Indhold

[Indledning 3](#_Toc389104185)

[Hvad er et spil 3](#_Toc389104186)

[Teknologianalyse 4](#_Toc389104187)

[Indledning 4](#_Toc389104188)

[Valg af spilmotor 4](#_Toc389104189)

[CryEngine 4](#_Toc389104190)

[Unity3D 5](#_Toc389104191)

[GameMaker 6](#_Toc389104192)

[Unreal 6](#_Toc389104193)

[Byg eget framework 7](#_Toc389104194)

[Konklusion 7](#_Toc389104195)

[Valg af udviklingsmiljø 8](#_Toc389104196)

[Valg af billedredigeringsprogram 8](#_Toc389104197)

[Lydeffekter og musik 9](#_Toc389104198)

[Versionsstyring 9](#_Toc389104199)

[Unity Team Server 9](#_Toc389104200)

[GitHub 9](#_Toc389104201)

[Bitbucket 9](#_Toc389104202)

[Dropbox 10](#_Toc389104203)

[Konklusion 10](#_Toc389104204)

[Markedsanalyse 10](#_Toc389104205)

[Indledning 10](#_Toc389104206)

[Analyse af spilmarkedet kontra Anden Underholdning 12](#_Toc389104207)

[Analyse af spilplatform 14](#_Toc389104208)

[Kvalitetsforventninger 14](#_Toc389104209)

[Konklusion 15](#_Toc389104210)

[Segmentanalyse 16](#_Toc389104211)

[Salgsplatforme – AppStore mv. 17](#_Toc389104212)

[Målgruppeanalyse 18](#_Toc389104213)

[Konklusion 20](#_Toc389104214)

[Indtægtsmetoder 20](#_Toc389104215)

[Direkte Salg 20](#_Toc389104216)

[Reklamer 20](#_Toc389104217)

[Free-To-Play / Mikrotransaktioner 20](#_Toc389104218)

[Abonnementsordning 21](#_Toc389104219)

[Konklusion 21](#_Toc389104220)

[Virksomhedsanalyse 22](#_Toc389104221)

[Virksomhedens interne ressourcer 22](#_Toc389104222)

[Håndgribelige ressourcer: 22](#_Toc389104223)

[Uhåndgribelige ressourcer 22](#_Toc389104224)

[Virksomhedens evner 23](#_Toc389104225)

[Kernekompetence 23](#_Toc389104226)

[Kerneydelser 23](#_Toc389104227)

[Virksomhedens eksterne miljø 23](#_Toc389104228)

[SWOT 25](#_Toc389104229)

[Strategiplan 25](#_Toc389104230)

[Målsætning & Handlingsplan 26](#_Toc389104231)

[Opstartsbudget 26](#_Toc389104232)

[Finansiering 28](#_Toc389104233)

[Virksomhedsform 28](#_Toc389104234)

[Projektvalg 29](#_Toc389104235)

[Projektbudget 29](#_Toc389104236)

[Design 31](#_Toc389104237)

[Udviklingsproces 31](#_Toc389104238)

[Scrum 31](#_Toc389104239)

[Extreme Programming 32](#_Toc389104240)

[Sprint rapport 32](#_Toc389104241)

[Sprint 1 32](#_Toc389104242)

[Konklusion 33](#_Toc389104243)

[Sprint 2 34](#_Toc389104244)

[Sprint 3 34](#_Toc389104245)

[Problemformulering 34](#_Toc389104246)

[Udvikling 34](#_Toc389104247)

[Teststrategi 34](#_Toc389104248)

[Scope og overblik 34](#_Toc389104249)

[Tilgang til test 34](#_Toc389104250)

[Testmiljø 35](#_Toc389104251)

[Testværktøjer 35](#_Toc389104252)

[Udgivelseskontrol 35](#_Toc389104253)

[Risikoanalyse 35](#_Toc389104254)

[Godkendelse af test 35](#_Toc389104255)

[Implementering 36](#_Toc389104256)

[Skærmstørrelse 36](#_Toc389104257)

[Videre support 36](#_Toc389104258)

[Perspektivering 36](#_Toc389104259)

[Konklusion 36](#_Toc389104260)

[Bilag 36](#_Toc389104261)

[Ordforklaring 36](#_Toc389104262)

[Problemformulering 38](#_Toc389104263)

[Hovedspørgsmål 38](#_Toc389104264)

[Delspørgsmål 38](#_Toc389104265)

[Projektafgrænsing 38](#_Toc389104266)

[IT-forundersøgelse 39](#_Toc389104267)

[Nuværende situation 39](#_Toc389104268)

# Indledning

Formålet med denne rapport er at undersøge, hvad der indgår i processen at udvikle et software produkt og udgive det. Rapporten forsøger at tage udgangspunkt i et virksomhedsperspektiv, hvor udviklingen og produktet skal danne grundlaget for en bæredygtig virksomhed.

Denne rapport indeholder derfor en række emner, som er vurderet relevant for dette projekt. Det først, som rapporten dykker ned i er en teknologianalyse, som redegør for de tekniske værktøjer mv., der er nødvendigt i dette projekt.   
Det næste er en markedsanalyse, hvis formål er at undersøge, hvordan man succesfuldt føre en virksomhed. Analysen indeholder statistikker omkring salgsplatforme, demografi mv. Derudover indeholder den samtidig en virksomhedsanalyse, hvor der bliver redegjort for de omkostninger en nystartet virksomhed kan forvente.

Den næste del af rapporten forsøger at redegøre for den proces, produktet er gennemgået. I denne del bliver der gjort redefor designfasen, hvor produktidéen mv. blev til. Endvidere gøres der rede for, hvordan produktet under og efter udviklingen skal testes. Herefter kommer den tekniske dokumentation af det endelige produkt. Her gives der kodeeksempler fra det endelige produkt for at give et indblik i funktionaliteten i spillet.

Rapporten indeholder ydermere en gennemgang af den arbejdsudvikling, som projektet er gennemgået. Sidst afsluttes der med en perspektivering samt konklusion, hvor der samles op på produktet og projektet som hele.

### Hvad er et spil

Spil er et ganske vidt begreb, som kun bliver bredere dag for dag. I det store hele kan man argumentere at såfremt et spil kræver brugerens input for at nå sit potentiale, så er det et spil. Hvis det ikke var tilfældet kunne der lige så vel være tale om en bog, en film eller en anden slags fortælling. Det interaktive element er det som adskiller et spil fra disse andre ting.   
Samtidigt har et spil også elementer fra snart sagt alle de mere traditionelle kunstformer: film, bøger, musik, billedkunst mm. Spiludvikling kræver en kombination af alle disse kunstformer, og dertil at brugeren aktivt involveres.

Det betyder altså, at for at kunne producere et spil skal der samtidigt produceres en lang række grafiske og lydmæssige kunstværker. Dem skal vi så søge at knytte sammen via kode, til et produkt som er underholdende og attraktivt for så mange mennesker som muligt.

For effektivt at kunne producere vores første spil, skal vi bruge forskellige typer af værktøj, og mandskab. Eksempelvis skal det undersøges hvilke billedredigerings- og modelleringsværktøj vi kan gøre brug af. Der skal findes et passende udviklingsværktøj som kan bruges til at samle de forskellige komponenter. Vi skal finde en måde at producere de lyd-assets som vi har behov for, og så skal det undersøges hvilke andre typer teknologier og værktøj vi har behov for.

Spiludvikling kan overordnet set deles op i to kategorier, den ene side er den kunstneriske, som just beskrevet, mens den anden handler om godt købmandsskab. Det er irrelevant om vi producerer verdens bedste spil, hvis vi ikke samtidigt kan formå at få det solgt med profit, derfor skal der laves en markedsanalyse, så vi med en vis rimelighed kan konstatere at der rent faktisk findes en køber til vores produkt, og at det spil som vi producerer kan ramme et segment i den samlede køberskare.  
I løbet af denne proces er det også væsentligt at få undersøgt hvilke salgskanaler det er muligt at gøre brug af, samt hvilken, eller hvilke, monetiseringsstrategier som det kunne være relevant at gøre brug af.

For at kunne arbejde effektivt og målrettet skal der udvælges en passende udviklingsproces, og der skal udarbejdes designdokumenter således at alle de projektinvolverede trækker i den samme retning.

Efter gennemgangen af det ovennævnte bliver det nødvendigt at tage et skridt tilbage, og vurdere om virksomheden besidder de nødvendige ressourcer for at kunne påbegynde spilproduktion.

# Teknologianalyse

## Indledning

For at kunne bygge et spil skal vi i princippet bruge tre forskellige typer assets: Grafik, lyd og kode, samt den mest passende teknologi til at samle delene til et spil. Dertil kommer så hvad vi behov for i forbindelse med versionsstyring og testhardware.

For at kunne identificere de teknologier der matcher vores behov, gennemgås der i dette kapitel en række alternativer på hvert enkelt felt.

## Valg af spilmotor

Det er af afgørende betydning for virksomhedens fremtid, at vi vælger det rigtige udviklingsværktøj fra starten. At skifte fra en spilmotor til en anden er ensbetydende med et dyk i firmaets akkumulerede KnowHow, og en periode med indlæring af nye systemer, som ellers kunne være brugt på udvikling.

Der findes naturligvis en lang række andre, både proprietære og kommercielle spilmotorer, men for at bevare overskueligheden fokuseres der udelukkende på de følgende da de repræsenterer mest udbredte og veldokumenterede.

### CryEngine

|  |  |
| --- | --- |
| CryEngine 4 |  |
| Versioner: | Gratis til uddannelse og non-kommerciel brug. Ellers individuelt aftalt licens. Abonnementsordning på vej: 50kr/md |
| Unikke Features: | Bliver ofte fremhævet for dens håndtering af tung grafik. |
| Scripting sprog: | LUA |
| Understøttede platforme: | PC, Xbox 360, Xbox One, PlayStation®3, PlayStation®4 or Wii U. |
| Fokus: | Tung grafik, store teams, store projekter, PC og konsol |

CryEngine udvikles af tyske CryTek, og har i udgangspunktet kun været brugt af store og veletablerede firmaer. Grunden dertil skal formentlig findes i, at et spilfirma skal have en vis størrelse for at kunne retfærdigøre udgiften til en licens. Derudover har CryEngine bygget sit ry på at levere de, subjektivt, flotteste spil i branchen. Det har værdi for AAA-udviklere, men er nok mindre væsentligt for et lille spilhus.

Det er ikke til at sige hvad en pro-licens helt præcist koster, da priserne varierer efter størrelsen på det firma der henvender sig. Der er dog tegn der tyder i retningen af, at CryTek skal have 20% af bruttofortjenesten ved licenser givet til indie-udviklere.  
Dog har CryTek for nyligt annonceret, at de vil implementere en abonnementsplan til ca. 50kr. om måneden, og uden at kræve royalties for udgivne titler. Det er endnu ikke officielt annonceret om der bliver adgang til kildekoden, eller om der er andre restriktioner der skal tages højde for. Umiddelbart gør det dog at det med et slag er blevet den billigste af de undersøgte spilmotorer.  
Der er et aktivt fællesskab omkring CryEngine, og der er adgang til forskelligt undervisningsmateriale. Men af de spilmotorer som der her tages udgangspunkt i, er det den der virker til at have det mindste fællesskab. Konkret betyder det, at udviklingen af spil vil tage længere tid, og medføre flere frustrationer, end det ellers havde været tilfældet.

