Simplicity

Nicky og Nicklas 2.m

Indhold

[Indledning 1](#_Toc481445212)

[Udviklingssprog og programmer 1](#_Toc481445213)

[Unity 1](#_Toc481445214)

[C# 1](#_Toc481445215)

[Java script 1](#_Toc481445216)

[Adobe Photoshop 1](#_Toc481445217)

[FL Studio 2](#_Toc481445218)

[Problemformulering 2](#_Toc481445219)

[Beskrivelse af spillet 2](#_Toc481445220)

[Scene 1: 2](#_Toc481445221)

[Scene 2 3](#_Toc481445222)

[Scene 3 3](#_Toc481445223)

[Scene 4 4](#_Toc481445224)

[Teori 4](#_Toc481445225)

[Variabler 5](#_Toc481445226)

[Betinget udførsel 5](#_Toc481445227)

[Løkker 6](#_Toc481445228)

[Arbejdsproces 6](#_Toc481445229)

[Afprøvning af spillet 9](#_Toc481445230)

[Forbedringer 9](#_Toc481445231)

# Indledning

Vi har valgt at lave et program, hvor vi har brugt Unity’s engine. Vi har selv sammen sat assets, samt lavet vores egne scripts, lyd og graffik. Vores program er et simpelt 2D platform spil. Det minder lidt at det gamle Super Mario, bare uden fjender. Spillet er meget simpelt, det gælder om at du skal hoppe fra blok til blok, for så at komme hen til et flag, som så fører dig videre til næste bane. Som det ser ud lige nu, så kan man ikke gemme sin progress, så du bliver nødt til at klare hele spillet i en omgang. Det burde heller ikke være noget problem da der indtil videre kun er to baner.

# Udviklingssprog og programmer

I vores program har vi gjort brug af to kodningssprog: C# og Javascript. Vi har dog mest benyttet C#.

## Unity

Vi har brugt Unity til at animere vores karakter, samt hvordan den skal hoppe. Det hjalp os også med at give bestemte brikker et hitbox, så den kunne registreres når karakteren ramte den. Det ses eksempelvis ved de små firkanter, flag, portaler og vores ”void”. Alle disse har en hitbox (eller Collider), som gør den

## C#

Stort set hele programmet er lavet ved hjælp af C#. Det gælder alt fra player movement til hvordan kameraet er stillet og skal bevæge sig.

## Java script

Det eneste som vi har brugt Java script til er at skifte eller reloade en scene. Grunden til at vi skiftede til Java script her var fordi den kode som vi havde lavet i C# ikke virkede. Vi prøvede derfor med en simpel kode i Java script som viste sig at virke.

## Adobe Photoshop

Adobe Photoshop blev brugt til at lave alt det graffiske som ses i spillet. Det eneste som vi ikke har lavet i Photoshop er de firkanter man hopper på. Der brugte vi nogle som Unity havde, da vi mente den passede godt ind. Vi har lavet det hele efter et pixel map som vi startede med at lave i Photoshop. Derefter tegnede vi bare pixel for pixel til vi havde det vi skulle bruge.

## FL Studio

FL Studio blev brugt til at lave den meget simple baggrundsmusik som er i spillet. Det består mere eller mindre af et beat som FL Studio allerede havde, som blev modificeret lidt til at få den til at lyde lidt dybere. Så blev den sat i et loop, så den hele tiden spillede.

# Problemformulering

Vi ved godt er der er uendelig mange forskellige 2D platform spil på markedet lige nu. Vi føler dog at næsten alle spillene er ens. Derfor vil vi gerne prøve at lave et nyt og unikt spil, men som stadig holder på det nostalgiske.

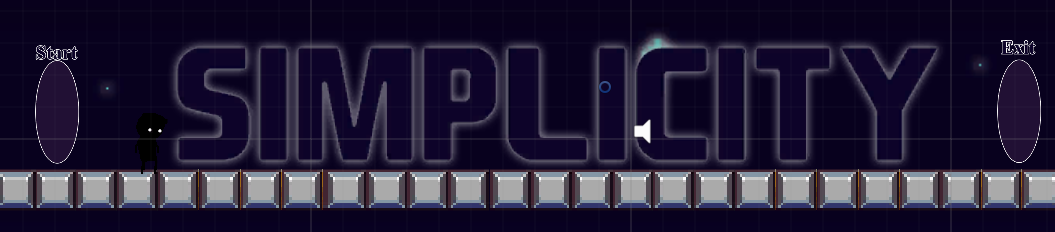
# Beskrivelse af spillet

Lige nu består spillet af fire scener. Spillet er som beskrevet tidligere et platformspil, hvor målet er et flag. Måden man kommer hen til flaget på er ved at hoppe rundt på nogle firkanter. Man kan også side fast på siden af firkanten, det kan bruges til at se hvor blokkene er, da nogle af dem ikke kan ses med det samme.

