Forside

Indholdsfortegnelse

[Specialisering – Proces 3](#_Toc368240938)

[Indledning 3](#_Toc368240939)

[Specialiserings Plan 3](#_Toc368240940)

[Opstart på projektet og planlægning 3](#_Toc368240941)

[Concept 3](#_Toc368240942)

[Forklaring af Første koncept som endte i skraldespanden: 3](#_Toc368240943)

[Forklaring af hvorfor det første koncept skrottes: 4](#_Toc368240944)

[Forklaring af Koncept nr. 2: 4](#_Toc368240945)

[Elaboration 5](#_Toc368240946)

[Opbygning i Unity 5](#_Toc368240947)

[AI programmering 5](#_Toc368240948)

[Statemachine 5](#_Toc368240949)

[GUI elementer i Unity 5](#_Toc368240950)

[Specialebeskrivelse 5](#_Toc368240951)

[Resume af forløbet 5](#_Toc368240952)

[Beskrivelse af konkrete problemstillinger 5](#_Toc368240953)

[Vurdering af arbejds – og læringsprocessen 5](#_Toc368240954)

[Refleksion af eget udbytte 5](#_Toc368240955)

[Konklusion 5](#_Toc368240956)

[Bilag 5](#_Toc368240957)

# Specialisering – Proces

## Indledning

Denne rapport skal klarlægge Processen i mit specialiserings projekt på 4. semester. Først beskrives processen omkring specialiseringsplanen, som skulle udføres inden projektet kunne gå i gang. Projektet har jeg delt op i opstart og planlægning, Concept og Elaboration. Opstart og planlægning består af at finde en Design Model til at udvikle spil med, som jeg kunne arbejde ud fra og lave en tilhørende planlægning ud fra den model. I Concept er spillets koncept, design og ide udviklet. Elaboration skal ideen fra Concept udføres

Dertil har jeg så haft nogle delmål i de forskellige faser, som jeg har kunnet følge op på.

## Specialiserings Plan

Inden projektet startede blev der udarbejdet en specialiserings plan som skulle beskrive hvad det var for et arbejde der skulle udføres og klarlægge mine mål for opgaven. Som hoved emne havde jeg at udgive et spil til android devices fra Unity 3D, helt specifikt på Samsung Galaxy S2, som er den android Device som jeg har tilgængelig nu. Som delmål til spillet skulle der laves en AI så det er muligt at gennemføre et singleplayer spil. Et andet delmål var at lave/ designe en god State machine som er mere lig den undervisning vi fik på sidste semester. Næste delmål var at udarbejde en forretnings strategi, i forbindelse med udgivelse af spillet på Google Play med henblik på at senere kunne lave et fritids firma hvor jeg kan udgive spil og tjene lidt penge på det.

## Opstart på projektet og planlægning

Til at udvikle spillet vil jeg gerne have et design værktøj, som jeg kan bruge igen efterfølgende til flere spil eller til videre udvikling af spillets videre forløb. Derfor har jeg søgt sådan et værktøj. Det som jeg har fundet er en game designer på nettet som har lavet en blog omkring game design, som er simpel og nem at forstå. Det er hans model eller template som jeg har taget og vil arbejde ud fra. Bloggen kan findes på: <http://gamedesigntools.blogspot.nl/>

Fra templaten har jeg valgt at bruge Concept, Elaboration og Tuning.

## Concept

### Forklaring af Første koncept som endte i skraldespanden:

Det første koncept jeg lavede var et spil hvor der er 2 spillere som skal spille mod hinanden, enten spiller mod spiller eller spiller mod AI. Hver spiller har et tårn som er spillerens liv eller hitpoints og tårnet er så samtidig modspillerens mål for at vinde spillet. For at kunne opnå det mål, har hver spiller et sæt kanoner som man kan skyde med for at ødelægge modspillerens tårn. Hver spiller skulle kunne vælge en karakter af 3 forskellige slags, altså en avatar for spilleren, som løbende får mere erfaring og åbner for forskellige skills jo højere man stiger i level. På de højere niveauer kan spilleren også vælge flere forskellige kanoner. Spillet skal være tur-baseret og for at spilleren kan bruge sine kanoner eller karakter skills skal han bruge nogle begrænsede ressourcer, som kunne være i form af for eksempel krudt. Meningen med gameplayet er så at spilleren i hver tur, skal tage en strategisk beslutning om hvad for nogle skills eller kanoner der er bedst at bruge ressourcer på, for at have størst chance for at vinde kampen.