### Unity3D

|  |  |
| --- | --- |
| Unity3D |  |
| Versioner: | Gratis udgave (inkl. Gratis iOS og Android moduler)  Pro-version til 8000kr. (abb. 400kr/md)  Pro-iOS: +8000kr. (abb. +400kr/md)  Pro-Android: +8000kr. (abb. +300/md) |
| Unikke Features: | Massiv brugerbase giver svar på alt |
| Scripting sprog: | C#, UnityScript (JS mod), Boo |
| Understøttede platforme: | PC, Mac, Linux, Xbox 360, Xbox One, PlayStation®3, PlayStation®4 or Wii U, WebGL, iOS, Android, Windows Phone, BlackBerry. |
| Fokus: | Generel spiludvikling, mindre teams, og mindre projekter. All-round devices |

Indtil tidligere I år var der reelt kun én spiller på markedet for virksomheder af vores slags, og det var Unity. Derfor er der også et massivt community omkring Unity-motoren, og da der er rigtigt mange nybegyndere betyder det at mængden besvarede spørgsmål omkring brugen af spilmotoren, og spiludvikling i det hele taget er massiv. Dertil kommer, at Unity selv er meget aktive med udviklingen af læringsmateriale. Det er også et absolut plus, at det er relativt let at portere et spil fra en platform til en anden, og Unity har pt. klassebiblioteker til alle de platforme som er relevante for virksomheden.  
Med en pris på nul kroner og nul øre for gratis-versionen, der indeholder langt de fleste af systemets komponenter, er der intet at sige til at mange nybegyndere har taget Unity til sig. Paradoksalt nok betyder det til gengæld, at det kan være svært at finde decideret læringsmateriale om mere avancerede emner.

Der er ikke mulighed for at redigere direkte i Unity´s kildekode, men man kan tilføje sine egne komponenter, og af den vej bøje systemet til sin vilje.

Prisen på pro-versionen er der ikke så meget at sige til, det koster 8100 for en licens til den cyklus Unity er i, og den ejer man så til evig tid. Hvis man vil opgradere til den efterfølgende cyklus koster det det halve. Nu, og indtil Unity 5 lanceres kan man pre-order version 5, og få adgang til Unity 4 pro indtil da.  
Hvis man vil udvikle til smartdevices koster det yderligere 8100 for iOS pro og igen 8100 for Android Pro.   
Man får altså ikke adgang til f.eks. iOS pro features selvom man har Unity pro.   
Det er der mange meninger om, men det betyder at det hurtigt kan eskalere i prisen. Især fordi det der er tale om her gælder én licens. Hvis et team er på 4-5 mand, hvilket er typisk, skal man altså gange op.  
Unity har også en abonnementsordning hvor man kan leje hver enkelt del for 400 kr./md.   
Benytter man sig af denne løsning så er det værd at bemærke, at man binder man sig for et år af gangen.

### GameMaker

|  |  |
| --- | --- |
| GameMaker Studio |  |
| Versioner: | Flere versioner, med forskellig pris efter features  Fra gratis til 4300kr. |
| Unikke Features: | Lav et spil helt uden programmering |
| Scripting sprog: | GML (Game Maker Language) |
| Understøttede platforme: | PC, Mac, Android, HTML5, iOS, Linux, Windows Phone, Tizen |
| Fokus: | For den absolutte nybegynder, Små projekter, PC, Web, mobil. |

GameMaker udvikles af YoYo Games, er mest at betragte som nybegynderens første værktøj. Som bruger kan man udvikle et spil uden at skulle skrive en eneste linie kode. Af den grund er det vidt udbredt blandt hobbyister. Alligevel er der eksempler på succesfulde spil der er udviklet med GameMaker, senest Hotline Miami, Spelunky og Risk of Rain.   
Omvendt er der ingen større titler på deres showcase liste, hvilket indikerer at der er nogle begrænsninger i den pipeline som systemet er bygget op over.

Selvom det er muligt helt at undgå kode, så er der mulighed for selv at scripte ved brug af det javascript-lignende GML-sprog.  
  
Selvom GameMaker kan fås fra 0kr, så kræver det Master udgaven til 4300 kr, for at kunne eksportere til iOS og Android.

I lighed med Unity er der et stort fællesskab omkring systemet, hvilket væsentligt bidrager til at stabilisere læringskurven.

Unreal Engine

|  |  |
| --- | --- |
| Unreal Engine 4 |  |
| Versioner: | Individuel prissætning. Abonnementsordning  100kr/md + 5% af brutto |
| Unikke Features: | Fuld adgang til kildekoden. |
| Scripting sprog: | C++ |
| Understøttede platforme: | PC, Mac, iOS, Android.  Konsoller er ikke omfattet af abonnementsordningen. |
| Fokus: | Store teams, store projekter, PC og konsol |

Unreal Engine udvikles af Epic Games, og har navn efter det klassiske FPS Unreal fra 1998. UE har været anvendt i en meget lang række spil siden da, bla. Mass Effect og Gears of War-spillene. Selv om den oprindeligt primært var til brug for FPS-spil, viser listen at det har været muligt at udvide den og lave tilpasninger så den har kunnet bruges til spil i alle genrer.

Indtil nu har spilstudier kunnet forhandle sig til en licens hos Epic. Der er ikke angivet faste priser, så man kan kun spekulere i hvad det har kostet.  
Siden 2009 har man kunnet hente en gratis ”light”-udgave af Unreal Engine 3, kaldet UDK (Unreal Developer Kit). Hvis man udviklede og udgav noget ved brug af den, har det kostet et engangsgebyr på 535kr. samt 25% af bruttofortjenesten.

Ligesom tilfældet var med CryEngine, er det tidligere på året blevet annonceret, at Unreal Engine 4 bliver tilknyttet en abonnementsordning, hvor man for godt 100kr/md kan få en fuldt opdateret version af spilmotoren. Dog skal man også aflevere 5% af bruttofortjenesten på ens udgivelser.   
Hvis man benytter sig af denne model kan man udgive til Mac, PC, iOS og Android, men ikke konsoller. I givet fald skal man igen tage kontakt til Epic for en handel.

Epic stiller dokumentation og læringsvideoer til rådighed, og der er også et dedikeret fællesskab hvor man kan finde løsninger. Og, i modsætning til de andre systemer er der her mulighed for at redigere direkte i motorens kildekode.

### Byg eget framework

Den sidste mulighed er naturligvis, at firmaet går i gang med at udvikle sit eget framework. Det ville medføre 100% fleksibilitet i forhold til firmaets fremtidige udvikling. Samtidigt kan en god spilmotor generere massiv indtægt til virksomheden, hvis den kan sælges eller lejes ud til andre udviklere.  
Men det er en massiv opgave at give sig i kast med. Det kræver et højt teknisk niveau, og det kræver kapital der kan understøtte virksomheden indtil den er klar til brug. Det kan tage flere år at nå til det punkt.

### Konklusion

Selvom både CryEngine og Unreal Engine 4 er billigere i anskaffelse end Unity (For den fulde version), så er konklusionen at firmaet bør anvende netop Unity. Det skyldes at fælleskabet omkring dette system er enormt, og beviseligt gerne deler ud af dets erfaringer. Der er ikke det spørgsmål som ikke er blevet stillet, og besvaret, og det alene er nærmest nok til at vælge Unity frem for nogen andre.

Dertil kommer, at alle de involverede i virksomheden allerede har stiftet et indgående bekendtskab med Unity. Det er ikke nogen triviel opgave at blive omskolet til en ny spilmotor, så det trækker også væsentligt i retning af unity.

De sidste par måneders røre, som er forårsaget af at både Epic og CryTek har annonceret at deres systemer bliver tilgængeligt i et prisleje hvor selv de mindste kan være med, betyder dog at det ikke er sikkert at virksomheden skal binde sig alt for meget til en platform.  
Netop den type overvejelser er en vægtig grund for at udvikle sit eget system, om det så er et framework, eller en decideret spilmotor. Hvis man har muligheden for det, så har man også selv al kontrollen.  
Det er dog ikke en realistisk mulighed for dette firma, da ingen af de involverede har erfaring med den type udvikling. Selv i bedste fald ville der gå årevis før en egenudviklet spilmotor ville kunne bruges til seriøs udvikling, og i den tid ville der ikke være nogen indtægter ej heller nogen mulighed for virksomheden for at positionere sig på markedet.

Angående GameMaker, så er det måske nok et system som vi bør afprøve, da det muligvis kan være tidssparende i forhold til udvikling af prototyper. Som primær spilmotor virker det dog afskrækkende, at der ikke findes eksempler på store produktioner i deres Showroom. Det er ikke i virksomhedens bedste interesse at binde sig op på en motor hvorfra vi ikke kan skalere til større produktioner.

Altså bør virksomheden udvikle det første spil med Unity, og i øvrigt forsøge at holde sig til gratisversionen frem til udgivelsen, og så i den afsluttende fase benytte den 30-dages gratis pro-version.

## Valg af udviklingsmiljø

Da vi udvikler i Unity3D har vi mulighed for at gøre brug af tre forskellige scripting sprog: C#, JavaScript/UnityScript og Boo. Af de tre er vi bedst påklædt til at bruge C#, hvilket betyder at vi skal anvende det IDE som bedst understøtter C#.

Det betyder enten Visual Studio som er Microsoft´s proprietære system, og som vi skal bruge en licens for lovligt at måtte anvende. Alternativt kan vi benytte os af deres gratis version Visual Express, som er en ”lite”-version af den samlede pakke.

Som en tredje løsning kan vi benytte os af MonoDevelop, som er en open source implementering af .Net. MonoDevelop kommer med Unity3D installationen, og er supporteret af Unity.

Selv om Visual Studio afgjort er det stærkeste af de to IDE´er, så er det ikke afgørende for os, da vi faktisk kun har behov for at benytte os af en autocompletion-funktionalitet, samt en syntax-highlighter. Disse features findes i alle tre IDE´er, hvorfor vi uden videre kan benyttes os af MonoDevelop eller Visual Express.

Grunden til at det forholder sig sådan, er at det er Unity der skal kompile vores kode. Der er ingen executable som VS kan benyttes sig af, og derfor kan vi heller ikke benytte os af de indbyggede debug-værktøjer som ellers ville gøre VS til den oplagte kandidat.

## Valg af billedredigeringsprogram

For at kunne bygge spil, skal vi bruge en billededitor. Vores grafiker er vant til at arbejde med Blender til 3D-modeller og animationer, og Gimp til billedredigering, og da det samtidigt er gratis-produkter, så det er oplagt at fortsætte med at benytte disse værktøjer.  
  
Ikke desto mindre er vi opmærksomme på, at vi der er alternativer som vi kan få brug for på længere sigt.   
Den åbenlyse kandidat i forhold til billedredigering er Photoshop, som er en standard i industrien. Ud over Photoshop producerer Adobe også en række andre interessante produkter, som f.eks. Illustrator, som vi kunne have gavn af.  
  
Hvad angår 3D-modellering, så er det den samme situation, Blender er godt nok til vores umiddelbare behov, men mere professionelle produkter som f.eks. Maya og 3D Studio Max har flere og bedre funktioner end Blender, og derfor kan det på den længere bane være interessant at lave en mere konkret analyse.

