

Space Invaders

An android game

Casper Frost Andersen & Christian Møller Strunge

Android Game Introduction

03-06-2019

# Introduktion

Vi har til vores projekt valgt at udvikle et spil der tager udgangspunkt i formatet fra det klassiske spil Space Invaders.

Spillet går ud på, at man som bruger styrer et rumskib, der skal forhindre en armé af rumvæsener, som støt bevæger sig mod rumskibet, i at nå deres mål, og derved sikre universets overlevelse.   
Dette gøres ved, at det brugerstyrede rumskib affyrer projektiler mod fjenderne, og når alle fjender er fjernet, vil en ny bølge af fjender dukke op, indtil alle fjender er udslettet, og universet endnu engang er sikkert.

# Struktur

## Spillets struktur:

Spillets struktur har en lineær tilgang, der kan summeres under følgende trin:

1. Spillet startes, og brugeren ser hovedmenuen.
2. Når spillet startes, vil brugeren blive navigeret til første niveau, hvor fjenderne skal nedkæmpes.
   1. En fjende har ved eliminering en chance for at droppe en guldmønt, der giver ekstra 100 point, hvis rumskibet opsamler denne.
   2. Spillet kan til hver en tid sættes på pause, hvor det blandt andet er muligt at returnere til hovedmenuen, slå musik/lyd til/fra (se evt. forbedringer).
3. Når alle fjender er nedkæmpet, vil sværhedsgrad og point blive sat op.
4. Hvis fjenderne når rumskibet, er spillet slut, og brugeren får mulighed for at starte et nyt spil

## Teknisk struktur:

1. Vi har i vores spil valgt at lade rumskibet være berøringsstyret, da skibets projektilaffyring er kollisionsbestemt.
2. Fjenderne bliver indlæst i rækker, hvilket tillader udviklerne til at determinere antallet af fjender pr. niveau, og/eller udseende af fjende-blokkens udseende.
3. Fjenderne bliver målt individuelt, da der er flere parametre der skal måles på, herunder:
   1. Når en fjende bliver ramt af et projektil, skal kun den ramte fjende elimineres
   2. Når blokken af fjender rammer siden af skærmen, skal den yderste fjende i den pågældende side være den determinerende faktor der afgør, om blokken skal skifte bevægelsesretning.
   3. En bonus-mønt skal kunne falde fra en individuel fjende, hvilket bliver afgjort ved en tilfældig tal i en if-condition.

# Udfordringer

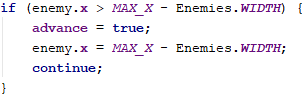
Under udviklingen af spillet stødte vi på nogle udfordringer, som vi vil redegøre for herunder:

## Enemy movement:

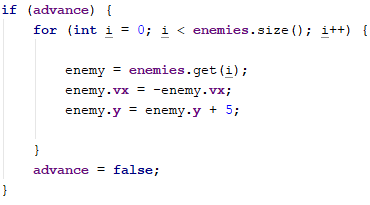
Fjenderne i spillet bliver indlæst i arrays. Vi har fastsat et array af 5 fjender, som der bliver genereret 6 af. Fjenderne skal bevæge sig fra side til side, og når de/den yderste fjender rammer højre eller venstre side af skærmen, skal alle fjenderne skifte bevægelsesretning, og rykke ned på skærmen.

Eftersom fjenderne bliver indlæst i arrays, blev fjenderne først målt på de individuelle arrays – dette betød, at når hvert array mødte kanten af skærmen, var det kun det ene array der skiftede bevægelse. Derfor overlappede fjenderne hinanden, hvilket ikke var hensigtsmæssigt for spillets layout og oplevelse.

