|  |  |  |  |
| --- | --- | --- | --- |
| [Wählen Sie das Datum aus] |  |  | |
| |  |  | | --- | --- | |  |  | |  |  | |  |  | |  |  | | |  | |
| 3D-Comupterspiel - DOFGII  *[Geben Sie den Untertitel des Dokuments ein]* | | | |
| |  |  |  | | --- | --- | --- | | **Name** | **E-Mail** | **Matrikelnummer** | | Jennifer Kuschnig | [Kuschnigje60640@th-nuernberg.de](mailto:Kuschnigje60640@th-nuernberg.de) | 2665234 | | Tobias Velke | [VelkeTo61783@th-nuernberg.de](mailto:VelkeTo61783@th-nuernberg.de) | 2693554 | | Micha Zimmermann |  |  | | Samuel Seiz |  |  | | Christoph Huxhagen |  |  | |  | |  |
|  |  | |  |

Inhalt

[1 Einführung 2](#_Toc439009249)

[1.1 Projekthintergrund 2](#_Toc439009250)

[1.2 Projektauftrag 2](#_Toc439009251)

[1.3 Aufgabenbereich Teammitglieder 2](#_Toc439009252)

[2 Analyse 3](#_Toc439009253)

[2.1 Funktionale Anforderungen 3](#_Toc439009254)

[2.2 Nicht Funktionale Anforderungen 4](#_Toc439009255)

[3 Entwurf 4](#_Toc439009256)

[3.1 UML 1 5](#_Toc439009257)

[3.2 UML 2 5](#_Toc439009258)

[3.3 UML 3 5](#_Toc439009259)

[4 Implementierung 5](#_Toc439009260)

[4.1 Quellcode – Konventionen 5](#_Toc439009261)

[4.2 Quellcode – Ausschnitte 5](#_Toc439009262)

[5 Tests 5](#_Toc439009263)

[5.1 Akzeptanztest 5](#_Toc439009264)

[5.2 Integrationstest 6](#_Toc439009265)

[5.3 Systemtest 6](#_Toc439009266)

[6 Fazit 6](#_Toc439009267)

[7 Glossar 7](#_Toc439009268)

# Einführung

## Projekthintergrund­

Dieses Projekt zielt darauf ab ein vollständiges Softwareprojekt zu entwickeln. Dabei muss das Team eigenständig alle Phasen des Projektes von der ersten Konzeptionierung bis hin zu abschließenden Qualitätstest durchführen.

Hierbei sollen die Teammitglieder erste Erfahrungen mit Methoden der Softwareentwicklung sammeln, Techniken entwickeln und sich in neue Werkzeuge einarbeiten.

## Projektauftrag

Die Praktikumsaufgabe umfasst die team-basierte Konzeptionierung, Umsetzung und Dokumentation eines Softwareprojekts unter Anwendung der in der Vorlesung behandelten Methoden und Techniken, welche folgendermaßen lautet:

***Entwickeln Sie eine Computerspiel-Demo basierend auf einer (2D-)Game-Engine*.**

Die Game-Engine soll vom Team gewählt werden, wobei darauf geachtet werden muss, dass diese eine Entwicklung mit C# bzw. C++ ermöglicht. Zudem muss sich das Team selbstständige in die gewählte Engine einarbeiten.

Bei der Entwicklung sollen die gelernten Prinzipien der objektorientierten Softwaretechnik angewendet werden. Der Auftrag soll in folgenden Kernphasen der Softwaretechnik gegliedert sein:

* Analyse
* Entwurf
* Implementierung
* Tests

Als abschließendes Projektziel soll eine lauffähige Demoversion präsentiert werden.