Både i forhold til Adobe produkterne og de nævnte modelleringsprogrammer gælder det dog, at de er i et prisniveau som gør er de pt. er uinteressante for os.

For nuværende er det dog ikke interessant, fordi behovet for den type produkter ikke eksisterer.

## Lydeffekter og musik

Der findes en lang række betalte, og gratis, systemer som kan benyttes i forbindelse med lydoptagelser til både effekter og musikkompositioner. For at kunne holde os inden for projektets scope, vil vi ikke lave analyse af de forskellige. Idet vi mener at kunne benytte os af gratis, og betalte lydeffekter og musikstykker. Der findes en lang række forskellige webshops der forhandler den type varer.

## Versionsstyring

For at sikre, at vi har fuld kontrol over vores produktion, og ikke er i farezonen for at miste data, er det afgørende at vi har versionskontrol på vores projekt. Der findes en lang række mulighed

### Unity Team Server

For det første har Unity selv et ”Team-License”-produkt der kan integreres direkte i Unity. Det koster 2700 kr. pr person, og der er ikke nogen Sky-funktion inkluderet, hvilket vil sige at vi selv skal have en server sat op.  
Til gengæld er Unity produktet skræddersyet til at håndtere både den type af filer som Unity producerer, og den størrelse de kan forvente at komme i.

### GitHub

Alternativt kan vi betale os fra en privat konto hos f.eks. GitHub. De kan fås fra 40 kr. om måneden. Til gengæld begrænser vi os til projekter der er mindre en 1GB, og filer som ikke overstiger 100MB.  
Ligeledes anbefaler GitHub at man bruger et andet redskab til ”mellemregninger” af tungere typer filer, som f.eks. 3D-modeller.  
Strengt taget behøvede vi slet ikke betale for en GitHub konto, men gratis-kontierne er offentligt tilgængelige. Det er ikke noget problem i første omgang, men det er usmart hvis vi skal arbejde med en tredjepart, eller generelt vil holde et projekt lidt tættere til kroppen.

### Bitbucket

Bitbucket er et alternativ til GitHub. I modsætning til GitHub er det muligt at få private repositories gratis for op til 5 brugere. Samtidigt kan vi have et ubegrænset antal repositories, hvilket vil sige at så længe firmaet har max 5 personer tilknyttet der skal bruge adgang, så kan vi få gratis opbevaring. I øvrigt er der integration med Jira indbygget, hvilket er relevant at tage i betragtning når vi kommer til et tidspunkt hvor vi får brug for et reelt bugtracker-værktøj.  
I forhold til projekt- og filstørrelser, så er det i store træk de samme regler der gør sig gældende her, som hos GitHub.

### Dropbox

Vi kan også benytte os af en ressource som Dropbox. Dels er det let at dele materiale imellem projektdeltagere, men det er også muligt at sætte et versionsstyringssystem op direkte i en mappe. Igen kan man benytte sig af gratisløsningen hvor hver bruger har 2GB (I udgangspunktet), men det er ikke stort bedre end med GitHub.  
Den optimale dropbox løsning ville koste 400 kr./md.

### Konklusion

Der er flere alternativer, som sætte et system op. Men i forhold til opstarten af virksomheden så er det at skyde langt over målet. Hvis vi selv satte en server op, så ville det også medføre at vi selv skulle stå for vedligehold af den, håndtere back-ups og opgraderinger og meget mere. Vi er et spilfirma, ikke en hostingservice. Eller sagt på en anden måde, det er ikke et område vi vil bruge vores tid og energi på, det skal bare virke.

Med alt dette i mente har vi besluttet at bruge en kombination af Dropbox og GitHub til vores første projekt. Alle midlertidige assets gemmer vi i vores, gratis, Dropbox mapper, og de bliver først importeret til vort Unity projekt når de er klar til det. Selve Unity-projektet er decideret versionsstyret og ligger på en gratis-konto på GitHub. Det er ikke specielt relevant lige nu, at det er offentligt tilgængeligt, for det er svært at forestille sig, at nogen skulle være interesserede i det. Hvis det bliver vurderet som nødvendigt, kan vi slette det Git Repository som projektet ligger i, når spillet er klar til udgivelse.

Af selvsamme grund kunne Bitbucket være et relevant produkt til det næste projekt. Når vi har valgt det fra i denne omgang, så skyldes det at vi har erfaring med at bruge GitHub, og det er ikke et område vi er interesserede i at bruge energi på i denne ombæring.  
Reglerne angående fil- og projektstørrelse behøver vi i første omgang ikke bekymre os om, da det er er urimeligt at forestille sig vores projekter overstige 1GB. Når vi kommer til lidt større projekter, så er det et emne der kan blive taget op igen.

# Markedsanalyse

## Indledning

I dette afsnit vil vi analysere og danne os et overblik over underholdningsindustrien som hele, hvilket skal danne grundlag for resten af markedsanalysen, da det er relevant at kende til den overordnede udvikling af industrien. I analysen bliver der gennemgået statistikker omkring spilplatforme, salgsplatforme, segmenter samt målgruppeanalyse, hvor hver enkelt leder op til den næste og har fokus på det den retning, som projektet skal tage.

Dette afsnit afsluttes med en konklusion, hvor der bliver gjort overvejelser over, hvordan virksomheden bedst udnytter disse informationer.

## Analyse af spilmarkedet kontra Anden Underholdning

Inden for underholdningsbranchen findes der et væld af forskellige underbrancher og nicher, som er i indirekte konkurrence med spilindustrien. Det er vigtigt at redegøre for, hvilke industrier, som har fremgang, samt hvilke industrier, som svinder ind, da dette teoretisk set kan have indflydelse på virksomhedens fremtid. Dog betyder størrelsen af industrien, at projektets succes ikke afhænger af industriens umiddelbare fremtid.

Underholdningsindustrien er i stor fremgang og er vokset med over 50% det sidste årti og er vokset fra $449 mia. i 1998 til $745 mia. i 2010.

*“The movie industry cleared 180 billion US dollars in 2009. Sports earned 63 billion, computer games 54 billion, live performances 35 billion, and music 7 billion. Global entertainment from all sources is expected to reach 1.4 trillion dollars by 2015.”   
– prweb.com*

Som det ses i figuren er spilindustrien en relativ lille del af underholdningsindustrien, men ser man på spilindustriens alder i forhold til de andres har den haft enorm fremgang på utrolig kort tid.

*“No other sector has experienced the same explosive growth as the computer and video game industry. Our creative publishers and talented workforce continue to accelerate advancement and pioneer new products that push boundaries and unlock entertainment experiences. These innovations in turn drive enhanced player connectivity, fuel demand for products, and encourage the progression of an expanding and diversified consumer base. ”   
- Michael D. Gallagher, president and CEO, Entertainment Software Association*

[[1]](#footnote-1)

Som ses af denne graf har spilindustrien haft stor positiv udvikling, specielt de seneste år, og vi forventer ikke den aftager i fremtiden. På dette grundlag er det derfor naturligt at konkludere, at computerspilsbranchen klart bør overvejes at føre virksomhed i.

## Analyse af spilplatform

Specielt i spilindustrien er der en markant større forskel på de forskellige platforme sammenlignet med f.eks. TV-industrien, vor det ofte er irrelevant om produktet ses på mobiltelefonen, på computeren eller på tv’et. Spilindustriens platforme har enorm betydning for udviklingen og produktionen af nye spiltitler af flere grunde. Først og fremmest er det en teknisk udfordring, da de forskellige platforme er udviklet forskelligt, og det kræver derfor ressourcer fra udviklernes side at tilpasse en spiltitel til en ny platform. Dette gælder og ressourcestyring i form af computerkraft, som spænder vidt fra små håndholdte enheder til optimerede Pc’er. Derudover har platformene vidt forskellige interaktionsmuligheder fra touch skærm til joystick til mus og tastatur.

Spilindustrien kan opdeles i 5 forskellige hovedplatforme, som ses i grafen herunder:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Spilindustriens indtægt på verdensplan (i mio. USD) | | | | |
| Segment | 2012 | 2013 | 2014 | 2015 |
| Spilkonsoller | 37.400 | 44.288 | 49.375 | 55.049 |
| Håndholdte konsoller | 17.756 | 18.064 | 15.079 | 12.399 |
| Mobile spil | 9.280 | 13.208 | 17.146 | 22.009 |
| PC-spil | 14.437 | 17.722 | 20.015 | 21.601 |
| Total | 78.872 | 93.282 | 101.615 | 111.057 |

Det, der kan læses ud fra denne tabel, er, at alle områder har stabil fremgang med undtagelse af håndholdte konsoller (Nintendo DS, PS VITA mv.), som forudsiges at miste markedsandel. Det skyldes sandsynligvis den enorme fremgang af mobile spil, som i væsentlig grad opfylder det samme behov som de håndholdte konsoller. Dette hænger tæt sammen med den enorme udbredelse af smartphones i verdenen, som er steget fra 1,13 mia. telefoner i 2012 til 1,75 mia. telefoner i 2014 – næsten en fordobling på få år. Skulle man derfor vælge en branche ud fra disse tal står valget imellem spilkonsoller og mobile spil, hvor begge har haft stor fremgang de seneste år. Af de to er spilkonsolmarkedet det største, hvorimod den mobile platform er mere udbredt i verden. Samtidigt kan der være store omkostninger forbundet med at udvikle til konsollerne, da de respektive platformsudviklere har omfattende procedurer der skal overholdes ved udgivelse til deres produkter.

Det er også værd at notere sig, at stigningen i PC-spil er kraftigt stigende som slutningen af den forgående konsol-generation kommer nærmere. Figur XXX herover viser tydeligt denne tendens. For at kunne udgive spil til PC, kræves der i princippet intet ud over et produkt.

Som nystartet udvikler er det derfor lettere at udgive til mobile platforme eller PC.

## Kvalitetsforventninger

Hvis vi ud fra det ovenstående holder fast i, at vores marked er enten PC eller mobil, så er det vigtigt at forstå forskellen blandt de to målgrupper. Overordnet vil vi benyttede de gængse termer ”Gamer” og ”Casual”.

En gamer er en person (m/k) som virkelig holder af spil, en som har brugt en masse tid på at spille spil og diskutere dem med andre, og i et større eller mindre omfang følger med i hvad der sker i branchen, og som derfor helt naturligt opbygger nogle højere krav til kvaliteten af det spil som han eller hun bruger sin tid på. Spil der henvender sig til denne type spiller kan være af en næsten uanet kompleksitet og scope.

En casual derimod, er en som godt kan bruge lidt tid i ny og næ på at spille, men som ikke er interesseret i at bruge energi på at sætte sig ind i komplekse spil. Spil der produceres til casual-spillere er derfor typisk genkendelige på deres simple design, der gerne er modelleret omkring en enkelt, eller kun ganske få game mechanics. Oplagte eksempler er Angry Birds eller FarmVille.

Både for gamer og casual arketyperne, gælder det at der er mange gyldne middelveje; altså brugere som er lidt af begge dele. For at kunne bruge opdelingen til noget må vi derfor se på hvilke spil der henvender sig til hvilken gruppe, og hvor de spil befinder sig.

