**Requirements for GetOut**

Die Story von Steve ist ein storygetriebenes, möglichst realitätsnahes Psychological Horrorspiel, das den Spieler in die Rolle von Steve versetzt – einem Mann, der inmitten einer persönlichen Krise steht. Durch den Tag und seine albtraumhaften Erlebnisse begleitet, taucht der Spieler tief in Steves Realität und dessen brüchige Psyche ein.

Autoren:

Christian Schäfer

Andrei Costeniuc

Timo Skrobanek

1. **Rahmenbedingungen**

Ziel in diesem Projekt ist es ein Video-Spiel zu entwickeln, welches der oben genannten Beschreibung entspricht.

Das Spiel bekommt für den Zeitraum der Entwicklung den Namen „GetOut“. Der finale Name wird zum Schluss festgelegt.

GetOut soll ein möglichst realistisches psycho-horrorspiel werden. Dafür werden auf realistische Assets von verschiedenen Plattformen gesetzt sowie einer modernen Engines wie Unreal Engine.

1. **Phasenaufteilung**
2. **Konzeptionelles Design**

Das Spiel wird sowohl von der Story als auch vom ersten Aussehen beschrieben. Es wird ein Storyboard erstellt und erste Statemachine Diagramme werden erstellt um den Ablauf im Spiel zu beschreiben.

Zeitaufwand: 10h

1. **Technisches Design**

Die Architektur sowie das genaue Aussehen vom Programmiercode werden hier beschrieben. Dabei werden Klassendiagramme, Sequenzdiagramme und ähnliche angelegt.

Zeitaufwand: 14h

1. **Entwicklung**

In dieser Phase wird GetOut schlussendlich umgesetzt. Mithilfe einer Engine werden nun alle Anforderungen übersetzt.

Zeitaufwand: ~120h

1. **Tests**

Nachdem GetOut implementiert wurde, gibt es mehrere große Testphasen, die möglichst viele Szenarien abdecken sollen.

Zeitaufwand: 26h

1. **Fertigstellung**

In dieser letzten Phase wird das Spiel für die verschiedenen Plattformen fertiggestellt und ein Report geschrieben. In diesem Report wird nochmals genauer auf die einzelnen Phasen eingegangen

Zeitaufwand: 10h

1. **Definition der Anforderungen**

Folgende Usertasks sollen durch das Spiel umgesetzt werden. Aus diesen ergeben sich die Systemfunktionen.

|  |  |  |  |
| --- | --- | --- | --- |
| **Nr.** |  | **Titel** | **Beschreibung** |
| 1. |  | Play Game | Der Benutzer soll das Spiel starten und anschließend spielen können. Seine Spielstände sollen gespeichert und wieder abgerufen werden. |
|  | 1 | Save Game | Spielstände müssen in Dateien gespeichert werden. Dabei muss besonders der Stand der Story gespeichert werden sowie welche Achievements erlangt wurden. |
|  | 2 | Load Game | Aus dem Spielmenü heraus soll es möglich sein verschiedene Spielstände anzusteuern und diese zu laden. |
|  | 3 |  |  |
| 2. |  | Stop Game | Es soll möglich sein das Spiel in jeder Situation schließen zu können.  Dabei soll jeweils „1.1 Save Game“ ausgeführt werden. |
|  |  |  |  |

1. **Projekt-Meilensteine**

GetOut wird modular entwickelt, dabei werden die Szenen jeweils für sich gestaltet und mit Logik versehen. Somit kann eine Abnahme jeder einzelnen Szene erfolgen, was den gesamten Entwicklungsablauf nicht großartig beeinflusst, da die restlichen Szenen parallel weiterentwickelt werden.

Sind die Szenen für sich betrachtet fertig, so werden die Übergänge zwischen diesen einprogrammiert. Somit werden alle Teile der Software zu einem zusammenhängenden Spiel.

Zum Schluss wird das Spielmenü programmiert damit der Spieler vom Spielstart bis zum eigentlichen Spiel durch unsere Software navigieren kann.

Nachdem alle Einzelteile der Software zusammengesetzt wurden, gibt es eine erste Testrunde durch die Programmierer (WhiteBox-Testing). In dieser werden mögliche Situationen durch uns aufgestellt und schlussendlich getestet. Sind alle Tests abgeschlossen und die Fehler behoben, wird eine zweite Testrunde angesetzt, wobei nun Personen die Software testen, welche nicht an der Entwicklung beteiligt waren (Blackbox-Testing).

Sind auch diese Tests abgeschlossen, wird das Spiel zur Abgabe an Prof. Hesser fertiggesteltl.

1. **Anlagen und Resourcen**

Alle Diagramme sowie ein Handout mit der Übersicht über alle Szenen und den gesamten Spielablauf gibt es in der Git-repository.