Ideen bag spillet var at det skulle være et simpelt 2D platform spil, som blev opbygget omkring parkour.

## Scene 1:

Den første scene er startsscenen. Det er her hvor man for første gang kan se karakteren, baggrunden, de blokke man hopper rundt på, navnet på spillet samt musikken. Scene et består af en lang række af de blokke vi bruger, så man kan hoppe lidt rundt, så er der to portaler, en som begynder spiller, og en som lukker spiller. Navnet på spillet står i baggrunden.

 (scene 1)

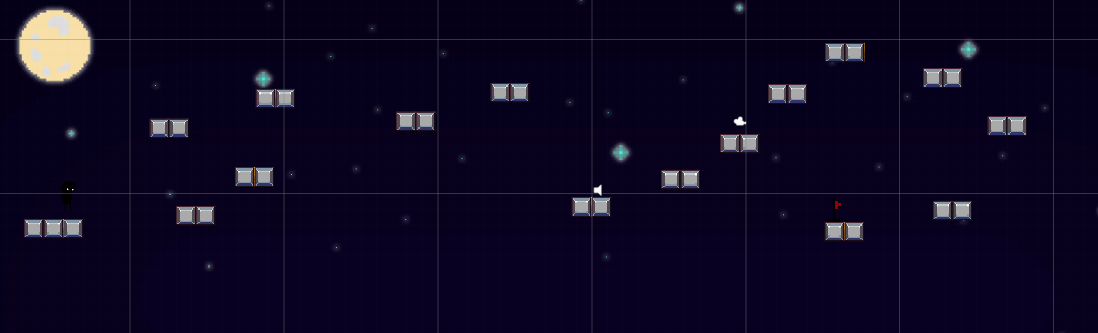
Oven over ses et billede af alt det relevante for den første scene.

## Scene 2

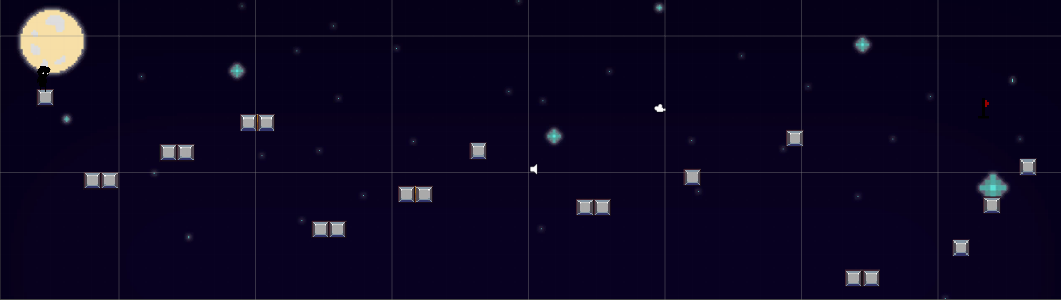
Hvis du vælger at gå gennemportalen til at starte spillet, så kommer du til anden scene, eller scene 2. Det er her hvor spillet for alvor starter. Det er her hvor du bliver nødt til at hoppe rundt på forskellige blokke for at komme videre.

Når scenen bliver loaded starter man på en platform som er tre blokke lang. Alle de andre platforme som man skal hoppe hen på er to blokke lange. Der er i alt 15 platforme som man skal hoppe hen på, hvilket vil sige i alt 30 blokke (dette er uden startsplatformen). På den sidste platform er der et lille flag, som det gælder om at ramme. Når man rammer dette flag bliver man sendt videre til scene 3.

Hvis man hopper og ikke rammer en platform, kommer man ned i voiden. Man bliver ikke sendt specielt langt ned før spillet reloader scene 2, så du kan starte forfra. Det er lige meget hvor og hvornår du rammer ved siden af, når du gør starter du forfra.

(scene 2)

## Scene 3

Den tredje scene er så det sidste færdige og spilbare level som vi har lavet. Opbygningen af dette level er næsten det samme som i scene 2. Det er stort set en helt identisk scene, bortset fra placeringen af blokkene, og antallet. I scene 3 starter man på en enkelt blok, i stedet for tre. Der er også platforme som kun består af en blok. Ellers er det det samme som i scene 2. 

## Scene 4

Scene 4 er en slags afslutningsscene. Formålet ved denne scene er at vise at spillet måske ikke er helt færdigt. Der er et skilt i baggrunden som siger at der kan komme flere baner senere hen. Vi havde ikke helt tid og følte ikke behov for at have mere end to baner. Scene 4 ligner meget scene 1, bortset fra der kun er en enkelt portal. Denne portal lukker spillet. Før du kan nå portalen bliver man nødt til at gå på en lige linje, med skiltet i baggrunden. Dette er for at sikre at folk ser det.