Candy Crush Saga, som fås på mobile devices, og som Facebook-app på PC, er det mest udbredte casual spil for øjeblikket. På Facebook har det små 60 mio. daglige brugere, og til de mobile enheder er det downloadet over 500 mio. gange. Men det findes ikke på Steam, GOG.com eller andre dedikerede PC salgskanaler. Det samme scenarie gør sig gældende for mange andre spil af samme type. Det er en kraftig indikator for at den type brugere som de spil har, er meget lettere tilgængelige på de nævnte platforme.

Omvendt findes de mere ”hardcore” spil ikke i samme omfang på mobile enheder (eller Facebook), hvilket delvis skyldes, at maskinkraften på smartphones ikke er den samme som på en PC, men også at markedet for den type spil ikke er der i samme grad som på de dedikerede PC og konsol salgskanaler.  
Der findes naturligvis spil som gør sig gældende på alle platforme, i casual-kategorien kan man f.eks nævne Plants vs. Zombies, som startede på mobilplatformene, men nu også findes på PC og konsol som standalone klient. Mens Minecraft der startede som et PC-spil nu findes på Xbox og i en mobiludgave også.

Det er absolut væsentligt, at vores spil opfylder de forventningskrav som slutbrugeren har, og fordi casual-spillere stiller lavere krav til spillets kompleksitet, og assets, så vil vi som et lille firma have bedre muligheder for at opfylde de krav, end hvis vi valgte at udvikle til et mere kræsent publikum. For at kunne gøre det, må vi se på eksempler på spil som ligner det vi laver og som minimum sikre at vores produkt er af samme kvalitet.

## Konklusion

Baseret på det ovenstående, og en vurdering af virksomhedens ressourcer, finder vi, at det vil være mest formålstjenstligt at udvikle vores første spil til en mobilplatform.

Derfor fokuseres der udelukkende på det mobile marked i resten af afsnittet for at undgå irrelevant markedsanalyse på brancher, som ikke er virksomhedens fokus.

## Segmentanalyse

Når der er lagt fokus på en platform at udgive til, er der brug for at segmentere branchen i typer af apps, da der i dette tilfælde er fokus på mobilbranchen. I dette projekt er der valgt at lave et spil, og der kigges derfor på, hvilke genre af spil, som dominerer.

Som ses ud af figuren over er de dominerende genrer sociale spil samt ”hjernevrider”-spil, som ligger dominerer med 70% af markedet. Derefter er det Action-spil mv., som tager tredjepladsen af populære genrer med 13%, hvor MMO har små 4% af markedet. Ud fra dette burde fokus lægge på sociale spil eller puzzles mv. Kigger man dog på tal fra 2012 har sociale spil haft fremgang, hvorimod puzzles mv. er faldet fra 47% til 35%.

Baseret på tallene her kan vores beslutning om at lave et runner-spil sagtens forsvares, da det falder i kategorien Action-spil. Det er ikke genren med den højeste procentsats, men der er mange mennesker der spiller den type spil.

## Salgsplatforme – AppStore mv.

Da markedet for mobile spil er i stærk fremgang er det relevant at undersøge, hvilke muligheder, der er, for at udgive sit produkt og ikke mindst optimering af salg. Først og fremmest er det relevant at vide, hvilke platforme, som er populære og som har det største marked. Dernæst skal der kigges nærmere på værdien af disse platforme samt hvilke genre af apps, som bliver købt.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Smartphone App Statistics** | **iPhone** | **Android** | **Blackberry** | **Windows** |
| Total app downloads | 27 mia. | 29 mia. | 2,4 mia. | 4,1 mia. |
| Percent of app users who have never paid more than $1 for an app | 45 % | 62 % | 63 % | 58 % |
| Average number of downloaded apps per phone | 88 | 68 | 49 | 57 |
| Total number of apps in store | 905,000 | 850,000 | 130,000 | 220,000 |
| Total app store revenue in 2013 | $6,400 mio. | $1,200 mio. | $550 mio. | $950 mio. |

Som ses i ovenstående tabel er iPhone- og Android-markedet stærkt dominerende i antal downloadede apps og det vil derfor være naturligt at fokusere på disse som det primære mål for udgivelsen af produktet, da der er størst chance for succes. Dog skal omsætningen af Windows apps bemærkes, da det relative lave antal downloads har givet en omsætning næsten tilsvarende til Androids omsætning. Derfor kunne det overvejes at distribuere igennem omsætning per udgivet app er højere.

Det er dog vigtigt at notere sig, at figuren herover ikke tager højde for apps med IAP, men kun direkte salg.

**DET HER SKAL UNDERSØGES NÆRMERE!!!! -> fordeling af gratis apps på de forskellige stores.**

**I det hele taget skal emnet det uddybes noget mere.**

Af de fire markeder som er repræsenteret herover, er det umiddelbart Apple og Googles markedspladser som er mest valide. Vores primære valg af salgskanal bør være en af de to, men den endelige beslutning kommer også til at bero på andre faktorer, eksempelvis test hardware mv.

## Målgruppeanalyse

Når produktet er lagt fast på genre og spillestil er det vigtigt at redegøre for den målgruppe, man håber at nå, for at være i stand til at optimere markedsføringen.

For at få et overblik over det generelle billede af spilindustrien, kigges der på et nogle grundlæggende demografiske tal:

Den første figur viser kønsfordelingen. Klassiske fordomme omkring, at det kun er mænd, som spiller computerspil er for længst blevet modbevist, og det er derfor vigtigt at erkende det kvindelige marked.

Den midterste figur viser aldersfordelingen af computerspillere verden over, og som det ses af figuren er spillerne relativt unge, da 80% er 35 år eller derunder. Det næste, som bør bemærkes, er det relativt lille segment af unge under 18. Da computerspil let kan fortolkes som en form for legetøj, er det derfor naturligt at forestille sig et ungt publikum. Men da gennemsnitsalderen for computerspillere er 26 år, bør man indstille sig på at rette sin markedsføring mod det modne publikum.

Dette leder til den sidste figur, som viser computerspillernes daglige beskæftigelse. Som det ses er 50% i et fuldtidsarbejde, sammenlagt 30% er enten fuld- eller deltidsstuderende, og kun 11% er arbejdsløse. Mens kun 2% er pensionerede, hvilket i øvrigt hænger fornuftigt sammen med aldersfordelingen.



Kigger man på figuren over, kan man læse at arkadespil og ”Runner”-spil lægger i det neutrale felt, hvor der ikke er specifikke kønsdominans. Dog hælder genrene mere til den kvindelige side frem for den mandlige. Samtidig lægger begge genre i den yngre del af denne figur, men stadig omkring gennemsnitsalderen for computerspillere. Dog er der et spænd imellem de to, hvor gennemsnitsalderen for arkadespillere er 10 år ældre end ”runner”-genren. Det er derfor vigtigt, at have en klar idé om, hvilken genre produktet tilhører når det skal designes.

## Konklusion

Når man efterhånden har fået redegjort for de tal og statistikker, som man føler er relevante for ens virksomhed / produkt, så bør det være muligt at danne sig et billede af det marked, man bevæger sig ind på.

I det forgående afsnit har vi redegjort for spilmarkedet som hele. Vi har forsøgt at redegøre for markedets udvikling og om hvorvidt industrien er profitabel også i fremtiden. Baseret på den analyse konkluderer vi, at industrien er velvoksende med en god fremtid og på det grundlag er det oplagt at starte et spilfirma.

Dernæst blev der kigget på underbrancherne, hvor de forskellige tekniske platforme dikterer opdelingen. Her konkluderes det, at udvikling til den mobile platform er et naturligt valg for virksomheden. Sidst bliver der redegjort for det mobile marked i form af segmentanalyse af de forskellige spilgenre, målgruppeanalyse samt hvilke distributionsplatforme, som er relevante.

## Indtægtsmetoder

### Direkte Salg

Direkte salg er den klassiske metode, hvor man sælger produktet direkte til forbrugeren for et fast beløb.  
Det direkte salg gør sig godt på PC og konsol-markedet, hvor forbrugeren altid har været vant til at det er sådan tingene gøres.  
Denne metode kan kombineres med de andre, men det kan være svært at forklare køberen, at han skal betale mere efterfølgende. En genre hvor dette har været udbredt er i sværvægtsklassen af MMO-spil. World og Warcraft har f.eks. kørt med betaling for spillet selv, og så en månedlig abonnementspris derefter.

### Reklamer

I stedet for at tage penge direkte fra produktet, kan man indsætte reklamer hvorfra man kan generere en kontinuerlig indtægt. For at denne metode kan være rentabel kræves det, at der er mange mennesker der bruger ens spil, da summen man får for enten vist reklame, eller klikket-på reklame er meget lille.  
Derfor bruges reklame-metoden ofte på mobil- og webbaserede platforme, da spillet så er tilgængeligt for et meget stort marked. Samtidigt er reklamefinansierede spil meget ofte gratis i anskaffelse, hvilket gør det let for kunden at anskaffe sig spillet.  
Reklame-metoden bruges ofte i ”demo”-udgaver af spil. I den forbindelse er håbet så, at spilleren skal blive så glad for spillet, at han køber den fulde version. Eksempler på dette er f.eks. Angry Birds Lite og Wordfeud.

### Free-To-Play / Mikrotransaktioner

Efter direkte salg er Free-To-Play er måske mest udbredte indtjeningsstrategi for spil i øjeblikket. Ved Free-To-Play forståes det, at selve spillet er gratis for brugeren at anskaffe sig. I stedet genereres virksomhedens indtjening ved at spilleren kan købe sig til visse goder via mikrotransaktioner. Ofte er der tale om en virtuel møntfod, som spilleren så kan bruge i spillets butikker.  
Metoden blev først anvendt i tidlige onlinespil, og har siden bredt sig til alle typer, og størrelser, af spil.  
Et af de mest kendte eksempler er vel Facebookspillet Farmville, hvor spillet er gratis, men hvor der så er rig mulighed for at spendere efterfølgende.  
Denne metode har givet anledning til en del debat, da der er mange måder at organisere et Free-to-Play-spil på, og nogle af dem er decideret forbrugerfjendske.   
Der er en række negative eksempler, men omvendt kan det også være en måde hvorpå virksomheden kan knytte endnu tættere bånd til sine kunder, hvis det gøres rigtigt. Se f.eks. Path of Exile, Loadout, League of Legends, World of Tanks

### Abonnementsordning

En ordning hvor forbrugeren betaler et månedlig beløb for adgang til spillet. Det har tidligere været den optimale måde at holde de store multiplayer-spil i live på, men tiden er tilsyneladende ved at løbe fra den model. I hvert fald ses det igen og igen hvordan nye MMO´er starter med en abonnementsmodel, men kort tid efter opgiver ævred og går Free-To-Play. Se F.eks. Star Wars – The Old Republic og Age of Conan.

### Konklusion

Som virksomhed kan man godt vælge at specialisere sig i en specifik indtjeningsmetode, og det kan også godt vise sig at være fornuftigt. Især i forbindelse med Free-To-Play, som dækker over en myriade af muligheder.  
Der er over de seneste par år blevet skabt et konsensus omkring hvilke metoder der matcher hvilke typer spil, og det er vores holdning, at vi er bedst tjent med at følge med strømmen på det område.  
I vores tilfælde vil det sige enten direkte salg, gratis med reklamer eller gratis med mikrotransaktioner.  
Til dette projekt mener vi, at det kan betale sig udvikle en Free-To-Play model, hvor spillet i sig selv er gratis, men kommer med   
Det passer med vores nuværende situation hvor ingen kender os, og vi ikke har nogle tidligere produkter at bryste os af. Der er det vigtigt, at vi kan nå så mange mennesker som overhovedet muligt.   
  
