Forside

Indholdsfortegnelse

[Specialisering – Proces 3](#_Toc369724161)

[Indledning 3](#_Toc369724162)

[Specialiserings Plan 3](#_Toc369724163)

[Opstart på projektet og planlægning 3](#_Toc369724164)

[Concept 3](#_Toc369724165)

[Forklaring af Første koncept som endte i skraldespanden: 3](#_Toc369724166)

[Forklaring af hvorfor det første koncept skrottes: 4](#_Toc369724167)

[Forklaring af Koncept nr. 2: 4](#_Toc369724168)

[Elaboration 5](#_Toc369724169)

[Opbygning af scener og arkitektur i Unity 5](#_Toc369724170)

[GUI elementer i Unity 6](#_Toc369724171)

[Resume af forløbet 6](#_Toc369724172)

[Beskrivelse af konkrete problemstillinger 6](#_Toc369724173)

[Vurdering af arbejds – og læringsprocessen 6](#_Toc369724174)

[Refleksion af eget udbytte 6](#_Toc369724175)

[Konklusion 6](#_Toc369724176)

Bilag:

B1 - Specialiserings plan

B2 – Spilleregler

# 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 på projektet i Unity med 3D objekter

Som en lille indledning til opstarten af spillet i Unity, vil jeg lige forklare hvad det er for nogle objekter og funktioner, som spillet skal bestå af. Dette er for at vise hvad det er jeg skal udvikle. Spillet skal have en start menu, hvor man kan se 2 bokse, som hører til hver sin spiller. I hver boks kan man indtaste sit spiller navn. Når man har gjort det kan man trykke start game, som starter spillet. Spillet består af en spilleplade med 6 terninger placeret i midten. I hver ende af spillepladen er spillerne placeret. På hver spiller er et spiller ikon med hver sin farve, så man kan kende forskel. Hver spiller har en score og et navn så man se hvor mange point man har og sin spillers navn. Ved terningerne er der en knap til at kaste terningerne med. Når teringerne bliver kastet bliver samlet en score spillepladen som bliver vist på en spilleplade score. Denne score skal så trækkes fra spillerens point alt efter hvad score de lander på ved slutningen af turen. Når en af spillerne rammer nul er spillet slut og man bliver smidt tilbage til start menuen. For yderligere oplysninger omkring spillereglerne se bilag B2 – SpilleRegler.

Når man har cirka ingen erfaring med Unity, kan det være svært at vide hvor man skal starte. Men som udgangs punkt har jeg det bedst med at lære nye ting ved at bare at gå i gang med at arbejde med det og forsøge sig frem. Erfaringer hænger bedre fast når jeg bruger dem i stedet for at lave en masse teoretisk research først, også gå i gang med arbejdet bagefter. Nogle gange kan det selvfølgelig være nødvendigt at slå nogle teoretiske ting op, men det tager jeg så løbene som jeg får brug for det. Så derfor har jeg startet med at oprette en scene i Unity som skal fungere som spillepladen i spillet. Med det viden jeg havde om Unity på dette tidspunkt, vidste jeg at man kunne finde og hente 3D modeller og sætte dem ind i Unity. Så jeg gik i gang med at søge på nettet efter modeller på blandt andet www.turbosquid.com og Unitys egen Asset store. Efter have fundet de modeller som jeg mente at jeg skulle bruge, begyndte jeg at sætte dem ind i Unity. Det gik sådan set nemt nok da man kunne importere direkte til Unity fra websiderne af. Nu havde jeg så fundet en spilleplade i form af en tønde, 6 terninger og 2 Menneske modeller, en kanon og et tårn. Det som jeg så troede at jeg skulle i gang med var lige at lave et par animationer til de forskellige modeller. For eks. Var planen at jeg ville lave så når man kastede terningerne, skulle de flyve lidt op i luften, rotere i luften og ud fra en random værdi mellem 1-6 skulle terningen så lande med den værdi op ad. Når en spiller gav modspilleren minus point ville jeg gerne have at en kanon skød modspillerens tårn. For at kunne lave disse animationer vil de skulle laves i et eksternt animations program, som kan være dyre at anskaffe sig. Eller også skal man bruge Unitys fysik til at manipulere objekter med. Uanset hvad kunne jeg godt se at det ville være alt for tidskrævende med den tidsramme jeg har til rådighed. Samtidig er det mere arbejde for en grafiker at lave den slags. Så jeg bestemte mig for at jeg ikke ville bruge tiden på det. Dertil kan det også tilføjes at de modeller som man finder ikke nødvendigvis har alle de ting der skal bruges for de for eksempel lave et terning kast.