# Teori

Alle programmer, om det er på en computer, tv eller telefon er blevet programmeret i det ene eller andet sprog. Selv styresystemet som programmerne kører på er blevet programmeret. At programmere betyder egentlig bare at give en computer nogle instrukser. Dette betyder at når man går ind på sine beskeder på facebook, så er der en eller anden programmør som har siddet og skrevet linje efter linje af kode. Når man skal give en computer instrukser skal det være meget præcist hvad den skal gøre. Derfor har folk gjort det nemt og skrevet sprog til at programmere i. Ellers havde det taget en million år at skrive alt ind i binær.

## Variabler

For at forstå hvad en variable er skal vi først vide hvordan programmer håndterer hukommelsen. Når man skriver en variable ind i programmet snupper den en del af hukommelsen. Størrelsen af delen kommer an på variable typen. Der er forskellige typer variabler. De mest normale er:

Int: En int er et helt tal fra -2147483648 til 2147483647 eller 0 til 4294967295 hvis den er unsigned. Det svarer til 4 bytes.

Float: En float er et nummer med decimaler. Dette kan være en uendelig mængde tal.

Bool: En bool eller boolean er et enkelt binært tal. Altså enten sandt eller falskt, 0 eller 1.

String: En string er et stykke tekst. Det består at flere char’s, som der er enkelt bogstaver.

I variabler med tal kan man bruge operatorer til at sammenligne værdier. Dette kan f.eks. være =, >, <, !, & og |.

## Betinget udførsel

I programmering bruger man ofte betinget udførsel. Dette bliver brugt i kommandoen:

if(betingelse){

kommando

}

Det betyder at betingelsen skal være sand før at kommandoen kan køres. Altså hvis man har en variable ”skostørrelse” og en butik kun sælger sko mellem størrelse 37 og 45 kommer det til at se sådan ud:

if(skostørrelse < 45 && skostørrelse > 37){

sælg sko

}

Det vil sige at skostørrelsen skal være mindre end 45 OG større end 37 før at butikken kan sælge skoene.

## Løkker

En løkke er en kommando som bliver kørt om og om igen indtil at udtrykket ikke længere er sandt. Hvis man vil have en kommando til at køre uden stop kan man lave den sådan her f.eks:

while(1>0){

kommando

}

Man kan også bruge den med variabler som f.eks:

while(variable < 10) {

kommando

variable++

}

Denne løkke vil køre 10 gange da ”variable” starter på 0 og ”variable++” betyder at den plusser sig selv med 1 hver gang løkken bliver kørt.

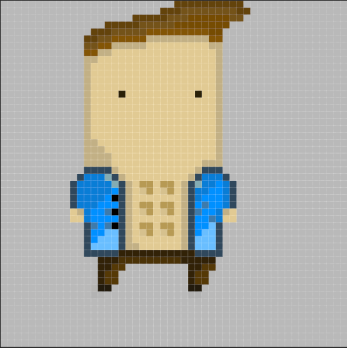
# Arbejdsproces

Da vi lidt efter jul blev sat i gang med at skulle lave vores eget program fra bunden, uden rammer, måtte vi selvfølgelig i gang med at generer ideer. Vi blev meget hurtige enige om at vi ville lave et spil, da vores begge fritids interesser ligger her. For så ikke at gøre det for kompliceret valgte vi et 2D platform spil. Vi begge har haft spillet Mario som mindre, så det var ikke kun for det ikke skulle være kompliceret, men også for at få lidt nostalgi ind over.

Nu da vi havde fundet ud af hvilket spil vi overordnet ville lave, så skulle vi i gang med at komme med konkrete ideer til et 2D spil. I starten ville vi gerne have fjender og ligne med, men blev hurtigt enige om at dette var for normalt, og ses i alle platform spil. Så vi ville have et spil som udelukket fokuserede på movement. Det var så her hvor vi fik ideen til at slags parkour platform spil.

Vi har nu vores ramme og ideer til spillet, nu skal vi sætte det sammen for at få det til et program som kan bruges. Vi skulle først finde en engine som var nem og gratis at bruge, derfor valgte vi Unity. Unity havde også allerede mange af de helt basale ting med sig.

Efter Unity var downloadet gik det første stykke tid med at lave en karakter som vi ville bruge. I starten var det meget lyse karaktere.



Dette var en af de første karaktere som vi havde lavet. Vi blev dog begge enige om at det ikke skulle være den, og vi valgte derfor at gå med et andet look for hele spillet. Vi følte os inspireret af spillet Limbo, så vi ville lave et mørkt og en smule dystert spil. Derfor blev vores karakter helt sort, med hvide øjne.



Billedet over er den færdige karakter, samt de sprites som blev brugt som at vise karakterens movement.

