# 

**Tenebris**

**24**

**Projektplan**

**V. 1.04**

**2018-05-011**

# **Dokumenthistorik**

|  |  |  |  |
| --- | --- | --- | --- |
| Datum | Version | Beskrivning | Författare |
| 180316 | 1.0 | Skapandet av projektplanen | Samtliga gruppmedlemmar |
| 180318 | 1.01 | Uppdatering enligt feedback | Anton Persson |
| 180405 | 1.02 | Dokumentgranskning och skrev riskanalys | Oscar Rehnberg |
| 180505 | 1.03 | Uppdatering enligt feedback. | Anton Persson |
| 180511 | 1.04 | Uppdatering enligt feedback | Oscar Rehnberg |
|  |  |  |  |

**Innehåll**

Dokumenthistorik. 2

Projektplan. 3

Syfte. 3

Ordlista. 3

Översikt av projekt. 4

Syfte. 4

Omfattning. 4

Mål 4

Produkt 4

Produktbeskrivning. 4

Målgrupp. 5

Process. 5

Utvecklingsprocess. 5

Bemanning och ansvarsområden. 6

Planering. 6

Grovplan. 6

Riskanalys. 7

Referenser. 9

# **Projektplan**

# **Syfte**

Projektplanen baseras på den fundamentala idéen att skapa en beskrivning av projektets livscykel - alltså frågor som var, hur, när och vem?

Informationen i hela detta dokument har som mål att hjälpa och vägleda medlemmar av “No Content” om hur man fördelar de begränsade resurserna som projektet innehåller; som sådant kommer det finnas veckoplanering, samt beskrivningar om hur processen i dess helhet kommer att hanteras. Projektplanen kommer framöver att uppdateras innan varje sprint.

# **Ordlista**

**Levels:** Nivåer som spelaren tar sig igenom för att besegra spelet.

**Items:** Föremål som hjälper spelaren att besegra eventuella motståndare.

**Abilities:** Krafter som hjälper spelaren att besegra eventuella motståndare.

**Gamebreaking:** Något som gör så att spelet ej blir exekverbart.

**Turn Based Combat System:** Tur baserat strid system, där spelaren interagerar med motståndare en runda i taget.

**Healing Item:** En sak som återställer förlorade liv poäng till spelaren.

**Procedur Genererade spelnivåer:** procedur generering är en metod för att skapa data algoritmiskt i motsats till manuellt.

**Dungeon-crawler:** En dungeon-crawler är en typ av scenario i fantasi-spel där hjältar navigerar i en labyrint miljö (en "dungeon"), kämpar med olika monster och plundrar olika sorter av skatter som de kan hitta inom spelet.

**Roguelike:** Roguelike är en undergrupp av rollspel-spel som karaktäriseras av en dungeon-crawler genom procedur genererade spelnivåer, turbaserad gameplay, tile-baserad grafik och permanent död hos spelaren. De flesta roguelikes är baserade på en hög fantasi berättelse, vilket speglar deras inflytande från brädspels rollspel som Dungeons & Dragons.

**MoScoW modellen:** kraven delas upp i must( krav märkta som must är avgörande för den aktuella leveranstiden för att det ska bli en framgång.), should( krav märkta som should är viktiga men inte nödvändiga för leverans i den aktuella leveranstiden.), could have( krav märkta som could have är önskvärda men inte nödvändiga, och kan förbättra användarupplevelse eller kundnöjdhet för lite utvecklingskostnad.), och wont have(krav som är märkta som wont have har kommit överens av intressenter som de minst kritiska, lägsta återbetalningsobjekten, eller inte lämpliga vid den nuvarande tidpunkten.).

# **Översikt av projekt**

## **Omfattning**

Projektet utgår på skapelsen av ett spel. Spelet är ett roguelike dungeon crawler med element som turn based combat och platformer. Det kommer i huvudsakligen bestå av programmering inom C#, samt dokumentering av våra framgångar inom spelets utveckling. Vi tänker inte försöka skapa eget ljud till spelet, och har bara planer på att fixa egna sprites till spelet skulle våran tidsplan tillåta oss att utforska hur man gör det.

## **Mål**

Målen kan uppdelas i två olika kategorier; kategori ett är projektmålen. Projektmålen beskriver vad gruppen vill uppnå inom projektet och dess krav. Andra kategorin är personliga mål, vilket relativt uppenbart beskriver de mål som varje medlem vill personligen uppnå eller förbättra sig på.

Projektmålet är att uppnå full färdigställande i Must-kategorin, och förhoppningsvis slutföra alla should-specifikationer som har en direkt korrelation med ovanstående Must-kategori.

Personliga målet är att förbättra “No Content”s kunskap inom projekthantering och utveckling samt att vidareutveckla gruppens kunskaper inom både programmering och grafisk hantering.