## Aufgabenbereich Teammitglieder

|  |  |
| --- | --- |
| Mitglied | Aufgaben |
| Jennifer Kuschnig | Shop-Design, Waffen-Design, Level-Design |
| Micha Zimmermann | Enemy-Design, Level-Design |
| Samuel Seiz | Enemy-Design, Git-Hub |
| Tobias Velke | Game-UI, Bonus-Design, Projektmanagement |
| Christoph Huxhagen | Player-Design, Waffen-Logik |

# Analyse

Die in diesem Kapitel aufgeführten Anforderungen verstehen sich nur als ein Auszug aller von uns bestimmten Anforderungen an das zu entwickelnde Produkt und spiegeln lediglich exemplarisch unsere Vorgehensweise wieder.

## Funktionale Anforderungen

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Story\_ID: | 1 | Story\_Name: | Player\_Movement | |  | Risk: | 8 |
| As an: | Spieler | |  |  |  | Priority: | 10 |
| I would like to | möchte ich das Spielobjekt auf der XY-Ebene in alle Richtungen bewegen können. | | | | | est. Work [h]: | 6 |
|  |  |  |
| Given: | Das Spiel befindet sich im Play Mode | | | | |  |  |
|  |  | | | | |  |  |
|  |  | | | | |  |  |
| So that: | jeder Punkt der X-Z-Ebene ist durch betätigen der WASD-Tasten erreichbar | | | | |  |  |
|  | es gibt keinen Bereich der X-Z-Ebene der nicht erreichbar ist | | | | |  |  |
|  |  | | | | |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Story\_ID: | 2 | Story\_Name: | Player\_Attack | |  | Risk: | 10 |
| As an: | Spieler | |  |  |  | Priority: | 8 |
| I would like to | möchte ich die gegnerischen Spielobjekte durch Schüsse zerstören können. | | | | | est. Work [h]: | 3 |
|  |  |  |
| Given: | Das Spiel befindet sich im Play Mode | | | | |  |  |
|  |  | | | | |  |  |
|  |  | | | | |  |  |
| So that: | Das gegnerische Spielobjekt nicht mehr existiert, | | | | |  |  |
|  | Ich als Spieler dafür Punkte bekommen | | | | |  |  |
|  | Und ggf. ein neues gegnerisches Spielobjekt erstellt wird | | | | |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Story\_ID: | 3 | Story\_Name: | Player\_Health | |  | Risk: | 6 |
| As an: | Spieler | |  |  |  | Priority: | 7 |
| I would like to | Möchte ich über einen vorgegebenen Vorrat an Lebenspunkten verfügen, welche durch Kollision oder durch Treffer gegnerischer Spielobjekte reduziert werden | | | | | est. Work [h]: | 2 |
|  |  |  |
| Given: | Das Spiel befindet sich im Play Mode | | | | |  |  |
|  |  | | | | |  |  |
|  |  | | | | |  |  |
| So that: | Ich eine optische Anzeige meiner verbleibenden Lebenspunkte habe | | | | |  |  |
|  | Wenn mein Lebenspunkte Vorrat gleich 0 ist, das Spiel beendet wird | | | | |  |  |
|  | Ich meinen Vorrat an Lebenspunkte zu Beginn eines Levels oder durch spezielle Boni wieder auffüllen kann | | | | |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Story\_ID: | 4 | Story\_Name: | Weapon\_Shop | |  | Risk: | 7 |
| As an: | Spieler | |  |  |  | Priority: | 7 |
| I would like to | möchte ich nach absolvieren eines Levels in einen Waffenshop wechseln können um dort neue Waffen für mein Spielobjekt kaufen zu können. | | | | | est. Work [h]: | 4 |
|  |  |  |
| Given: | Das Spiel befindet sich im Play Mode | | | | |  |  |
|  |  | | | | |  |  |
|  |  | | | | |  |  |
| So that: | Nach beenden eines Levels soll ein klar erkennbarer Wechsel in einen Shop stattfinden | | | | |  |  |
|  | In diesem Shop soll es mir abhängig der Anzahl meiner Coins möglich sein neue Waffen zu kaufen. | | | | |  |  |
|  | Nach dem Kauf soll zurück zum Spiel und somit zum Beginn eines neuen Levels gewechselt werden | | | | |  |  |

