

**Academy Profession (AP) Degree in Computer Science**

**TITLE:**

Breakout in Unreal Engine 4

**PROJECT PERIOD:**

GrnXXdatXX,  
December 2015

**PROJECT GROUP:**

XX

**STUDENTS:**

Nichlas Bruun  
Mathias Forsberg  
Bjarne Kristensen

**SUPERVISOR:**

Jonathan

**ABSTRACT:**

Problemformuleringsspørgsmålet?

Hvorfor dette emne er spændende.

Hvad vi har lavet og fundet ud af.

**COPIES:** 1

**REPORT PAGES:** 21

**APPENDIX PAGES:** 1

**TOTAL PAGES:** 22

# Indhold

<b>1</b>	<b>Forord</b>	<b>1</b>
<b>2</b>	<b>Problemformulering</b>	<b>3</b>
<b>3</b>	<b>Introduktion</b>	<b>5</b>
3.1	Unified Process . . . . .	5
3.2	Process . . . . .	6
3.3	Versionsstyring . . . . .	6
3.4	Object Orienteret Analyse og Design . . . . .	6
<b>4</b>	<b>Object Orienteret Analyse</b>	<b>7</b>
4.1	Systemdefinition . . . . .	7
4.2	Funktionstabel . . . . .	7
4.3	Klasse Diagram . . . . .	7
4.4	Eventtabel . . . . .	7
4.5	Use Cases . . . . .	8
<b>5</b>	<b>Object Orienteret Design</b>	<b>13</b>
5.1	Gameplay . . . . .	13
5.2	Grafik . . . . .	13
5.3	Blueprints . . . . .	14
<b>6</b>	<b>Implementering</b>	<b>15</b>
<b>7</b>	<b>Testing</b>	<b>17</b>
<b>8</b>	<b>Reflektion</b>	<b>19</b>
<b>9</b>	<b>Konklusion</b>	<b>21</b>

# Kapitel 1

## Forord



# Kapitel 2

## Problemformulering



# Kapitel 3

## Introduktion

### 3.1 Unified Process

Unified Process forkortes også UP og er en systemudviklingsmetode. Det blev udviklet i slutningen af 90'erne af Ivar Jacobsen, Grady Booch og James Rumbaugh. I Unified Process gøres der brug af iterationer som programmet skal udvikles i, et større projekts iterationer varer typisk 2 til 6 uger. Når man udvikler et stykke software med Unified Process skal man igennem fire faser. Inception fasen hvor man forbereder hvilke ting der skal laves og danner et overblik over de forskellige opgaver der skal løses. De næste fase kaldes Elaboration, i denne fase udvikles de centrale dele af programmet samt de dele der er sværest at lave, da de skal udvikles tidligt i processen så de ikke bliver udviklet under tidspres. Elaboration fasen er længere end Inception fasen, når fasen er slut vil kerneelementerne af programmet være udviklet. Efter Elaboration kommer Construction fasen, her udvikles de elementer i systemet som skal til for at programmet bliver færdigt, samt at programmet testes løbende igennem fasen. Construction fasen består af mindre iterationer, og det er også fasen hvor man begynder at alpha teste programmet. Udover tests bliver der også udført performance tuning og dokumentation i Construction fasen. Sidst men ikke mindst er der Transition fasen, dette er fasen hvor programmet bliver udgivet, Transition fasen kan have flere iterationer hvor programmet får anmeldelser og feedback inden det bliver udgivet.

I processen med at skabe Unreal Breakout har det ikke været muligt at nå alt som Unified Process indebærer, dog har det stadig været muligt at følge det. Inception fasen blev fuldført med objekt orienteret analyse og design, samt udarbejdelsen af et gant chart, som estimere tiden der skal bruges på de forskellige dele af projektet. Der blev også udarbejdet en risikovurdering i inceptionfasen som skulle vurdere de forskellige risici der er ved at lave et større projekt, samt sætte tal på hvor alvorlige

Breakout i Unreal Engine 4

Nichlas Bruun, Mathias Forsberg & Bjarne Kristensen

problemerne der kan opstå er.