# **Produkt**

## **Produktbeskrivning**

Ett roguelike dungeon crawler platformer turn based combat spel med ett mörkt tema. En stor del av spelet är boss fights i olika levels, bossarna ska ha unika mekaniker och vara distinkta från varandra. Det är tänkt att spelaren ska hitta olika “items” eller “abilities” för att besegra bossarna som i de klassiska Zelda spelen. Att combat systemet är turn based innebär att spelaren har ett visst intervall där de kan göra något, till exempel använda ett healing item eller attackera. Men när spelaren är klar med vad de vill göra så är det motståndarens tur och under det intervallet kan inte spelaren göra något. Spelets tema är mörkt och det innebär att nivåerna i spelet ska ha mörka färger, alltså inte mycket ljus eller ljusa färger dessutom skrämmande fiender.

## 

## 

## **Målgrupp**

Målgruppen för detta projekt kommer att vara folk som uppskattar att ta sig an svåra utmaningar som begär en stor del strategiskt tänkande, är villiga att investera en stor del tid för att lära sig spelets layout samt bygga muskel reflexer för att navigera plattformarna. Det finns inga ålder eller köns krav för spelet, det handlar helt om en persons attityd mot spel. Dock så är det förväntat att det kommer huvudsakligen spelas av folk i åldersgrupp 15-45. Gruppen anser inte att folk utanför de åldrarna kanske har motivationen eller tiden för att ta sig an spelet.

**Process**

## **Utvecklingsprocess**

Våran utvecklingsprocess kommer att hanteras enligt den agila metodologin[[1]](#footnote-1), och således kommer inte enorma resurser att fördelas på att följa planeringen linjärt; projektet kommer anses som ett dynamiskt fenomen - “No Content” är alltid redo på att förutsättningarna kommer ändras.

Arbetet enligt den agila metodologin kommer bedrivas inkrementellt och iterativt vilket innebär att fungerande delleveranser av funktionalitet sker regelbundet enligt nedan schema och att planer och metoder löpande utvärderas och förbättras. Ändamålsenlig och användarcentrerad utveckling eftersträvas genom ett nära samarbete under hela utvecklingstiden med täta och regelbundna möten mellan utvecklare. Dessa möten kommer att bedrivas på skolan men även på discord och liknande kommunikationsmedium.

Utformningen av kraven hanterades under ett planerat gruppmöte där det diskuterades hur gruppen ville att intressenterna och eventuella kunder skulle uppfatta spelet, dess mekaniker och dess tema. Kraven bestämdes genom användning av Moscow modellen[[2]](#footnote-2) och delades således upp i *Must, Should* och *Could* krav. En riskanalys gjordes för att lokalisera begränsningar som påverkar krav (begränsningar som t.ex. tid och kunskaper). Processen kommer innehålla kodstruktur (Objektorienterad programmering), testdokumentation (validering -verifieringtester) samt granskning av olika slag. Eftersom processen är agil så kommer utvecklingsprocessen att uppdateras efterhand då t.ex. hur man ska angripa tester kommer att diskuteras vid lämpligt tillfälle framöver, och även förbättras under projektets gång.

## 

## **Bemanning och ansvarsområden**

Stora beslut kommer görs helt demokratiskt.

Oscar Rehnberg: Programmerat spel med c#. Ansvarsområde: Dokumentgranskning och protokoll.

Anton Persson: Har omfattande grundkunskap inom C# och liknande kunskaper inom databasteknik/sql. Ansvarsområde: Kodgranskning och design dokumentationen.

Christian Steneram Bibby: Har omfattande grundkunskap inom C# och expertis inom spel. Ansvarsområde: Konsult, användbarhetsanalys.

Dionist Peci: Programmerat spel i c#, ansvarsområden: Testning.

# **Planering**

## **Grovplan**

### **Vecka 12**

Aktiviteter

· Handledning 2/4 timmar: Möte med handledare.

· Närvara vid dokumentation hjälp 12/4 timmar: Lösa Git samt YouTrack

· Projektplan 8/4 timmar: Färdigställa allting i projektplanen

· Kravdokumentation 8/4 timmar: Färdigställa allting i kravdokumentation

· Koda 30/4 timmar: Göra klart ett demo med placeholder sprites

Milstolpar

· Projektplan i första version 23/03 samt kravdokumentation

o Ett första utkast av kravdokumentation och projektplan

o Leverabler:

§ Kravdokumentation

§ Projektplan

· Demo av spelet

o Ett demo av spelet där några av Must-kraven är uppfyllda

o Leverabler:

