcade*-lyd.

Det lydtekniske - *equalizing, compriming, mastering* - er lavet ud fra en forventning om, at spillet oftest vil blive spillet på en computer med middelmådige højtalere, eksempelvis en bærbar computer. Dette medfører, at de lavere frekvenser (bass-frekvenserne) ikke vil være tilstede, når spillet kører på en bærbar computer. De helt lave frekvenser på al musikken er derfor minimeret i produktionsfasen, da den ikke vil have nogen auditiv indflydelse på spillets gameplay. Desuden kan sub-frekvenserne skabe forvirringer i lydbilledet, hvis det blev spillet på en bærbar computers højtalere, som ikke besidder en sub-/bass-udgang

Der er i alt blevet udarbejdet/komponeret seks stykker musik - et stykke til hvert formål i forhold til spillets kontekst:

Det første stykke musik - som man hører i sammenhæng med intro-sekvensen - er et akustisk/symfonisk musikstykke. I dette er der blevet brugt akustiske strygere, brassere og kor til at skabe en dramatisk stemning, som følger i takt med spillets historie.

I spillets første bane er der blevet brugt banjoinstrumenter til at skabe en glad og idyllisk stemning. Banjoen er med til at tilføre spillet en autentisk stemning af, at man befinder sig på landet, men stadig med en vis energisk intensitet, der fungerer som et forvarsel til spillets videre historie.

Musikken i bane 2 lægger vægt på en flydende overgang fra det glade land-idyl til et mere intenst miljø og gameplay. Her præges lydbilledet af syntetisk-mekanisk musik, som gentagne gange loops for at bevare tempoet i gameplayet.

I den efterfølgende bane sker der en dramatisk ændring i billede og lyd. Bane 3 foregår i et underjordisk våbenlager, hvor både gameplay og musik er mere intenst og udfordrende. Musikken er her udviklet på baggrund af, at spilleren skal kunne mærke en markant ændring i gameplayets seriøsitet. Her er der valgt dramatiske synthesizer-lyde, der kører i et hurtigt tempo, som følges i takt med underliggende, syntetiske trommer.

Da spilleren omsider når til spillets sidste bane, boss-kampen, bliver både lydniveauet og -intensiteten skruet op. Musikken afspejler her Nuggets indre vrede og hævn-gerrighed i form af et nummer, der indeholder traditionelle elementer som hurtige strygere og klaver, men som tager sin energi fra overdrive- og distortion-guitar og trommer. Dette resulterer i et stykke musik, som er konstrueret som moderne speed metal, der bruger klassiske instrumenter til at indfange opmærksomheden omkring kernepunkter og har tempoet, lethed og enkeltheden, der er karakteristisk for arkade-spil.

I spillets slutscene og epilog, hvor Nugget genforenes med sin familie, er der igen blevet produceret/komponeret et stykke akustisk musik med klokker, kor, og strygere til at formidle en følelsesladet stemning, som efter hensigten skal opstå med genforeningen.

Evaluering

Det færdige spil er noget anderledes end det, vi havde i tankerne de første dage. Grundet tidsplanen har vi måtte nedskære visse ting i spillet. I begyndelsen var det eksempelvis planen, at Nugget skulle have mulighed for at opsamle flere forskellige våben, der hver havde unikke egenskaber - fx shotguns, flammekaster, granater, m.fl. Samtidig skulle fjenderne have mere karakteristiske bevægelsesmønstre og angreb, således at man som spiller skulle tænke taktisk over, hvilket våben man brugte i en given situation. Tanken var at give spillet mere strategisk dybde, og at hver våben havde både fordele og ulemper. Derudover skulle det i spillet være muligt at opsamle diverse *power-ups*, som gav Nugget anderledes evner så som at flyve/svæve og dobbelt-hop.

I sidste ende valgte vi at skære ned på alt det, som ikke var essentielt for spillets kerneoplevelse. Hele spillet er udviklet efter filosofien, at selve kernen af spillet skulle på plads, før vi tilføjede ekstra ting. Med det begrænsede tidsrum var det vigtigt at holde sig inden for rammerne, og ikke forsøge at få for mange features med (*feature creep*). Det kunne være interessant med flere våben og fjendetyper, men vi vurderede, at det ikke var altafgørende for spillet, og at det godt kunne fungere med færre våben. Det samme gør sig gældende med spillets baner: Oprindeligt havde vi designet flere baner, men på grund af tidsmangel valgte vi at slå nogle af dem sammen. Dette viste sig at være en god idé, da det mindskede behovet for at producere mere grafik og lyd, som vi allerede havde forholdsvis meget af.