I Elaboration fasen blev det lavet én iteration for at få grundelementerne i spillet lavet, iterationen varede i alt 5 dage og resulterede i at produktet blev til et spilbart spil. Det blev i fasen taget højde for hvilke elementer der ville være tidskrævende at programmere og disse elementer blev lavet først i fasen.

Efter Elaboration fasen begyndte Construction fasen. Construction fasen bestod af én kort iteration hvor der blev foretaget en Blackbox test på programmet samt rettet småfejl.

Transition fasen blev ikke taget i brug i processen med at lave Unreal Breakout da produktet ikke skulle leveres til nogen.

## **3.2 Process**

## **3.3 Versionsstyring**

## **3.4 Object Orienteret Analyse og Design**



# Kapitel 4

## Object Orienteret Analyse

### 4.1 Systemdefinition

### 4.2 Funktionstabel

Funktions tabel.

### 4.3 Klasse Diagram

### 4.4 Eventtabel

Event tabel.

Tabel 4.1: My caption

Navn	Kompleksitet	Type
Start Game	simpel	update
Exit Menu	simpel	update
Exit Game	simpel	update
Release Ball	medium	update / compute
Bounce Ball	medium	update / compute
Move Paddle	simpel	update / compute
Break Brick	simpel	update
Change Score	simpel	update / compute
Show Score	simpel	read

Breakout i Unreal Engine 4

Nichlas Bruun, Mathias Forsberg & Bjarne Kristensen

Tabel 4.2: My caption

Events/Klasser	Paddle	Ball	Brick	Menu	Gameworld	Camera	UI
Player clicks on menu obj				X			
Player moves paddle	X				X		
Player releases ball	X	X			X		
Ball collides with object	X	X	X		X		
Score updates		X	X		X		X

## 4.5 Use Cases

### Start Spillet:

#### Scope:

i menuen.

#### Description:

Spilleren trykker på *play*-knappen, med musen, i menuen for at komme til *gameworlden*.

#### Preconditions:

Spilleren skal befinde sig i menuen for at use casen kan startes.

#### Success Guarantee:

Når use casen er opfyldt vil spillet gå fra menuen til *gameworlden*.

#### Main Success Scenario:

Spilleren befinder sig i menuen og klikker med musen på *play*-knappen og *gameworlden* vises frem på skærmen.

#### Extensions:

Befinder spilleren sig allerede i *gameworlden* vil det ikke være muligt at trykke på *play* knappen da den ikke er vises.

### Luk Spillet.

#### Scope:

i menuen.

#### Description:

Breakout i Unreal Engine 4

Nichlas Bruun, Mathias Forsberg & Bjarne Kristensen

Spilleren kan lukke spillet ved at trykke på *exit*-knappen med musen.

**Preconditions:**

Spilleren skal befinde sig i menuen for at use casen kan startes.

**Success Guarantee:**

Når use casen er opfyldt vil spillet blive lukket.

**Main Success Scenario:**

Spilleren befinder sig i menuen og klikker på *exit*-knappen, med musen, hvorefter spillet vil lukke.

**Extensions:**

Befinder spilleren sig i *gameworlden* vil *exit*-knappen ikke være vist og spilleren kan derfor ikke interagere med den.

**Bevægelse af paddle.**

**Scope:**

I *Gameworld*.

**Description:**

Spilleren kan ved hjælp af højre og venstre piletast bevæge *paddlen* til henholdsvis højre og venstre.

**Preconditions:**

Spilleren har startet spillet igennem menuen.

**Success Guarantee:**

Bliver use casen opfyldt korrekt vil *paddlen* blive bevæget til siderne i takt med at spilleren trykker piletasterne ned.

**Main Success Scenario:**

- a. Spilleren trykker højre piletast ned og *paddlen* bevæger sig til højre.
- b. Spilleren trykker venstre piletast ned og *paddlen* bevæger sig til venstre.

**Skyd.**

**Scope:**

I *Gameworld*.

**Description:**

Bolden skydes væk fra *paddlen* ved brug af *space*-knappen, dette kan kun gøre når spillet startes eller efter bolden har været uden for spilbanen.