På lidt længere sigt kunne det være rigtigt interessant at kigge på Free-To-Play modeller. For en udvikler er der nærmest kun positive sider ved den metode:  
1. Man kan give produktet til kunden gratis, hvilket formindsker kundens barriere for køb.  
2. Spillet kan blive ved med at tjene penge lang tid efter kunden har fået det mellem hænderne.  
3. Det er en strategi der kan benyttes til alle typer spil, og på de fleste platforme.  
   
  
Et sidste argument imod direkte salg, og for Free-To-Play og reklamebaserede modeller, er at visse platforme har store problemer med piratkopiering. Et kopieret eksemplar af et produkt der kun har direkte salg som strategi, er mistet indtjening for udvikleren. Men hvis indtjeningsmodellen på produktet er af en karakter hvor det er lige meget om produktet i sig selv er gratis, så er piratkopiering udelukkende en fordel for udvikleren.

Det er ikke relevant for os at spekulere i abonnementsmodeller på nuværende tidspunkt. Den generelle tendens i industrien er, at gå væk fra den model og i retning af modeller med mikrotransaktioner. Hvis det en dag vender, så forbrugerne hellere vil betale premium priser for det fulde produkt, så er det en diskussion der kan tages op igen.

Udgangspunktet for virksomheden er, at der er to programmører og en grafiker til rådighed for virksomheden på deltid. Alle er i princippet ulønnede, da evt. løn hænger direkte sammen med den omsætning som kan genereres. Ingen af de tre iværksættere har før udgivet et spil, eller noget andet produkt. Ligeledes er der heller ingen erfaring med iværksætteri blandt opstarterne.  
Alle involverede har arbejdet på spilprojekter på fritidsbasis i et par år, og har derfor en grundviden om hvordan et spil skal skrues sammen. Desuden er alle involverede vokset op med at spille spil af alle mulige slags, på alle tænkelige platforme, hvilket giver os en solid platform hvorpå vi kan diskutere forskellige designløsninger, indtjeningsstrategier osv.  
Der er ingen reel organisationsstruktur i virksomheden, da udgangspunktet er at grundlæggerne er lige stillet, men med er der en grovopdeling af opgaver hvor vores grafiker er hovedansvarlig for at tegne vores projekters vision, mens det er programmørernes primære opgave at sikre at den vision kommer til live.

I forhold til opgaver som idegenerering, design, historiefortælling mv. Altså de aspekter som ikke direkte omhandler produktionen af assets, så betragter vi det som en fælles opgave.  
Det økonomiske udgangspunkt for virksomheden er, at vi ikke har egentlig kapital at starte op for.

Derfor er vi for nuværende tvunget til at arbejde deltid, arbejde hjemmefra, og arbejde med gratis-værktøj. Der er heller intet markedsføringsbudget, eller mulighed for at deltage i messer af nogen art.

## Virksomhedsanalyse

I det følgende kapitel vil vi kigge på virksomhedens interne ressourcer, de eksterne markedsvilkår og vores position. Formålet er at afdække virksomhedens styrker og svagheder, de eksterne trusler og vores muligheder på markedet.

### Virksomhedens interne ressourcer

I dette afsnit beskrives de interne ressourcer som virksomheden har til rådighed. Det er værd at bemærke, at der i visse sammenhænge, og her tænkes der på de uhåndgribelige ressourcer og virksomhedens evner, mere er tale om hensigtserklæringer end reelt tilgængelige ressourcer.

### Håndgribelige ressourcer:

* Almindelige windows PC´er
* 1 Samsung Galaxy Tab 10.1 2 (4.4.2)
* 1 Samsung S4 telefon (4.4.2)
* 1 HTC One (4.4.2)
* Unity3D 4.3.4 (gratis version)
* Visual Studio 2013
* Gimp 2
* Blender 2.69

### Uhåndgribelige ressourcer

* Virksomhedens brand – I skrivende stund har virksomheden ikke noget brand, men det er en af de ting som vi aktivt kan arbejde på at udvikle allerede fra dag et. Et velrenommeret brand kan hjælpe os til mersalg i forhold til almindelige kunder. Men det kan også åbne døre for os i forbindelse med virksomhedens vækstmuligheder.
* Virksomhedens kreative kraft – Dette er uden tvivl vores vigtigste kort, vi er hvem vi er, og det kan ikke kopieres af konkurrerende firmaer. Vores personlighed og vores holdning skal skinne klart igennem i alt hvad vi laver.

### Virksomhedens evner

* Godt branchekendskab – Vi kender de forskellige kundesegmenter som vi kan komme i berøring med. Derfor har vi et godt udgangspunkt for at levere et produkt som matcher det segment som et givent projekt specifikt henvender sig til. Ligeledes kan vi også med stor sikkerhed vælge den rette prisstrategi til spillet. Vi ved hvad der er kvalitetsstandarden for hver type platform og for de kundegrupper der anvender disse platforme. Dermed kan vi kan placere vores produkter på et passende niveau på en mere effektiv måde.
* Et solidt kendskab til spil af alle slags – Vi kender vores spil, og i kraft af den erfaring er vi i stand til at træffe fornuftige designvalg igennem alle faser af udviklingen.
* Består af uddannede programmører og grafiker.
* Fleksibilitet – Her er der igen mere tale om en hensigtserklæring. Vi vil være fleksible i vores tankegang og vores processer, således at vi hele tiden kan udvikle os som fagpersonel, og som firma. Vi er ikke bange for at op eller nedskalere firmaet efter behov. På samme måde er vi heller ikke bange for at tage projekter af en type som ikke umiddelbart følger den fra starten lagte plan, hvis det giver mening for virksomheden.

### Kernekompetence

Spiludvikling med Unity3D spilmotoren.   
Stram projektplanlægning   
Virksomhedens kreativitet

### Kerneydelser

Spiludvikling – Egen udvikling  
Spiludvikling – Eksterne kontrakter

### Virksomhedens eksterne miljø

* Konkurrenter

Virksomhedens umiddelbare konkurrenter er alle andre mobil udviklere i hele verden. Det gælder både de som har penge bag sig, og dem der, som os, starter fra den absolutte bund.

I Danmark er det oplagt at nævne Tactile Entertainment (Chasing Yello) som en direkte konkurrent, men de markedspladser som vi skal sælge på, er globale. I det lys er konkurrencen langt skarpere fra firmaer som Rovio (Angry Birds), ZeptoLab UK (Cut the Rope) eller PopCap (Plants Vs. Zombies).

Derudover vil der være specifikke konkurrenter i forhold til det enkelte projekt, forstået på den måde at vores direkte konkurrent ikke er den samme hvis vi udvikler et PC-spil som hvis det er et mobil-spil.

* Teknologisk udvikling

Udbredelsen af mobile enheder der kan afspille spil, samt den modning spilmotorer som Unity3D har gennemgået i de senere år, gør det muligt for små virksomheder at udgive spil af høj kvalitet for et meget lavt budget.

Den næste ting der kommer, bliver formentlig udbredelsen af Virtual Reality enheder. At Facebook for ganske nyligt betalte 2 mia. USD for startup-virksomheden Occulus Rift, og Sony´s nyligt annoncerede Project Morpheus signalerer klart at det er et område som tages alvorligt. Pt. Er det kun muligt at købe en prototype af Occulus VR, og Morpheus er slet ikke tilgængeligt. Men det er afgjort et område som vi som nyopstartet virksomhed kunne drage fordel af at holde øje med.

* Industriforhold

Det samlede billede af industrien generelt er noget mudret, fordi spil kommer i mange forskellige størrelser, og dermed prisklasser. I den tunge ende finder man firmaer som Ubisoft og Blizzard, og generelt er der kun en begrænset mængde udviklere der kan håndtere udviklingen af såkaldte AAA-spil. Jo mere man nedskalerer et spil, jo flere virksomheder findes der som kan håndtere opgaven, og det er ikke unormalt at virksomheder i størrelsen 1-5 mand producerer mobil eller pc-spil.

* Kundegrundlag

Samlet set er kundegrundlaget massivt. Der er solgt mere end 700 mio Apple devices (tlf og pads), og på Android siden er tallet formentligt omkring en milliard. Selv om disse tal dækker over udgåede modeller, ure, ipods o.lign. typer udstyr der ikke er direkte relevante for salg af apps, så er det dog rimeligt at antage, at der er et tilstrækkeligt kundegrundlag til at starte en virksomhed der udvikler mobilspil.

Dertil kommer at vi også kan udvikle til mere modne platforme såsom: PC, Playstation, Xbox, WiiU eller Webbrowsere. På PC findes der en række markedspladser hvoraf de mest oplagt for os er Steam og GoG. For de andre gælder det at hver platform har sin egen markedsplads, og sit eget kundesegment.

* Overordnet analyse af det eksterne miljø

Der er ingen tvivl om at konkurrencen på spilmarkedet er knivskarp. De samme teknologiske fremskridt der gør det muligt for os at starte op, har gjort det samme for tusindvis af andre udviklere, og de har alle sammen mere erfaring end os.   
På den anden side er kundegrundlaget massivt, og konstant stigende. Det er også et marked hvor de der kan udskille sig fra mængden og/eller levere konstante kvalitetsprodukter har alle tiders mulighed for at slå igennem.

### SWOT

|  |  |
| --- | --- |
| **Styrker** | **Svagheder** |
| * Fleksibilitet * Virksomhedens kreative kraft * Teknisk knowhow * God indsigt i spilbranchen generelt | * Mangel på kapital * Manglende erfaring med virksomhedsdrift * Svagt netværk i branchen |
| **Muligheder** | **Trusler** |
| * Stort marked med mange selvstændige segmenter * Lav indgangsbarriere * Mange konkurrenter leverer lavkvalitetsprodukter | * Vores produkter kan meget let drukne i mængden |

### Strategiplan

Hvor er virksomheden om et år

Om et år har virksomheden udgivet mindst et spil og beskæftiger en mand på fuld tid. De(t) udgivne spil udgør virksomhedens porteføjle, og med dem i hånden vil vi forsøge at finde investorer, således at fremtidige projekter kan udvikles med fuldt fokus. Derudover har vi fået alle dele af håndværket ind under huden fra præ-produktionen, og til udgivelse.

Hvor er virksomheden om 2 år

Om to år beskæftiger virksomheden minimum ejerne på fuld tid. Meningen er, at virksomheden skal skaleres på en organisk måde, hvor vi altid er på et niveau hvor vi ikke tager projekter som er for store. Vi bygger os selv op, og vi tager projekter i den størrelse der passer for konstant udvikling.

Hvad det for en position som virksomheden ønsker at indtage:

I første omgang er det virksomhedens mål, at nå en størrelse hvor det kan fuldtidsbeskæftige ejerne. Vi vil ikke lægge os fast på om vi er et firma der kun udvikler til App-markedet, eller PC, eller Konsol. Vores fleksibilitet er vigtig. Derfor har vi også valgt Unity3d som udviklingsplatform, da den netop understøtter mange formater, uden at vi i virksomheden skal starte forfra med nye værktøjer for hvert projekt.