Planen med at der skulle være 3 forskellige karakterer, som man kan vælge i mellem, var for det første at variere gameplayet ved at de forskellige karakterer har deres egen unikke skills, som giver forskellig oplevelse af gamplayet og dermed forlænger spillets holdbarhed. Et andet mål med det var at jeg gerne vil lave noget digitalt salg på Google Play, hvor man så kunne købe udstyr til sin karakter og til kanonerne kunne der laves forskellige skins som der også kunne sælges. Dog med henblik på at spillet ikke må være pay to win, eller ikke virke som det. De features som man skal kunne købe er kun visuelle forbedringer af spillerens karakter og nye skins til kanoner.

### Forklaring af hvorfor det første koncept skrottes:

Efter at have stiftet bekendtskab med Unity 3D fandt jeg ud af at der er mange ting som ville være meget tidskrævende i forhold til de mange animationer og grafiske objekter der skal laves til spillet. Og jeg ville ikke kunne nå at få ret meget af spillet færdig til aflevering af specialet. Samtidig kunne jeg se på mine skitser af spillet at det havde en høj kompleksitet, som ville gøre det svært at se på mit target device, Samsung Galaxy S2. det som jeg gjorde for at sikre mig at det var for småt, var at hente et spil på Google Play som havde en lignende kompleksitet. Jeg kunne tydeligt se at spillet var alt for komplekst til det format og man kunne ikke rigtig se hvad der foregik på skærmen. Men ideen ville måske passe godt til tablet størrelse. Derfor har jeg valgt at droppe dette første koncept og lave et mere simpelt koncept, hvor jeg kan nå at få de vigtigste hovedfeatures til spillet færdig.

### Forklaring af Koncept nr. 2:

Koncept nr. 2 holder sig helt det oprindelige terning spil som laves i 3D og hvor spilleren har en kanon og et tårn. Når man så slår mellem 31-36 kan så skyde modstanderens tårn med sin kanon. Når en spillers liv/tårn er nede på nul, har modstanderen vundet. Den opgave der skal løses som går ud over de 2 kurser der hører til specialet, vil så være udvikling af AI modstander, så er singleplayer spil bliver en feature i spillet. Der skal laves stat emachines til at holde styr på de forskellige states i spillet. Med disse mere simple features er det muligt at opnå et færdigt produkt til tiden.

1. Læg regler til spil i bilag

2. forklaring af begreber i bilag.

Forklar proces omkring udvikling af spil design.

## Elaboration

### Opbygning af scener og arkitektur i Unity

Opstart af projektet i Unity. Som udgangspunkt har jeg et spil som er kodet i c# visual studio med windows forms og som er bygget op ved hjælp af MVC med GUI layer, logic layer og modelLayer. Det skal jeg have overført til en arkitektur som kan fungere med unity og hvor ansvaret for at oprette de forskellige objekter bliver styret et centralt sted, så koden er nem at navigere i. dette vil også gøre det nemmere at arbejde videre på spillet i fremtiden, så det er meget nemmere at udvide det. Til det formål har jeg lavet et tomt GameObject i unity som er kaldt GameManager. Denne klasse er kernen i spillet og skal binde det hele sammen. Klassen skal stå for at oprette og have forbindelse til spiller, terninger og statemachine. Samtidig skal den også står for at binde det hele sammen, så det bliver funktionelt. Det vil sige funktioner til at kaste terninger, tilføje spillere, have en reference til en liste af spillere, en reference til en liste af terninger og en kø af spillere som skal have turen.