**Preconditions:**

Spilleren har startet spillet igennem menuen og har ikke skudt endnu, eller bolden har lige været udenfor banen.

**Success Guarantee:**

Bliver use casen opfyldt korrekt vil bolden blive skudt væk fra *paddlen*.

**Main Success Scenario:**

Spilleren har startet spillet og bolden er stadig placeret på *paddlen*, eller bolden har lige været uden for spilbanen. spilleren trykker på *space*-knappen og bolden bliver skudt væk fra *paddlen*.

**Extensions:**

Bolden er blevet skudt væk fra *paddlen* og har ikke været udenfor spilbanen og kan derfor ikke blive skudt fra *paddlen* igen da den ikke er placeret der.

**Gå tilbage til menuen.**

**Scope:**

i *Gameworld*.

**Description:**

Spilleren går tilbage til menuen fra *gameworlden* ved at trykke på *escape*-knappen.

**Preconditions:**

Spilleren har startet spillet og befinder sig i *gameworlden*.

**Success Guarantee:**

Bliver use casen opfyldt korrekt vil spilleren blive vist menuen igen og *gameworlden* vil lukke.

**Main Success Scenario:**

Spilleren har startet spillet og trykker på *escape*-knappen, herefter vil menuen blive vist og *gameworlden* vil stoppe.

**Extensions:**

Befinder spilleren sig allerede i menuen vil *escape*-knappen ikke have nogen effekt.



## Kapitel 5

# Object Orienteret Design

### 5.1 Gameplay

Unreal Breakout er et spil med et bat<sup>1</sup>, en bold<sup>2</sup>, en masse brikker<sup>3</sup>, tre lukkede sidder, og en åben side. Formålet er at skyde bolden op og ødelægge alle brikkerne og sørge for at bolden ikke ryger ud af banen i bunden hvor spillerens bat og den åbne side er. Når man rammer en brik, går den i stykker eller skifter farve, og så får man point tilføjet sin score. Når spillet starter har man 3 forsøg(bolde) og bolden er låst fast på éns bat i midten. Når man trykker på en dertil bestemt knap skyder man bolden afsted, og med nogle andre knapper bevæger man battet fra side til side for at undgå at bolden ryger ud af spilområdet. Hvis bolden ryger ud af spilområdet, mister man et forsøg(bold) og en ny bold bliver låst fast til batet og klar til at blive skudt afsted på ny. Når man har opbrugt alle forsøg og den sidste bold ryger ud af spilområdet, slutter spillet og man kan se sin score. Får man til gengæld alle brikker fjernet vinder man banen og den samme bane kommer igen uden at ændre på antal forsøg man har tilbage eller nulstille éns score, og bolden bliver igen fastlåst til batet.

### 5.2 Grafik

Spillets grafik er fundet på *opengameart.org*, som er en hjemmeside hvor folk lægger grafik op til fri afbenyttelse. Dette betyder at det er uden licens men man kan dog donere penge til de brugere der har lagt grafikken tilgængelig på siden.

Breakout er et gammelt spil udviklet af Atari og derfor er der fundet grafik som er tro mod det klassiske spils grafik. Grafikken ligger nogle spritesheets som er kollager

---

<sup>1</sup>Paddle på engelsk, bliver brugt i vores paddle klasse.

<sup>2</sup>Ball på engelsk, bliver brugt i vores ball klasse.

<sup>3</sup>Bricks på engelsk, bliver brugt i vores bricks klasse.

af billeder som bliver brugt til at skabe de forskellige elementer i spillet. Når spritesheetet er delt op vil de forskellige brikker blive bygget op til et level. Eftersom spilleren rammer brikkerne med bolden vil de skifte farve, farverne indikere hvor mange gange en brik skal rammes for at blive ødelagt. De grafiske elementer er ikke noget der vil blive sat stor fokus på i spillet, da det er spillets gameplay der skal være grundpillen i det.

## 5.3 Blueprints



# Kapitel 6

## Implementering



# Kapitel 7

## Testing

Breakout i Unreal Engine 4

Nichlas Bruun, Mathias Forsberg & Bjarne Kristensen



# Kapitel 8

## Reflektion



## Kapitel 9

## Konklusion