Dette løste vi ved at lave en boolean, der, når den blev sat til true, gik ind og hentede hver eneste fjende i et for-loop, og satte hver enkel fjendes bevægelse til at skifte:



*1.1: Condition der måler på, om en fjende rammer højre side af skærmen*



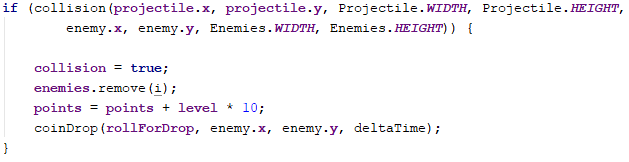
*1.2: Condition der angiver, at hvis advance er sandt, hentes alle fjender i et for-loop, ændrer bevægelsesretningen, rykker ned på y-værdien (vertikalt), og sætter advance til værende false igen.*

## Coin-drop:

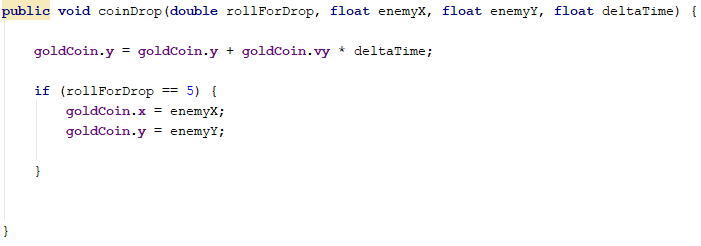
Vi har valgt at implementere en bonus-funktion for spilleren, som kan opsamle en guldmønt, der giver spilleren en point-bonus, der bliver højere for hvert niveau man er på.

En guldmønt skal falde tilfældigt fra en fjende, hver gang en fjende bliver ramt med et projektil. Hertil skal rumskibet opsamle mønten, før pointene bliver tildelt.

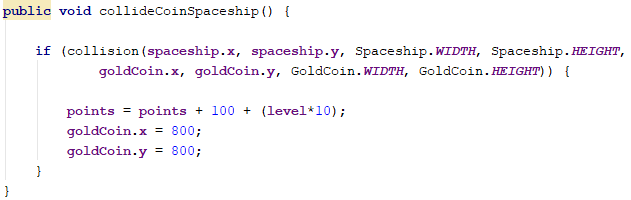
Vi havde en udfordring i kun at få rendered mønten, når projektilet kolliderer med en fjende, og få den til at forsvinde, når den blev opsamlet. Vi har lavet en midlertidig løsning, hvor mønten bliver indlæst uden for skærmen, og bliver sat på en fjendes position, når coin-drop conditionen bliver mødt (se figur 2.3).



*2.1: Et if-statement i vores collideProjectileEnemy() metode. I dette statement angiver vi, at hvis projektilet kolliderer med en fjende, bliver en boolean-condition sat til true (bruges til at lave et nyt projektil), fjenden bliver fjernet fra arrayet, point for fjenden bliver tildelt, og en coinDrop() metode bliver kaldt. Metoden tager imod en random generator (rollForDrop) og fjendens position.*



*2.2: coinDrop() metoden angiver møntens vertikale bevægelse, og hvis random generatoren rammer tallet 5, bliver mønten placeret på fjendens position, som vil falde mod bunden af skærmen.*



*2.3: collideCoinSpaceship() metoden. Når rumskibet og mønten rammer hinanden, bliver spilleren tildelt point ud fra det niveau spilleren er på, og guldmønten bliver sat uden for skærmen.*

# Forbedringer

Til en forbedring af spillet har vi følgende forslag, som vi vil prøve at få implementeret inden eksamen:

## Skudaffyring:

Skud fra skib affyres tidsbaseret i stedet for på kollision.

## Point

En tekst vil blive vist når skibet opsamler en guldmønt (ex. +100).

## New level

En skærm skal vises mellem levels, der angiver, at man har gennemført det nuværende niveau, og prompter for at gå videre til næste niveau.

## Fjendtlige skud

Fjenderne skal kunne affyre skud, for at tilføje en udfordring til spillet.

## Bunkere

I forlængelse af punkt 4, kunne man tilføje bunkere som skjold mod fjendtlige projektiler, som ville være tro mod det originale spil.