Hvad gør jeg nu ? for at kunne nå at lave noget funktionalitet af spillet i Unity, var jeg nød til at komme op med ny plan, som matchede den tidsramme der var til projektet. Så jeg besluttede mig for at lave det meste af spillet i GUI elementer, så hver terning for eks. Er en GUI button med et billede af en terning på. Og spilleren skulle også repræsenteres af GUI elementer med textlabel til score, navn og et billede til spiller ikon. Det eneste der skal være en 3D model er spillepladen. Hvor spilleren og terningerne vises på. Med den nye plan var det så på tide at sætte det i værk.

### Opsætning af spillet med GUI elementer

Efter fjernelsen af alle 3D modellerne, kunne jeg gå i gang med at lave GUI elementer i stedet for. Heldigvis er Unitys Dokumentation omkring GUI elementer godt beskrevet og der er tilmed et nogle simple tutorials med de helt basale elementer som man hurtigt kan komme i gang. Efter at have læst dokumentationen var det nemt at finde ud hvad for nogle elementer jeg skulle have til de forskellige objekter i spillet. Til terningerne har jeg fundet et billede som skal vises på en knap. For ikke at hardcode stien til billederne af terningerne. Søgte jeg i Unitys dokumetation fandt en Resources klasse som netop kan håndtere dette. Jeg fandt ud at man skal oprette en Resources mappe i sin Assets mappe, i den mappe hvor projektet er oprettet. Når det er gjort kan man lægge sin billed filer i den mappe også i Unity kan man bruge metoden Resources.Load(ressource navn) hvor man giver den navnet på recourcen som input. Dermed returneres den ressource så til unity, i mit tilfælde et billede af en terning.

### Praktisk Problem med opsætning af GUI elementer

I forbindelse med opsætning af de forskellige GUI elememter stødte jeg en i et problem som kostede lidt ekstra tid. Når man sætter en GUI.Box op for eks. Skal den have et Rect objekt med x,y, længde og bredde. Altså et rektangel med x og y som nulpunkter og længde og bredde på rektanglet. det gav et problem da jeg havde lavet et build af mit spil som var på en skærm størrelse 800x600 og det build som kører i editoren kører med den skærmstørrelse som editoren kører på, i mit tilfælde 1920x1080. Det resulterede i at build jeg havde lavet var mange af mine GUI elementer ikke at se. Jeg fandt ud af fordi jeg lige skulle teste mit spil i et build hvor jeg valgte en mindre opløsning end den den der kørte i editoren. Så jeg lavede et nyt build med en større opløsning også kunne jeg se alle mine GUI elementer igen. Men problemet var ikke løst, forspillet skulle jo gerne kunne køre på flere opløsninger. Så igen tilbage til Unitys forum og finde ud af hvad man gør for få en relativ skærmplacering. Mission succes unity har en metode som returnerer skærmens længde og bredde. Ved at bruge disse metoder kunne jeg placere GUI elementerne relativt til den skærm buildet kører på. Metoderne er Screen.Length og Screen:height. Også har jeg divideret dem med 2 for at lægge alle elementerne ud fra præcis i midten af skærmen, på den måde vil det altid være centeret i midten af skærmen på alle skærm størrelser.

### Scene problem med at have forbindelse til de forskellige scripts.

Opbygningen af de forskellige objekter i spillet har jeg lavet ved at tilføje tomme GameObjekter, som jeg har kaldt for eks. Player og dice også oprettede jeg scripts som jeg så kunne tilføje på de GameObjekter. Skriv kort om hvad du fandt ud med prefabs.

* Scene problem med at have forbindelse til de forskellige scripts.
* Startet med at lave en scene hvor jeg bare kastede GameObjekter på
* Fundet ud at jeg vil lave en gameManager og hvorfor
* Arbejdet med gameManger
* Tilføjelse af flere scripts til game manageren. Terninger og spillere
* Få noget GUI til at vise de objekter som gamemanageren har oprettet.
* Yderligere GUI med terniger, tablescore og 30 states
* Arbejdet lidt med trello i slutfasen.

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 kommet frem til at jeg som sagt skal have et 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 Manageren 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 ?

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.

### GUI elementer i Unity

## Resume af forløbet

## Beskrivelse af konkrete problemstillinger

## Vurdering af arbejds – og læringsprocessen

## Refleksion af eget udbytte

## Konklusion