## Nicht Funktionale Anforderungen

# Entwurf

## Sequenzdiagramm



## UML 2

## UML 3

# Implementierung

## Quellcode – Konventionen

## Quellcode – Ausschnitte

# Tests

## Akzeptanztest

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test\_ID | Test\_Name | Test\_Preconditions | Test\_Scenario | Test\_Result |
| 1 | Player\_Movement | Spiel befindet sich im Spiel Modus, Spieler ist aktiv | Spieler mit WASD- Tasten über das Spielfeld bewegen | Der Spieler konnte zu jeder Position auf dem Spielfeld und darüber hinaus bewegt werden. Der Spieler war zu keinem Zeitpunkt manövrierunfähig. |
| 2 | Player\_Attack | Spiel befindet sich im Play Modus, Gegner und Spieler sind aktiv | Mit dem Spieler soll auf einen Gegner gezielt und drücken der STRG-Taste beschossen werden. Wurde der Gegner getroffen so soll er zerstört werden und der aktuelle Punktestand erhöht werden | Sobald ein Gegner getroffen wurde, wird es zerstört. Des Weiteren wird der aktuelle Punktestand um Eins erhöht.  Dieses Verhalten konnte bei allen verschiedenen Gegnertypen nachvollzogen werden. |
| 3 | Player\_Health | Spiel befindet sich im Play Modus, Gegner und Spieler sind aktiv, Lebenspunkte des Spieler sind >0 | Die verschieden Gegner können dem Spieler unterschiedlich Schaden zufügen um dies zu testen soll jeder Gegnertypus dem Spieler Schaden zufügen und die Auswirkung überprüft werden. | Enemy 1 erzielt 50 Schadenpunkte durch Kollision mit dem Spieler,  Enemy 3 erzielt 40 Schadenpunkte durch Schüsse auf den Spieler,  Enemy 3 erzielt 10 Schadenspunkte durch Schüsse auf den Spieler.  Sobald der |
| 4 | Weapon-Shop | Spiel befindet sich im Play-Modus,  nach zerstören eines Gegner ist die benötigte Punktzahl zum beenden des Levels erreicht | Nach beenden eines Levels soll automatisch in den WeaponShop gewechselt werde.  Dort sollen unterschiedliche Waffen zum Kauf angeboten werden, jede dieses Waffen hat einen unterschiedlichen Preis, vor dem Kauf soll überprüft werden ob der Spieler genügen Coins besitzt.  Der Spieler soll den Shop über einen Button verlassen und somit das nächste Level starten können | Nach zerstören des letzten Gegners wechselt das Spiel automatisch in den Shop  Dort kann man aus (bisher) drei unterschiedlichen Waffen wählen.  Wurde eine Waffe gewählt ohne genügend Coins zu besitzen wird ein Dialog mit entsprechender Meldung angezeigt.  Wird eine Waffe gekauft und der Shop verlassen wechselt das Spiel wieder in den LevelModus. Die vorherigen Punkte sowie die aktuelle Anzahl an Coins werden in der GUI angezeigt |

## Integrationstest

## Systemtest

# Fazit

# Glossar

|  |  |
| --- | --- |
| TermBen | Explanation |
| Spieler | Damit ist das GameObject gemeint welches durch einen menschlichen Benutzer gesteuert werden kann |
| Gegner | Damit sind GameObejects gemeint welche durch das Spiel erstellt werden und autonom agieren, |
| GameObjects | Damit sind Objekte der Engine Unity gemeint welche mit verschiedensten Eigenschaften und Aufgaben innerhalb der Engine implementiert wurden. Sie repräsentieren unter anderem den Spieler, die Gegner, die einzelnen Schüsse oder aber auch die Bonusitems |
| Waffen | Als Waffe wird hier ein abstraktes Konzept einer Waffe bezeichnet. Die Waffen im Spiel unterscheiden sich in ihren Eigenschaften wie zum Beispiel der Häufigkeit mit welcher sie abgefeuert werden können oder der Geschwindigkeit der einzelnen Schüsse oder dem Preis der Waffe |
|  |  |
|  |  |
|  |  |