En relativ central ting som vi fravalgte hurtigt var en *multiplayer*-del. Det kunne være sjovt at spille flere personer samtidig, men da hverken spillets design eller genre lagde op til det, valgte vi kun at fokusere på én spiller. Ligeledes valgte vi ikke at inkludere nogen form for highscore-liste grundet spillets lineære struktur. Vi vurderede, at et element som points ikke ville tilføje noget til spillet.

Konklusion

I sidste ende valgte vi ikke at lave det mest teknisk avancerede spil, men i stedet at udvikle et solidt platform-spil med charme og sjove figurer. Mange spil, som slår igennem i dag, indeholder ikke udelukkende nye og innovative idéer, men tager i stedet allerede-eksisterende koncepter og polerer dem, indtil at de er fuldstændig strømlinet. Et eksempel herpå er finske Rovio, der i 2009 udgav spillet *Angry Birds* til iPhone. Spillet er i dag blevet downloadet over 350 millioner gange og findes nu til et hav af forskellige platforme. (CNN, 2011). Spillets hoveddesigner, Jaakko Iisalo, fortæller i et interview med spil-magasinet EDGE, at hemmeligheden bag gigantsuccesen at give spillet "hjerte og sjæl":

"You have to have a great idea for a game. It has to have heart and soul. For me, that's the most important thing. I see a lot of technically great games, but they have no soul and the content is lacking." (Iisalo, 2011)

Med *Nugget's Revenge* har vi forsøgt at skabe et interessant univers med nogle letgenkendelige figurer, som skiller sig ud fra mængden. Havde vi haft mere tid til rådighed, ville vi nok ønske flere baner, mere variation i fjendernes adfærd, samt en større palette af våben. Alligevel føler vi, at spillet i dets

nuværende form er en fin, kompakt oplevelse, der fungerer efter hensigten: et sjovt, lille spil, der er let at gå til og tilbyder en tilpas mængde udfordring.

Værktøjer

Under udviklingen af *Nugget's Revenge* brugte vi adskillige værktøjer. Herunder ses en liste over programmerne og en kort beskrivelse af, hvordan de er blevet brugt.

- **Game Maker 8.1.135:** Spilmotoren og IDE'et bag *Nugget's Revenge* - brugt til level design, programmering, mv.
- **Photoshop CS5:** Er blevet brugt til at tegne og redigere de forskellige sprites til spillet
- **Ableton Live 8.1.2:** En del af spillets musik er sammensat i Ableton
- **Audacity 1.3:** Open-source lydredigeringsprogram - er blevet brugt til at klippe og justere lydniveau
- **Adobe Audition 3.0:** Programmet er blevet brugt til at redigere småting i nogle af lydeffekterne, samt konvertere dem til de ønskede filformater
- **Guitar Pro 5:** Dette program blev brugt til at skabe de rå midi-filer til henholdsvis banjo- og boss-musikken - basal komposition
- **Sonar X1:** Sonar er et lydredigeringsprogram, der vha. plugins giver adgang til et enormt bibliotek af instrumenter, som lyder forholdsvis ægte - blev brugt til at behandle midi-filerne fra Guitar Pro 5
- **sfxr:** Et gratis program til at generere lydeffekter - er blevet brugt til bl.a. hoppelyd
- **Google Documents:** Gratis cloud-baseret kontorpakke - er blevet brugt til rapportskrivning
- **Dropbox:** Smart fildelings-program, som tillader alle i gruppen at dele filer med hinanden

Referencer

- CNN (2011): <http://tech.fortune.cnn.com/2011/09/12/angry-birds-350-million/>; tilgået 22. september 2011
- Iisalo, Jaakko (2011); *An audience with ...*; EDGE Magazine; #232; september

Eksterne ressourcer

Vi har ikke brugt deciderede kilder i P0-forløbet. Stort set alt i spillet er lavet/redigeret selv. Dog har vi benyttet en række hjemmesider og fora for at få hjælp til Game Maker, samt diverse scripts. Enkelte af lydfilerne er ligeledes hentet fra et gratis online-lydbibliotek kaldet SoundSnap.

- <http://www.soundsnap.com> - diverse lydeffekter (motorsav, dyreløyd, m.m.)
- <http://gamemaker.info>
- <http://game-maker.wikidot.com>
- <http://wiki.yoyogames.com>
- <http://testgmc.yoyogames.com>
- <http://gamemaker.wikia.com>
- <http://gmc.yoyogames.com>