

Yume Nikki Inspireret videospil

Produkt Rapport

Projekt af: Jack Conradsen

Lærer: John Ellehammer

Tec Ballerup

03-03-2023



1 Indholdsfortegnelse

[1 Indholdsfortegnelse](#)

[2 Kravspecifikation](#)

[2.1 Tekniske krav](#)

[2.1.1 Platform](#)

[2.1.2 Controls](#)

[2.1.3 Indstillinger](#)

[2.1.4 Refresh Rate](#)

[2.2 Systemer](#)

[2.2.1 Save system](#)

[2.2.2 Movement & Collision system](#)

[2.2.3 Kamera system, skal oppdateres](#)

[2.2.4 Tekst box system](#)

[2.3 Menuer](#)

[2.3.1 Start-menu](#)

[2.3.2 New save-menu](#)

[2.3.3 Load save-menu](#)

[2.3.4 Pause-menu](#)

[2.3.5 Effect inventory](#)

[2.3.6 Indstillinger](#)

[2.4 Gameplay elementer](#)

[2.4.1 NPC'er](#)

[2.4.2 Triggers](#)

[2.4.3 Interaktive Objekter](#)

[2.4.4 Dører](#)

[2.4.5 Effects](#)

[2.5 Udviklingsmiljø \(Gamemaker studio 2\)](#)

[2.6 Versions styring \(Github\)](#)

[2.7 Ekstensions](#)

[2.7.1 Snowstate](#)

[2.7.2 Input](#)

[2.7.3 Lexicon](#)

[2.7.4 STANNcam](#)

[2.8 Testing](#)

[2.9 Visuelle krav](#)

[2.9.1 Text](#)

[2.9.2 Pixel-art](#)

2 Kravspecifikation

2.1 Tekniske krav

2.1.1 Platform

Det udvikles til PC

2.1.2 Controls

Det skal kunne spilles med tastatur, og eller en controller.

2.1.3 Indstillinger

Der skal være en indstillings menu, hvor man kan ændre sin skærmstørrelse. Og ændre om spillet skal være i fullscreen/windowed mode. Og ændre hvad sprog spillet bliver vist på.

2.1.4 Refresh Rate

Spillet skal køre med 60 FPS.¹

Siden det er et 2D pixel art spil, vil det ikke kræve mange ressourcer for at køre selv på low-end hardware. Derfor er 60 FPS meget nemt opnåeligt. Den høje FPS er mest til for at sikre responsivt input, trods at de fleste sprite animationer vil køre med meget lavere FPS.

2.2 Systemer

2.2.1 Save system

Et system der kan gemme spil data, og loadere eller slette et spil gem. Det skal også være muligt at gemme indstillinger.

2.2.2 Movement & Collision system

Et system så både spiller, NPC'er, og andre objekter kan bevæge sig, og kolliderer med omverdenen.

2.2.3 Kamera system, skal opdateres

Kameraet skal kunne følge spilleren inden for en vis margin, blive inde for spille området. Derudover skal det styre shader effekter der måtte komme på, camera-shake, unniike baggrunde, farve effekter. Og sørger for at opskalere spillet til skærmstørrelsen, grundet at spillet i realiteten har en meget lille opløsning.

¹ FPS (Frames per second) https://en.wikipedia.org/wiki/Frame_rate

Derudover skal det også være muligt for kameraet seamless at gå ud fra venstre side af en bane og komme ind til højre.

Jeg benytter min egen extension STANNcam, men vil i løbet af projektet opdatere den til at kunne håndtere flere kamerae ad gangen.

2.2.4 Tekst box system

Tekst box system der nemt kan kaldes til at tegne tekst på skærmen. Den skal også være i stand til at få teksten til at dukke op et symbol ad gangen efter noget tid, ligesom i klassiske RPG spil.

2.3 Menuer

Her beskrives de menuer som er planlagt i spillet. Nogle menuer skal kun kunne ses i starten af spillet, andre mens man spiller, nogle begge steder.

2.3.1 Start-menu

Det første man ser når man starter spillet. Her skal man kunne gøre følg.