Efter at havde lavet vores karakter, så skulle vi i gang med at lave en baggrund. Baggrunden blev meget hurtigt lavet. Vi blev enige om at den skulle være mørk, men at der skulle være stjerner og en måne til at give en fornemmelse af lys.



Da vi havde fået næsten alt det graffiske på plads, så skulle vi i gang med at lave selve spillet. Vi startede med at lave hvordan hoppet skulle være. Vi skulle finde den perfekte tid i luften. Derefter skulle vi lave hvordan man bevægede sig. Man kan bruge WASD eller pile tasterne. Vi skulle også finde en fart til vores karakter. Altså hvor hurtigt han bevæger sig. Dette skal også kunne bruges til hoppet. Altså hvor langt karakteren kan hoppe.

Efter at havde lavet movement for karakteren skulle vi i gang med at lave en void. Voiden skulle kunne starte scenen forfra. Uden voiden ville man forsætte lige ned og ville aldrig stoppe. Inden voiden havde vi blokke så man kunne løbe tilbage til start, de er selvfølgelig blevet fjernet nu.

Efter at havde fundet et script til at lave voiden, ville vi i gang med at designe et level. Det var egentligt ret simpelt. Vi tilføjede først en del blokke som man kunne gå på, og gav dem en boks collider. Efter at havde sat selve banen op for levelet, skulle vi lave hvordan banens kamera var. Vi tilføjede et kamera lidt til højre fra midten af skærmen, også lagde vi et script som gjorde at kameraet fulgte vores karakter, så det altid vil se ud som om at vores karakter er helt ude i venstre side af skærmen.

Efter at havde lavet det første level, ville vi i gang med at lave endnu et. Først skulle vi lave et script som fortalte spillet at en scene var ovre, så den kunne skifte til en ny. Vi valgte at det der skulle indikere hvornår et level var slut skulle være et flag. Når karakteren rammer flaget skifter den altså scene. Nu kunne vi lave endnu et level på samme måde som det første.

Da vi nu havde to levels følte vi at vi kunne lave en startskærm. Start skærmen ville vi havde simpel og nem. Vi lavede derfor et skilt med navnet på spillet i baggrunden, en lang linje af blokke og to portaler. Den ene portal fungerer som flagende, den anden lukker programmet.

Efter at havde lavet en starts skærm ville vi også have en slutskærm. Den skulle være næsten identisk med startskærmen, bortset fra at der kun skulle være en portal som lukkede spillet. Efter at havde lavet en slutskærm, mente vi at spillet var færdigt.

# Afprøvning af spillet

Efter at havde prøvet spillet et par gange, så kan man hurtigt klare det. Spillet er faktisk meget sjovt, og vi har haft flere til at prøve det. Til at starte med syntes alle at det var virkeligt svært. Der var mange som skulle have hjælp med at komme gennem level 1. Efter de så havde prøvet level 1 et par gange, kunne de klare den på under 20 sekunder. Level 2 har alle dog stadig lidt svært ved. Dem som fik klaret level 2 brugte omkring 10-15 forsøg på det, de fleste gav dog op, og sagde den var umulig at klare.

Efter selv at havde spillet det igennem en del gange, fandt jeg den hurtigste tid på at klare hele spillet på omkring 40 sekunder. Lige nu mangler der en del mere content før det er et virkelig godt spil. Dette kan dog ret nemt laves.

[(Se gameplay)](https://www.youtube.com/watch?v=wGLO_1hauFA&feature=youtu.be)

# Forbedringer

En ting som helt klart kunne havde været gjort bedre var mere gameplay. Vi havde dog ikke tid til at kunne lave mere end to levels. Hvis man ved hvad man skal gøre, så kan man klare spillet på under et minut. Så flere levels vil helt klart være en forbedring.

Med flere levels ville der også skulle være mere variation i grafikken. Det ville blive ret kedeligt at skulle kigge på den samme baggrund gennem 20 levels. Vi burde også have lavet en menu, så man kunne sætte spillet på pause og afslutte det. Der er to måder at lukke spillet når man er i det. Den ene er på scene 1 ved at gå hele vejen til højre, og gennem portalen med EXIT. Den anden er ved at klare spillet, hvor man så skal gøre det samme. Hvis man gerne vil lukke spillet mens man er i gang bliver man nødt til at bruge ALT + F4, eller lukke det via joblisten.

Et bedre lydtrack eller flere lydtracks havde også været bedre. Igen, 20 levels med den samme lyd ville blive kedeligt.

En bug som der er i spillet er hvis man kommer hoppende højt oppe fra, og lander hvor to blokke mødes, så kan man komme lige i gennem. Vi leder stadig efter en måde at fikse dette på.

Det kunne også være en fordel at kunne gemme sin progress. Hvis man lige har klaret level 1 og gerne vil vente med level 2, og lukker programmet, så skal man starte helt forfra.