Det er ikke vores holdning, at der er et reelt loft over hvor meget en spilvirksomhed kan vækste, hvis den først kommer ud over hobbyniveau.

### Målsætning & Handlingsplan

* Om 3 mdr. Skal der være en feature-complete prototype klar Dvs. alt funktionalitet skal være på plads.
* Om 4 mdr. Skal virksomheden have firmahjemmeside og pressemateriale klar.
* Om 6 mdr. Skal spillet være i beta-tilstand. Dvs. færdigt, men med plads til fejlrettelser
* Om senest 8 mdr. skal virksomheden have selvudgivet første spil.
* Om senest 8 mdr. skal virksomheden have kontakt til investorer.
* Om et år skal virksomheden være fuldtidsbeskæftigelse for mindst en af ejerne.
* Om et år skal virksomheden have udviklet og udgivet mindst to spil, eller porteret første spil til flere platforme.

Det er afgørende, at virksomheden hurtigst muligt får udgivet et spil, af en kvalitet som gør at det kan anvendes til at tiltrække investorer. Efter udgivelse på en platform skal det vurderes om der er basis for, at lave opdateringer til spillet, eller om det skal porteres til flere platforme. Hvorvidt vi skal det ene eller det andet, beror på om spillet er en succes eller fiasko. Såfremt der er værdi i videreudvikling på spillet, kan målet om to udgivne spil på et år fraviges.  
Uanset om spillet har været en kommerciel succes eller ej, er det vores udgangspunkt for at tage kontakt til potentielle investorer. Det er en proces som vi påbegynder umiddelbart efter udgivelsen. Udover at kunne referere til det spil vi allerede har lavet, er det også væsentligt, at vi har et pitch klart til det næste projekt.

Uanset om vi kan få hul igennem til en eller flere investorer, så er det væsentligt at vi så hurtigt som muligt får etableret en firmahjemmeside. Fordi spil er en global virksomhed, er det vigtigt at have et fast anker som vi kan henvise til hvor alle og enhver kan se at vi er et firma der hele tiden arbejder aktivt imod færdiggørelsen af det næste projekt.   
Samtidigt er det af største vigtighed at få påbegyndt opbyggelsen af et brand og et ry så hurtigt som det overhovedet kan lade sig gøre. Både for at kunne være genkendelige over for kunderne, men især for den interne branchegenkendelse. Som i så mange andre sammenhænge handler det om at have de rigtige kontakter, og for at få det skal vi ud og være synlige på så mange måder som vi kan.   
Dvs. Tilstedeværelse på sociale medier som f.eks. facebook, Twitter og Youtube, samt deltagelse på relevante internet fora.

## Opstartsbudget

Selv om vi forsøger at holde vores budget på så lavt et niveau som muligt, så er der stadig nogle omkostninger som vi er nødt til at forholde os til. Når vi vil udvikle vores spil til iOS, så kræver det at vi indkøber nogle Apple produkter til test, samt en Mac til at lave et build på. Det koster desuden 535kr. at blive registreret som iOS developer.

|  |  |
| --- | --- |
| **Etableringsbudget** |  |
|  |  |
| **UDGIFTER** | **Kr.** |
| **Kontorinventar:** | 30.350,00 |
| IT (computer, printer, netværk mv.) | 14.197,00 |
| Unity Pro + iOS | 16.153,00 |
|  |  |
| **Vareindkøb:** | 4.000,00 |
| Assets, grafik | 1.000,00 |
| Assets, lyd | 2.000,00 |
| Andet: Unity assets | 1.000,00 |
|  |  |
| **Rådgivere:** | 0,00 |
| Advokat | 0,00 |
| Revisor | 0,00 |
| Andet: | 0,00 |
|  |  |
| **Markedsføring:** | 100,00 |
| Hjemmeside | 100,00 |
| Webshop | 0,00 |
| Andet: | 0,00 |
|  |  |
| **Andre udgifter:** | 535,00 |
| Andet: iOS developer program licens | 535,00 |
|  |  |
| **Udgifter i alt:** | **34.985,00** |

Dertil kommer Unity Pro og Unity iOS Pro, samt eventuelle assets som vi ikke kan eller vil producere selv til vores spil. For eksempel kan det blive nødvendigt at købe os til et soundtrack til spillet. Det er også vores hensigt at indkøbe et plugin som kan håndtere det tunge arbejde i forhold til vores IAP.

Det skal også påregnes at et mindre beløb kommer til at gå til at oprette en firmahjemmeside, men selve opsætningen af denne, samt hvad vi skal bruge af markedsføringsmateriale har vi til hensigt selv at stå for.   
Vi laver ikke markedsføring for virksomheden som sådan, men i stedet på et projekt.

Da vi agter at gå til Nordjysk Iværksætter Netværk for advokat og revisor rådgivning, har vi i første omgang sat udgifterne til disse til nul. Således bliver eventuel fremtidig rådgivning på de felter en del af virksomhedens driftbudget.

På denne måde ender vi med et relativt lavt beløb som vi selv kan finansiere.

## Finansiering

Til virksomhedens opstart vil vi i første omgang forsøge at klare os ved at bruge vores egne penge. Vi kan lave en prototype af vores første spil for omkring nul kroner, da de værktøjer vi skal bruge findes i gratis versioner. Faktisk behøver vi ikke starte virksomheden overhovedet før vores prototype er ved at være klar til fremvisning. Når vi kommer dertil er vi, jf. opstartsbudgettet, nødt til at finde nogle flere penge til opstarten.  
Vi kan her bruge vores egne penge, men af helt åbenlyse årsager ville vi hellere undgå det. Da vores økonomiske udgangspunkt er lidt svagt, har vi ikke den store tiltro til at en bankforbindelse ville være en mulighed i første omgang. Det samme gør sig gældende for venture kapital. I øvrigt er der altid en pris forbundet med det sidstnævnte, og som udgangspunkt vil vi gerne bevare vores frihed. Det lægger også bedre i tråd med vores strategi om organisk udvikling.

Det vi har til hensigt at gøre, er at søge forskellige fonde og offentlig støtte. I forhold til opstart af virksomheden vil vi f.eks. gøre god brug af den gratis rådgivning vi kan få af Nordjysk Iværksætter Netværk.   
Derudover er der en meget lang række forskellige tilskud man kan søge, hvilket er et område som vi vil undersøgte ganske nøje.

## Virksomhedsform

**Zen Rabbit Studios** skal startes op med så lille en startkapital som muligt, hvilket udelukker virksomhedsformerne Aktieselskab (A/S) og Anpartsselskab(Aps) der ville kræve indskudsværdi for hhv. 500.000 og 50.000. Tilbage er der Enkeltmandsvirksomhed, Interessentskab(I/S) og Iværksætterselskab(IVS). Enkeltmandsselskaber og I/S´er kan startes uden videre, men ejeren (eller ejerne hvis der er tale om et I/S) hæfter personligt for virksomhedens eventuelle gæld, og kan på den måde give de involverede massive kvaler i det tilfælde at virksomheden går konkurs.

Tilbage er der IVS, der er en ny forretningsform etableret i januar 2014. Et IVS har den samme struktur som et anpartsselskab. Blot kan det startes op med et indskud på 1 kr. Hvilket er et beløb som vi i fællesskab er i stand til mønstre. Kravet er så at man opsparer penge i virksomheden indtil man runder de 50.000 som det koster at konvertere til et Aps. Først herefter kan man hæve et udbytte i virksomheden.

Et IVS er et selvstændigt retsobjekt, hvilket vil sige, at ejerne ikke hæfter personligt for eventuel gæld.  
Argumentet imod opstart af et IVS er, at det ikke nødvendigvis vil inspirere en finansiel institution til at investere i virksomheden når ejerne ikke har noget på spil. Men på den anden side kan vi heller ikke på anden vis fremskaffe nogen form for sikkerhed, hvorfor spørgsmålet om virksomhedstypen er irrelevant.  
Altså er det hensigten at starte virksomheden som et IVS.

## Projektvalg

Det spil vi har udvalgt som vores første projekt er en hybrid imellem en Endless Runner og et almindeligt action platformspil. Det er to genrer som vi kan se gør sig godt på de mobile platforme, og som, jf vores markedsanalyse, rammer en bred målgruppe. Af samme grund findes der naturligvis en række direkte konkurrenter som vi skal positionerer os i forhold til.

Det gør vi på fire punkter:

1. Spillet er ikke ”endless”, men derimod delt op i kortere baner hvilket giver os mulighed for at etablere en decideret fortælling.
2. Spillet adskiller sig fra typiske platformspil ved at have en proceduralt genereret levelstruktur. Altså, vores baner er forskellige fra spil til spil.
3. Spillet er bygget op med en blanding af 2D og 3D (Typisk betegnet 2.5D), hvilket er unikt for genren, hvor de konkurrenter vi sammenligner os direkte med enten er helt 3D eller helt 2D.
4. Spillet indeholder bosskampe i form af et duelsystem, hvor vores protagonist i bedste westernstil står ansigt til ansigt med de primære antagonister.

Med tanke på virksomhedens relativt begrænsede ressourcer giver det god mening at lave så meget som muligt i spillet proceduralt genereret. Når vi bygger en level op af generiske byggeklodser behøver vi ikke bruge store mængder tid på at lave leveldesign. I stedet skal vi sikre at vores algoritme ikke sætter spilleren i en umulig situation, og at sværhedsgraden er stigende. Begge dele er opgaver som skal løses via kode og almindelig playtest.

Playtest er nemmere at udføre hvis spillets game mechanics er let gennemskuelige. Ved at lave et spil med et featuresæt som må anses for at være casual, gør vi testfasen lettere for os selv, samtidig med at vi tilgodeser vores brugeres behov.

Vores forretningsmodel for dette projekt bliver en blanding af reklamer og IAP. Det vil sige at spillet i sig selv er gratis for brugeren at hente, men de må til gengæld leve med en hvis grad af reklamer.

Angående IAP vil vi implementere en butik i spillet hvorfra vi for mindre beløb kan sælge nye kampagner, og alternative skins til protagonisten.

### Projektbudget

Reklameindtægter er baseret på antal visninger, og beregnes ofte via eCPM.

Det står for “effective cost per thousand impressions” (mille = tusind), og er en industristandard I forbindelse med reklame indtægter. eCPM beregnes ved at dividere den totale indtjening med antal visninger i tusinde.

Der findes ikke et entydigt beløb som man kan forvente at få udbetalt, og den eneste stensikre måde at finde det svar på, er at sende et produkt ud i verden. Det billede der tegner sig når man undersøger området er dog, at der kan være stor forskel i udbetaling an efter hvilket netværk man er tilkoblet. Derfor kan man med fordel benytte flere forskellige reklamenetværk.

Et andet godt argument for at være tilknyttet flere reklamenetværk er, at man på den måde øger produktets fillrate.

Fillrate, i reklame sammenhæng refererer til resultatet af antal reklamer leveret fra reklamenetværk divideret med antallet af forespørgsler på reklamer. Altså, hvor ofte reklamenetværket kan levere en reklame når den givne app har brug for en visning. Hvis der, af den ene eller anden grund, ikke kan leveres en reklame når spillet forespørger det, så er det en tabt visning.