- Start et nyt spil.
- Load et tidligere gemt spil.
- Gå til indstillinger.
- Afslutte spillet helt.

2.3.2 New save-menu

En menu der gør det muligt at lave et nyt spil save, eller overskrive et eksisterende.

2.3.3 Load save-menu

En menu der gør det muligt at loade eller slette eksisterende spil saves.

2.3.4 Pause-menu

Spilleren kan åbne en inventar, som også pauser spillet midlertidigt, her vil man kunne vælge imellem forskellige effekter. Her skal man kunne gå til følgende

- Effect inventory
- Gå til indstillinger
- Lukke spillet
- Lukke inventaren igen

2.3.5 Effect inventory

Denne vil indeholde effects der kan equippes. Menuen sættes op som 2 kolonner, og blive fyldt ud hver gang man opsamler en ny effect. Rækkefølgen af de forskellige effekter skal være prædefineret, så altså hvis man opsamler effekt2, før effekt1, så skal effekt 1 stadig stå først når de begge er samlet.

Menuen skal være tom medmindre man er i drømmeverdenen.

2.3.6 Indstillinger

Dette er menuen hvor man kan ændre skærmstørrelse, om det er fullscreen, eller windowed mode. Samt sprog.

Denne menu skal være tilgængelig både fra start menuen. Og under selve spillet. Så hvis man er kommet fra start-menu, skal man kunne gå tilbage til den, Og hvis man er kommet fra Inventory-menu (altså under selve spillet) skal man kunne gå tilbage til den.

2.4 Gameplay elementer

Objekter der kan forekomme i en bane.

2.4.1 NPC'er

Karakterer der går rundt tilfældigt, som der kan interegeres med.

2.4.2 Triggers

Et objekt som er en usynlig kollisions boks i banen, der vil trigger noget kode når spilleren kolliderer med den

2.4.3 Interaktive Objekter

Objekter spilleren kan interagere med, og så udløse noget kode. Nogle af disse objekter, skal kodes så de kun aktiveres når bestemte effekter er taget i brug.

2.4.4 Dører

Dører der kan interagere med som skifter banen, og bestemmer hvor præcist i den næste bane spilleren skal starte. Altså siden forskellige verdener leder ind i hinanden, skal man kunne bestemme hvor præcist i en verden spilleren ender når man kommer derhen.

2.4.5 Effects

Effekter bliver opsamlet løbende i spillet, når man interagerer med bestemte NPC'er eller objekter, de bliver så tilføjet til Effect inventory menuen, og her når man vælger en effekt, bliver spiller karakterens udseende og kode ændret. Herunder er alle effekterne i spillet beskrevet:

1. Penguin - Forvandler spilleren til en Pingvin, der kan glide på maven. Og så bevæge sig meget hurtigere.
2. TV - Forvandler spillerens hoved til et tv, tv'et kan tændes og slukkes, hvis tændt vil NPC'er inden for en hvis omkreds komme tættere på spilleren.
3. Stick - Gør spilleren meget tynd, ligesom en pind. Man bevæger sig også en smule langsommere, man kan dog komme igennem meget smalle åbninger. Hvis man fjerner effekten mens man i en meget smal åbning, dør man (i drømmen) og vågner igen.
4. Nose - Forvandler spilleren til en kæmpe næse med fødder. Mens man er en næse kan man snuse til npc'er og bestemte objekter og en tekstboks vil så komme frem der beskriver hvordan de lugter.

Efter projektets forløb vil flere effekter blive tilføjet.

2.5 Udviklingsmiljø (Gamemaker studio 2)

Spillet skal udvikles i *Gamemaker*,² da det egner sig perfekt til *2D-spil*.

Gamemaker er en *IDE*³ Som eksklusivt bruger sit eget kodesprog *GML*.⁴ Man kunne sige at det er et objekt orienteret sprog, i det at alt kode foregår i objekter, som bliver placeret manuelt (eller ved hjælp af kode fra andre objekter) i banerne.