§ Ett demo

### **Vecka 13**

Aktiviteter

· Koda 30/4 timmar: Fortsätta implementera element

· V-V-dokumentation 16/4 timmar: Första utkast av V-V

· Dokumentation 14/4 timmar: Uppdatering av dokumentation

· Individuell fördjupning 4/4 timmar: Individuell fördjupning

Milstolpar

· Dokumentation 30/03

o Dokumenten ska vara klara och uppdaterade

o Leverabler:

§ V-V-dokument

§ Nya versioner av dokument

· Kodning 30/03

o Spelet ska innehålla fler krav från Must-kraven

o Leverabler:

§ Spel

**Vecka 15**

Aktiviteter

· Koda 30/4 timmar: Fortsätta implementera element

· Börja koda för battle

· Fixa traps, mm

Milstolpar

· Kod

o Kodning

o Leverabler:

§ 3 olika traps

**Vecka 16**

Aktiviteter

· Uppdatera dokument

· Planera inför retrospekt

· Fortsätta koda

Milstolpar

· Dokumentation

o Dokumenten ska vara klara och uppdaterade

o Leverabler:

§ Nya versioner av dokument

· Kodning

o Rensa upp koden

o Leverabler:

§ Ny version av kod

**Vecka 17**

Aktiviteter

· Uppdatera dokument

· Planera inför retrospekt

· Fortsätta koda

Milstolpar

· Dokumentation

o Dokumenten ska vara klara och uppdaterade

o Leverabler:

§ Nya versioner av dokument

· Kodning 30/03

o Spelet ska innehålla alla must-krav

o Leverabler:

§ Spel

**Vecka 18**

Aktiviteter

· Uppdatera dokument

· Planera inför retrospekt

· Fortsätta koda

Milstolpar

· Dokumentation

o Dokumenten ska vara klara och uppdaterade

o Leverabler:

§ Nya versioner av dokument

· Kodning 05/05

o Fler should krav ska implementeras

o Fix av bugs

o Leverabler:

§ Spel

# **Riskanalys**

Syftet med riskanalysen är att tydliggöra vad som kan hända i projektet och hur man ska gå tillväga för att förhindra eller lösa problemet. Riskerna graderas i sannolikhet och konsekvens från låg till hög.

Riskernas sannolikhet att förekomma:

Låg - Liten sannolikhet att risken inträffar.

Medel - Viss sannolikhet att risken inträffar.

Hög - Stor sannolikhet att risken inträffar.

Riskernas konsekvens:

Låg - Påverkar projektet men bara lite.

Medel - Projektet kommer lida av dessa och bör tas om hand om.

Stor - Projektet bör förebygga detta i största mån.

## **Identifierade risker**

**A1 Konflikt mellan gruppmedlemmar**

Sannolikhet: Låg

Konsekvens: Medel

Handlingsplan: Främst ska varje medlem uppföra sig och ha en god ton mot varandra. Om en konflikt uppstår kan vi förlita oss på det sociala kontraktet.

**A2 Spelets design**

Sannolikhet: Hög

Konsekvens: Medel

Handlingsplan: Detta kan leda till konflikter och därför är det viktigt att diskutera spelets design så tidigt som möjligt. Om det uppstår någon förvirring får det bli en demokratisk röstning eller en kompromiss.

**A3 Sjukdomar**

Sannolikhet: Hög

Konsekvens: Låg

Handlingsplan: Gruppmedlemmarna kommer att bli sjuka och beroende på hur stor arbetsbörda det är kan resten av gruppmedlemmarna ta över arbetet eller så kan den sjuke arbeta ikapp när hen är frisk.

**A4 Dålig kommunikation**

Sannolikhet: Medel

Konsekvens: Hög

Handlingsplan: För att undvika detta problem får alla medlemmar försöka vara ärliga och säga vad de tycker, man bör även fråga gruppmedlemmar som inte har uttryckt sig.

## **Riskdiagram**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | Sannolikhet |  |
|  |  | Låg | Medel | Hög |
|  | Låg |  |  | A3 |
| Konsekvens | Medel | A1 |  | A2 |
|  | Hög |  | A4 |  |

**Referenser**

1. Jesse Reynosa, *Complete guide to agile methodology,* <https://www.qasymphony.com/blog/agile-methodology-guide-agile-testing/> (hämtad 3/28/2018)
2. DSDM Handbooks, *MosCow Prioritisation*, <https://www.qasymphony.com/blog/agile-methodology-guide-agile-testing/> (hämtad 3/28/2018)

1. Jesse Reynosa, *Complete guide to agile methodology,* <https://www.qasymphony.com/blog/agile-methodology-guide-agile-testing/> (hämtad 3/28/2018) [↑](#footnote-ref-1)
2. DSDM Handbooks, *MosCow Prioritisation*, <https://www.qasymphony.com/blog/agile-methodology-guide-agile-testing/> (hämtad 3/28/2018) [↑](#footnote-ref-2)