Der er tegn der peger på at eCPM for mobilspil ligger mellem 5,5 og 55kr. I figur XXX tager vi udgangspunkt i en middelpris på 27kr. Hvis vi samtidigt antager at hver bruger der henter spillet ser 3 reklamer, så skal vi have sammenlagt 13 mio. brugere for at projektet opnår målsætningen.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Antal | eCPM(DKK) | Visning pr. download | Før skat | Platform |
| 6500000 | 27 | 3 | 526500 | iOS |
| 6500000 | 27 | 3 | 526500 | Google Play |
|  |  |  | 1053000 | I alt |

Hvis man i stedet tager udgangspunkt i IAP, så er indtjeningsgrundlaget baseret på antallet af brugere der rent faktisk køber noget. Igen er det naturligvis umuligt præcist at fastslå hvor mange der vil benytte sig af den mulighed, men forsigtige estimater indikerer en konversionsrate på 2,5%.

For at kunne lave en konkret beregning forudsættes det her, at vi sælger en ny ”World” til vores spil til 10kr. For at vi kan omsætte for en million skal vi have 4 mio. brugere.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Antal | CR(%) | Pris(DKK) | Før skat | Platform |
| 2000000 | 0,025 | 10 | 500000 | iOS |
| 2000000 | 0,025 | 10 | 500000 | Platform |
|  |  |  | 1000000 | I alt |

Da vi benytter os af en kombination af de to scenarier bliver det rent faktiske antal nødvendige brugere naturligvis lavere. Men da vores samlede potentielle brugerflade er et sted imellem 1,5 -2 mia. mennesker, så er det ikke et urealistisk scenarie.

# Design

## Udviklingsproces

En af de ganske få reelle fordele der er ved at være et lille firma, frem for et stort, er den øgede mulighed for at være fleksibel. Det gælder i alle aspekter af virksomhedsdriften, men er især en force i forbindelse med udviklingsarbejdet.

Spiludvikling er i sin natur en organisk proces, hvor det hele vejen igennem et projektforløb kan give mening at ændre i såvel den teknologi som anvendes til udviklingen, samt spillets design i øvrigt. Det kan altså på et hvilket som helst tidspunkt i udviklingen blive nødvendigt at udføre prototype arbejde på dette eller hint.

Vi er også nødt til at forholde os til, at firmaet ikke fra starten besidder den nødvendige erfaring til at kunne detailplanlægge forløbet på forhånd.

Derfor er det vigtigt at vi vælger en udviklingsproces som understøtter fleksibilitet. Derfor udelukker vi fra starten at udvikle efter traditionelle plandrevne procesmodeller. Plandrevne projekter gør sig bedst i større virksomheder med specialiserede afdelinger og en høj grad af erfaring. Et andet karakteristika ved plandreven udvikling er, at projektet planlægges i detaljer fra starten. Hvilket altså er det diametralt modsatte hvad vi forventer at kunne gøre.

Det giver langt mere mening at benytte en agil udviklingsmetode hvor det netop er fleksibiliteten der er i højsædet. Vores udgangspunkt er derfor at vi bruger Scrum til at styre vores projekt. Scrum i sig selv foreskriver ikke noget omkring den praktiske side af softwareudvilkling, hvorfor vi anvender eXtreme Programming som rettesnor i forbindelse med den regulære udvikling.

### Scrum

For at Scrum kan bruges til vores formål, er det blevet besluttet at lave nogle fundamentale ændringer. Medmindre andet er angivet, benytter vi Scrum og XP som de er.

Den væsentligste ændring vi har foretaget, er at vi accepterer at en story kan oprettes og tages ind midt under et sprint. Det er er ellers en af de få regler som anses for fastlagte i Scrum, at det skal undgås for enhver pris. Når vi alligevel har valgt at fravige den regel, så er det fordi vi ikke vil holdes tilbage af manglende erfaring. Ved oprettelsen af vores product backlog forsøger vi naturligvis at opdele alle aspekter af spillet i stories. Men vi kan ikke garantere, at der ikke vil opstå en situation hvor vi mangler en story for at kunne fortsætte produktionen. Under de omstændigheder vil vi ikke sidde på vores hænder indtil sprintet afsluttes.

Fordi vi forventer uforudsete stories, regulære såvel som spikes, udvælger vi et lavere antal story points til et sprint end det vi egentligt forventer. Fordelen er her, at såfremt vi får behov for at oprette en ny story for at kunne fortsætte produktionen, så har vi bedre tid til det. Hvis det behov ikke opstår, så har vi muligheden for at udfylde tiden ved at tage en anden story ind fra product backloggen.

Der er et par ulemper ved denne fremgangsmåde; først og fremmest, at det gør vores overordnede projekttidsestimering mere upræcis. Fordi vi afsætter mere tid end vi forventer, er tanken at estimeringen gerne skulle forskubbe sig i retning af at projektet tager kortere tid end forventet. På længere sigt er det nu alligevel ikke en holdbar løsning, men præcis estimering er et spørgsmål om erfaring, og for hvert sprint, og hvert projekt, skulle vi gerne blive mere præcise, således at vi kan udfase denne regel.

Måden vi opdeler vores stories på, er ved at sige: Èn feature = Èn story. F.eks. er ”Protagonist, Løb” en story for sig selv. Den involverer at finde en model, få lavet en character controller, påhænge et kamera, scripte brugerinput mv, og tilføje en animation.

Protagonist, Hop-story består af mange af de samme trin, så en del af arbejde ville være klaret hvis den bliver løst efter ”Protagonist, Løb”-story. Ved at opdele stories på den måde vil vi et langt stykke hen af vejen kunne opnå Scrum-tesen om at stories skal kunne løses i tilfældig orden.

Det er ikke hensigtsmæssigt at sikre den uafhængighed 100%, da det blot vil betyde at visse stories ville blive urimeligt omfattende. Hvis f.eks. der blev oprettet en story for ”Gem Spil”, og vi tog den ind først, så ville det medføre at en level og protagonisten blev oprettet først som en del af den story, for at det ville give mening. Altså anvender vi sund fornuft både ved oprettelsen af stories, og ved sprintplanlægningen hvor de stories det næste sprint byder på, skal udvælges.

Fordi vi arbejder på vores eget interne produkt har vi heller ikke en product owner som sådan. I stedet refererer vi til vores GDD, og TDD. Begge dele er dokumenter som alle projektmedlemmer vedligeholder igennem konsensus. I forhold til Scrum master-rollen, så er det ikke en titel vi benytter som sådan, men funktionen ligger primært hos et enkelt teammedlem. Da vi ikke behøver at skulle ”forsvare” os imod en øvre ledelse, eller andre afdelinger som trækker i forskellige retninger, og samtidigt ikke har en udefrakommende kunde, består Scrum masterens rolle primært i at sikre at Daily Scrums og andre aktiviteter bliver holdt som aftalt.

### Extreme Programming

I forhold til udviklingen af scripts, benytter vi os af XP næsten som det er beskrevet i manifestet. En undtagelse er ved konceptet om parprogrammering, hvis funktion vi har erstattet med et almindeligt code review. Det er en beslutning som er truffet af flere grunde. For det første har vi ikke ressourcerne til at kunne gennemføre parprogrammering på en fornuftig måde. Det kræver større skærme, og større arbejdsstationer i det hele taget, end vi har til rådighed. Den anden væsentlige årsag er, at vi mener at miste en del af vores fleksibilitet hvis vi binder to teammedlemmer op på den samme opgave hver gang vi skal producere et script. Det er i den sammenhæng væsentligt at erindre at selve scripting-arbejdet kun er en del af vores samlede arbejdsbyrde.

## Sprint rapport

## Sprint 1

Det første sprint i dette forløb startede naturligt ud med at redegøre og opsætte produktbackloggen samt de grundlæggende værktøjer, som bruges i Scrum og XP. I designfasen af dette projekt blev der redegjort for de features, som produktet skulle indeholde, hvilket i dette sprint blev omdannet til stories. Velocity, burndown chart mv. blev sat på plads og backloggen for det første sprint blev færdiggjort. Sprintbackloggen for det første sprint ses her:

* Player, Løb – 2 point
* Player, Hop – 5 point
* Player, Slide – 3 point
* Player, Opret- ½ point
* Level, Random platform generator – 2 point
* Level, Random baggrunds generator – 1 point
* Level, Baggrundslag – 2 point
* Level, Kamera - ½ point
* Level, ObjectPooling – 2 point
* Controls, mobil – 5 point

I alt 23 story points ud af 25, som vores velocity tillod os.

Med to udviklere på backloggen, blev opgaverne fordelt. En udvikler tog sig af Player-delen, hvor der skulle bygges funktionalitet omkring at få spilleren til at bevæge sige. Den anden havde fokus på den grundlæggende funktionalitet i banen. Begge disse emner var prioriteret højst i produktbackloggen, da de udgør det grundlæggende funktionalitet i produktet og samtidig gør den videre udvikling lettere.

I begyndelsen af Player-funktionaliteten var der brug for undersøgelse af emnet, da der ikke var erfaring med teknologien før. Værktøjet, som skulle bruges, var et system, der stykkede animationer sammen og skulle bruges til spillerens bevægelser. Der var dog problemer med placeholder-animationerne, som ikke passede til vores brug. Dette bremsede udviklingen, og selvom der var forventet en spike på dette område, så tog funktionaliteten længere tid end forventet. Dog lykkedes det i sidste ende at få de grundlæggende bevægelser på plads, men da de endelige animationer ikke er færdiggjorte, har det ikke været muligt at helt afslutte player-delen og kræver mere arbejde længere henne i processen. Ved slutningen af dette sprint blev Player-delen nedskrevet da de sidste tilretninger mv. først kommer senere i forløbet.

Udviklingen med banen gik lettere end forventet, og det var muligt at nedskrive hele Level-delen relativt hurtigt. Arbejdet bestod af nogle relative små spikes, omhandlende tilfældig generering af banen samt generering af baggrunden i banen. Dog er der igen brugt placeholder elementer, og det betyder igen tilretning senere i forløbet.

### Konklusion

Alt i alt er det første sprint vellykket, da det har været muligt at nedskrive alle stories i sprintbackloggen, selv med forventede spikes i forløbet. Dette gav mulighed for at finpudse og optimere den nuværende funktionalitet og inddrage nye stories i sprintet. Her blev dog fokuseret på at imødekomme de næste stories i rækken for at undgå eventuelle spikes og få skubbet funktionalitet ud. Derudover har det givet mere plads til at dokumentere funktionaliteten i rapporten, som i sidste ende er førsteprioriteten for dette projekt. Dog burde man overveje om det er værd at kigge på estimeringen af story points, da stories i dette sprint småt er blevet overvurderet.

## Sprint 2

## Sprint 3

## Problemformulering

# Udvikling

## Teststrategi

### Scope og overblik

Enhver virksomhed, som udvikler software, bør have et teststrategidokument, som beskriver den generelle tilgang til test af de projekter, som virksomheden arbejder på. Derudover laves der en testplan per projekt, hvor der meget specifikt redegøres for, hvad, hvor, hvornår, hvordan, hvem og hvorfor der skal testes. Her redegøres der for de generelle retningslinjer, men tager udgangspunkt i dette projekt. Argumentet er, at siden dette projekt er virksomhedens første og eneste indtil videre, giver det ikke mening at oprette de generelle retningslinjer for test, for derefter ikke at følge dem i planen. Samtidig har virksomheden lille erfaring og der forventes derfor at komme ændringer efter behov.