2.6 Versions styring (Github)

*Github*⁵ bliver brugt til versionsstyring, og som backup. Det hjælper til at projektet kan arbejdes videre på selv hvis source filerne går tabt fra min egen computer. Github er også integreret med Gamemaker Studio 2 som standard, så der nemt kan pulles og pushes.

² Gamemaker <https://en.wikipedia.org/wiki/GameMaker>

³ IDE (Integrated Development Environment)

https://en.wikipedia.org/wiki/Integrated_development_environment

⁴ GML (Game Maker Language) https://en.wikipedia.org/wiki/GameMaker#GameMaker_Language

⁵ Github <https://en.wikipedia.org/wiki/GitHub>

2.7 Ekstensions

Flere offentlige Gamemaker studio 2 extensions skal bruges under projektet for at streamline udviklingen. Alle har licenser der tillader kommerciel brug, hvis det skulle blive relevant.

Disse pakker ligger offentligt tilgængelige på github og bliver importeret manuelt. Herunder er ekstensions jeg vil tage i brug beskrevet.

2.7.1 Snowstate

*Snowstate*⁶, er en række funktioner der gør det let at lave og håndtere state-machines i ens program. I dette projekt vil jeg bruge state machines til at håndtere spille karakterens mange effects og animationer. Håndtere hvilken menu der er i brug. Og andre mindre roller.

2.7.2 Input

*Input*⁷ er en række funktioner der gør det nemt at håndtere flere forskellige input metoder. Den sørger for at alle moderne gamepads (Playstation, xbox, nintendo switch) Samt keyboard & mus, Kan benyttes uden særlige problemer.

2.7.3 Lexicon

*Lexicon*⁸, er en række funktioner og fremgangsmåde til at håndtere flere forskellige sprog nemt. Alt tekst i spillet er skrevet ind i en Json fil, som key value pairs, og ekstra sprog kan så nemmere blive tilføjet senere.

2.7.4 STANNcam

*STANNcam*⁹ er min egen gamemaker ekstention til at gøre det nemmere at håndtere kamera objekter og rendering af spillet i forskellige opløsninger. I løbet af projektets forløb vil jeg opdatere ekstentionen.

2.8 Testing

*Playtesting*¹⁰ af spillet sker løbende under udvikling, manuelt. Jeg vil i nogle tilfælde spørge venner og familie med hjælp til at spille igennem for at se om der er fejl, og ellers for at notere om ting burde designs anderledes.

Min fremgangsmåde når jeg udvikler spil, er et teste meget ofte. Når jeg har tilføjet et stykke kode, kører jeg strakts programmet og ser om det fungerer som det skal. I mange tilfælde vil jeg også finjustere nogle variabler, som f.eks spillerens hastighed, løbende under udviklingen.

⁶ Snowstate <https://github.com/sohomsahaun/SnowState>

⁷ Input <https://github.com/JujuAdams/Input>

⁸ Lexicon <https://github.com/tabularelf/lexicon>

⁹ STANNcam <https://github.com/jack27121/STANNcam>

¹⁰ Playtesting <https://en.wikipedia.org/wiki/Playtest>

2.9 Visuelle krav

2.9.1 Text

Alt tekst i spillet og dets menuer skal som udgangspunkt skrives på engelsk, men alt tekst skal skrives i en separat Json sprog fil. Så det er nemmere at tilføje flere sprog senere ved hjælp af ekstentionen Lexicon.

2.9.2 Pixel-art

Mange af de forskellige sprites for spilleren



Grafiken i spillet er *pixel-art*¹¹ og spillet skal have en pixelstørrelse på omkring 416 x 240. Og så skaleres op for at tilpasse sig skærmstørrelsen efterfølgende. Det er forholdsvis småt, men hjælper til at give det en retro følelse, som klassiske 2D pixel-art spil, deriblant Yume Nikki, som mit spil hovedsageligt tager inspiration fra, derudover gør det, det nemmere og hurtigere at tegne de grafiske elementer.

¹¹ https://en.wikipedia.org/wiki/Pixel_art