Mit Scenarie. Selve opbygningen af spillet i unity, består af scener som kan indeholde gameobjecter så som en spiller eller en terning. Det scenarie som jeg gerne vil lave består af 2 scener. Den første scene er en main menu, hvor man kan indtaste sit spiller navn og vælge om man vil spille mod en ven på samme skærm eller spille mod en AI modstander. Når man har indtastet navne til de 2 spillere som der kan være i spillet, kan der trykke på start og den næste scene i spillet starter. Scene nr. 2 er hovedscenen i spillet. I denne scene vises en spilleplade med 6 terninger. I hver sin ende af spillepladen er spilleren repræsenteret i form af et spiller ikon eller billede. Dertil hører spillerens score og en indikation af hvilken spiller der har den nuværnde tur. For at spilleren kan kaste terningerne, er der lavet en knap til netop det formål. Når en af spillerne har vundet spillet, skal der vises en besked der skriver hvilken spiller som har vundet spillet og derefter skal main menuen loades igen. sådan som det er nu er der ikke nogle tilstande eller objekter der skal vedvare efter at et spil er slut. Dog er det meningen at jeg i en senere udgave vil gemme en historik på hver spiller, der viser hvor mange spil de har vundet. Fra hovedmenuen kunne der så laves en knap til en scene der viser en top ti liste over de spillere der har vundet flest spil.

Så ud fra denne scene og objekt opdeling er jeg komet frem til at jeg skal have som sagt en gameManager objekt, spiller objekter, terninge objekter og en statemachine der skal holde styr på alle de forskellige states i spillet. For at kunne overføre spillerens navn fra main menuen til spil scenen, har jeg valgt at oprette Game manageren fra Main menu scenen, når der bliver trykket på start game knappen. For at kunne loade en anden scene i Unity, har jeg søgt i unitys officielle dokumentation og fundet metoden Application.LoadLevel som loader en scene ved at tage et scene navn eller scene index som input og derfra finde den specifikke scene. Men for at det kan virke skal hver enkelt scene i spillet tilføjes til projektets build settings, som der også står i dokumentationen. Dette gøres ved at vælge Filer -> Build Settings i Unity som åbner et vindue hvor et index af de scener som er tilføjet til buildet vises. Hver enkelt scene kan så vælges i projekt mappen og i build settings vælger man add current og nu bliver scenen tilføjet til listen af scnener i buildet. I dokumentationen står der yderligere at alle objekter der tidligere er oprettet eller loaded, vil blive destrueret når man loader en ny scene. Dette vil sige at man skal håndtere dette hvis ikke man ønsker at alle objekter bliver destrueret.

Nu hvor at jeg opretter Game Mangegeren i Main menu og skal videreføre den til hoved spil scenen skal jeg så finde ud hvordan dette gøres, da loadlevel jo som sagt destruerer alle objekter. Til netop dette formål har jeg fundet en anden metode på unitys officielle forum som hedder Object,DontDestroyOnLoad. I beskrivelsen af metoden står der at hvis man kalder denne metode, på et game object eller component vil objektet vedvare med Application.Loadlevel

Skrive noget om processen omkring de klasser der er slettet fra projektet og tilføjet som variabler på GameManager klassen i stedet for og hvorfor ?

Skrive noget om hvad du har fundet ud af med prefabs.

Proces for at finde ud af hvordan jeg adgang til de andre scripts og kalde metoder på dem. Eksempel hvor jeg lavede et forsøg med game\_script instance som singleton printet tablescore og canroll ud fra spillet.

### State machine

### AI programmering

### GUI elementer i Unity

## Specialebeskrivelse

Specialeplan hører til her. Ikke i bilag.

## Resume af forløbet

## Beskrivelse af konkrete problemstillinger

## Vurdering af arbejds – og læringsprocessen

## Refleksion af eget udbytte

## Konklusion

## Bilag