Dette dokument er beregnet til benyttelse af softwareudviklerne i virksomheden, som skal stå for testen.

### Tilgang til test

Når testforløbet påbegyndes, skal det besluttes, hvilke måder der testes på. Da virksomheden kan være meget alsidig ift. de spil, som kan udvikles, kan de fleste former for test være relevante. Selvom der er mulighed for at tage både unit test og integrationstest i brug, er der valgt at lægge tungt fokus på spiltest og brugertest, da disse prioriteres højst, hvor spiltest står i stedet for integrationstest. Derudover er der ikke stort behov for unit test, integrationstest mv., da spiloplevelsen er fokuspunktet for projektet. Samtidig er den tekniske kompleksitet ikke høj nok til at retfærdiggøre de ressourcer, som skal bruges på at gennemføre dem. I spiltest bliver spillet testet som hele for at sikre kvalitativ funktionalitet, hvor hver enkelt feature spilles igennem og afprøves. I brugertest gives produktet til brugeren for at få direkte feedback fra den målgruppe, man forsøger at ramme. Brugertest kan forekomme på forskellige måder, hvor virksomhedens mål er at have et tæt samarbejde med få antal brugere, som er i tæt dialog med udviklerne.

Derudover er det nødvendigt at afsætte ressourcer til test af performance, da det er essentielt at produktet kan afvikles glidende på de forskellige mobile platforme. Dog er dette et område, som forsøges at tages højde for under udviklingen og ikke i et eventuelt testforløb efter udviklingsprocessen.

### Testmiljø

I denne virksomhed kommer testmiljøet til at variere alt efter projekttype, platform mv. Specielt i dette projekt, som er udviklet i Unity3D og udviklet til mobile platforme kommer testmiljøet til at bestå af Unity3D og de mobile enheder, som virksomheden forventer at udgive til. Et nærmere kig på de mobile platforme og enheder, som skal bruges, kan ses i ’reference til afsnit’. Sidst er det nødvendigt at opsætte Android Development Kit for forbindelsen imellem Unity3D og de Androidenheder, som tages i brug.

Igen er det de to udviklere i virksomheden, som er ansvarlig for opsætning, vedligeholdelse mv. af de relevante miljøer, og der skelnes ikke i adgangsniveau mellem udviklerne. Sikkerhed er heller ikke et fokuspunkt for dette projekt, da produktet ikke indeholder følsomme oplysninger eller der på andre måder kan gøres skade ved misbrug.

Sidst forventes der at bruges versionsstyring og backup i form af GitHub.

### Testværktøjer

De værktøjer, som bruges til performance test hænger tæt sammen med det enkelte projekt, og det er derfor svært at definere en overordnet liste over værktøjer, der skal bruges. I dette projekt bliver der kigget nærmere på de tekniske metoder frem for performanceværktøjer.

### Udgivelseskontrol

Udgivelseskontrol er i første omgang et simpelt mål for virksomheden, da der forventes at arbejde med den samme version igennem hele forløbet. Dette hænger naturligt sammen med virksomhedens størrelse og brugen af versionsstyring, som garanterer den nyeste version er i brug.

### Risikoanalyse

Dette projekt er et relativt risikofrit projekt, da virksomheden har investeret meget lidt i dette projekt. I dette projekt er det tekniske problemer, som udgør den største trussel. Dette er dog et punkt, hvor det er muligt at hente hjælp, da der ikke er tale om nyudvikling eller lignende.

### Godkendelse af test

Undervejs i testprocessen skal testresultaterne godkendes, om de er tolket rigtigt, samt om der er enighed omkring de næste skridt, der skal tages. Igen er det de to udviklere i virksomheden, som har ansvaret for dette.

## Implementering

## Skærmstørrelse

## Videre support

# Perspektivering

# Konklusion

# Bilag

### Ordforklaring

Asset – Dækker over et stykke grafik, script eller lignende som indgår i spillet.

Antagonist – De(n) som spilleren er i mod. Alle fjenderne.

Content – Betyder indhold, og bruges specifikt til at henvise til en del af spillet, og altså ikke

indhold generelt.

Endless Runner – En variant af platformsgenren hvor protagonisten løber endeløst ud af en bane, uden mulighed for at stoppe, eller sætte hastigheden ned, mens han (m/k/u) undgår forskellige forhindringer og samler point sammen indtil han dør. Typisk øges hastigheden langsomt hen af vejen. Temple Run og Jetpack Joyride er populære eksempler på hhv. en 3D og 2D endless runner.

Feature – En specifik game mechanic.

Fillrate, GPU – Tilgår.

Free-To-Play – Betalingsmodel hvor spillet I sig selv er gratis, og hvor udvikleren I stedet bliver

betalt via reklamer, eller mikrotransaktioner.

Gameplay – Refererer til de features som spilleren har direkte til rådighed. F.eks. styringen af protagonisten.

Game Design Document (GDD) – Er et levende dokument der indeholder alle oplysninger om spillet. Dvs. En beskrivelse af spillets genre, spillets stil, hvilke features der indgår mv.

Game Engine – Er et system der bruges til udvikling af spil. Formålet er at forenkle spiludviklingen ved f.eks. at give adgang til en grafisk scene editor, indbyggede fysik-klasser, multiplatform kompilerer osv. Unity er en game engine.

Game Mechanics – Folk der reparerer spil.

In App Purchases – Bruges sammen med mikrotransaktioner, og er en ”butik” i spillet hvor

brugeren kan købe virtuelle dele til spillet for rigtige penge.

MMO – Massively Multiplayer Online. Et online multiplayerspil med en persistent verden.

Model – Refererer til et 3D objekt. Vores protagonist er en model, ligeså vores togvogne etc.

Monetiseringsstrategi – Fordanskning af monitization strategy. Dækker over de forskellige metoder man kan gøre brug af for at tjene penge på et produkt. Typisk menes der i forhold til

Free-to-Play-produkter, men i princippet dækker det alle salgsmetoder.

Placeholder – Kan være en texture, en model, en lydeffekt eller et andet spilelement, som optræder i stedet for det endelige asset. Formålet er at kunne bygge en prototype af spillet op uden at have den færdige version klar på forhånd.

Platformsspil – En klassisk spilgenre, hvor spilleren har som mål at traversere en bane med forskellige fjender og forhindringer. I modsætning til en endless runner kan spilleren normalvis stoppe op, gå bagud i banen mv. Super Mario er måske det mest berømte eksempel på genren.

Mikrotransaktioner – Salgsmetode hvor brugeren køber mindre forskellige goder til brug i spillet for rigtige penge.

Playtest – Testmetode hvor man tester spillet ved simpelthen at spille det, og forsøge at ”ødelægge” spillet via brug.

Proceduralt generet – Betyder at noget content er genereret ud fra en algoritme, og altså kan ændre sig fra spil til spil, baseret på den algoritme.

Protagonist – Vores hovedperson, den som spilleren styrer.

Script – Er et stykke kode der skrives oven på den game engine der benyttes, med det formål at manipulere spillets objekter. Konkret dækker det over al vor kode.

Scope – Relaterer til projektets størrelse. F.eks. ville det være ude af scope hvis vi besluttede at selvkomponere al musikken i spillet, eller lave motion-capture til vores animationer.

Skin – I denne sammenhæng refererer skin til en models udseende. F.eks kunne vi give vores protagonist et Skovhugger Skin, et Gangster skin, eller noget helt tredje. Det er altså et spørgsmål om at skifte figurens textures. Skins sælges ofte via IAP.

Standalone klient – En spilklient som kan afspilles på f.eks. PC uden at køre i en emulator eller igennem en webside (som f.eks. en Facebook app)

Texture – Er et billede som ”beklæder” et objekt.

## Problemformulering

I dette afsnit præsenteres projektets problemformulering. Den består af et hovedspørgsmål som er virksomhedens overordnede problem, samt nogle uddybende underspørgsmål.

Spørgsmålene er udledt af de problemer og behov der som IT-forundersøgelsen har afsløret i forbindelse med opstarten af Zen Rabbit Studios.

### Hovedspørgsmål

* Hvordan opstartes et bæredygtigt spilfirma.

### Delspørgsmål

* Hvilke forretningsmæssige initiativer skal virksomheden sætte i værk for at kunne eksistere.
* Hvilke tekniske forudsætninger er der bundet op på spilproduktion.
* Hvordan designer man et spil.

# Projektafgrænsing

I forbindelse med udarbejdningen af denne rapport kommer vi omkring en lang række teknologier og metoder. For at kunne begrænse omfanget af rapporten til det tilladte, er der nogle aspekter som vi enten helt udelader, eller kun berører overfladisk.  
I forhold til vores teknologianalyse vil vi kun beskæftige os med et smalt udsnit af de produkter der reelt er til rådighed, baggrunden for udvælgelsen af hvert enkelt produkt er beskrevet i det relevante kapitel.

I forhold til udviklingen af grafiske artefakter, og de teknologier der er forbundet med den udvikling, så er det et emne som vi kun ganske kort berører. Al endeligt materiale af den art bliver udviklet af den tredje virksomhedsgrundlægger, der er grafisk uddannet, og som i øvrigt ikke er involveret i udarbejdelsen af denne rapport.

I forbindelse med test findes der en lang række af bugtracker-værktøj som vi kunne vælge at gøre brug af. I forbindelse med dette projekt vil vi ignorere disse værktøjer, da vi ikke har et reelt behov for at gøre brug af nogen af dem.

Det er af, af flere grunde, ikke muligt færdiggøre vore spil inden for projektets deadline. Det skyldes primært, at produktionen kræver en lang række forskellige aktiver for at kunne færdiggøres. Vi har derfor kun til hensigt at have en prototype klar til aflevering ved projektets afslutning. Det vil sige at vi kun bruger placeholder grafik og lyd, og at vi kun laver en enkelt spilbar ”verden” færdig.   
  
Selvom spillet skal have reklamer og mikrotransaktioner indbygget, så er det heller ikke elementer som kommer til at figurere i prototypen. Den uddannelsesmæssige værdi vi kunne opnå ved at implementere det er begrænset, da vi vil benytte os af tredjeparts plugins.

Det sidste forbehold angår det faktum, at vi ikke udvikler prototypen til iOS, men Android. Det gør vi fordi vi ikke umiddelbart råder over de tekniske virkemidler der skal til for at udvikle til iOS.

# IT-forundersøgelse

IT-forundersøgelsen klarlægger de styrker og svagheder som eksisterer ved opstarten af **Zen Rabbit Studio**. Dette opnåes ved dialog imellem iværksætterne, og strategisk analyse. Efter dette kapitel er der en klar forståelse af hvilken type virksomhed der er under opbygning, samt dens fremtidige muligheder. Ligeledes forelægger der en detaljeret handlingsplan for den første periode af virksomhedens eksistens, samt en mere løs vision for tiden derefter.  
Kapitlet er baseret på <KILDER> og ejernes vision.

## Nuværende situation

1. http://vgsales.wikia.com/wiki/Video\_game\_industry [↑](#footnote-ref-1)