7.10.2019

Produktion digitaler Medien

TITEL TBD



Projektdokumentation

Team C

Christian Dirks  
Benedict Ertelt  
Kim Yuan Kan  
Timo Peters  
Benjamin Schesler

Jens Bendig  
Dipl.-Ing. Björn Arp

Inhaltsverzeichnis

[1. Vorwort 2](#_Toc21345260)

[2. Vorgehensweise 3](#_Toc21345261)

[2.1 Merkmale 3](#_Toc21345262)

[2.2 Vorgehen 4](#_Toc21345263)

[3. Anforderungsanalyse 5](#_Toc21345264)

[3.1 Funktionsanforderungen 5](#_Toc21345265)

[3.2 Randbedingungen 5](#_Toc21345266)

[4. Aufwandsplanung 6](#_Toc21345267)

[4.1 Geplanter Aufwand 6](#_Toc21345268)

[4.2 Tatsächlicher Aufwand 6](#_Toc21345269)

[5. Aufgabenverteilung 7](#_Toc21345270)

[6. Projektablauf 8](#_Toc21345271)

[6.1 Projektziele 8](#_Toc21345272)

[6.2 Projektstrukturplan (Tabellarisch) 9](#_Toc21345273)

[6.3 Projektablaufplan 10](#_Toc21345274)

# 1. Vorwort

Im Rahmen des Moduls „Produktion digitaler Medien“ an der Hochschule Emden-Leer,

geleitet von Prof. Jens Bendig und Dipl. Ing. Björn Arp, sollen mehrere Gruppen der

Größenordnung von bis zu fünf Personen ein digitales Medium produzieren und zur Abgabe

bereitstellen.

Welche Form jenes Medium erhalten soll, ist den Gruppen allerdings selbst überlassen.

Zusätzlich zum Medium soll jede Gruppe eine dazugehörige Präsentation vorbereiten, sowie

eine vollständige Dokumentation anfertigen.

Hierbei soll die Präsentation eine etwaige Dauer von X Minuten einnehmen und die

Dokumentation den Standards einer Projektdokumentation gerecht werden.

Das von Team C, bestehend aus den Mitgliedern Christian Dirks, Benedict Ertelt, Kim Yuan

Kan, Timo Peters und Benjamin Schesler, gewählte Medium ist das eines Videospiels für den

Computer.

Das fertige Produkt soll ein lokales Multiplayer Spiel des „Plattform-Fighter“-Genres werden,

welches von Titeln der Reihe „Super Smash Bros.“ inspiriert ist.

Das Team legt dabei besonders großen Wert auf Planung, Präsentation, Programmierung,

Dokumentation sowie Tests.

# 2. Vorgehensweise

## 2.1 Merkmale

Im Verlaufe des Projekts bedienen wir uns am Vorgehensmodell Scrum, welches folgende

Merkmale und Vorgehensweisen aufweist:

Scrum ist ein Vorgehensmodell, das in der agilen Softwareentwicklung ihre Verwendung

findet. Jenes Modell wird für Projekte, deren Anforderungen sich in ihren Anfangsstadien

entweder unvollständig oder nur in Ansätzen - also mit möglichen

Anforderungserweiterungen - befinden, eingesetzt.

Das Entwicklerteam trifft sich in regelmäßigen Abständen, üblicherweise monatlich oder gar

wöchentlich, um den Fortschritt, sowie überwundene Schwierigkeiten zusammenzufassen

und im Anschluss gegebenenfalls Anforderungen neu zu definieren oder anzupassen. Diese

Treffen werden als Sprints bezeichnet.

Darüber hinaus erfolgt ein täglicher Informationsaustausch über aktuelle Fortschritte oder

Probleme innerhalb des Teams in Form eines Daily Scrums. Mithilfe dieser Besprechungen

werden frühzeitig Ergebnisse sichtbar.

Scrum besteht im Kern aus drei Rollen, dem Product Owner, dem Team und dem Scrum

Master.

Der Product Owner fungiert als Auftraggeber beziehungsweise als Kunde. Er setzt die

Anforderungen an das Produkt, pflegt und priorisiert den Product Backlog und trägt die

Verantwortung für den Gesamtprozess.

Das Team arbeitet als eine Einheit und wird nicht weiter unterteilt, da die Interaktion

zwischen den Entwicklern wichtig ist, um die Arbeit an den Anforderungen aufzuteilen.

Der Scrum Master ist der Berater des Entwicklerteams und sorgt für die reibungslose

Abarbeitung der Anforderungen.

## 2.2 Vorgehen

Recherche

Lasten-/Pflichtenheft erstellen

Aufgabenverteilung

Entwurf erstellen

Implementierungen beginnen

Integration & Tests

Produkt abliefern

# 3. Anforderungsanalyse

## 3.1 Funktionsanforderungen

|  |  |  |
| --- | --- | --- |
| Funktion | Erklärung | Priorität |
| fMultiplayer | Das Spiel soll von mehreren Nutzern zeitgleich spielbar sein | Hoch |
| fCharacterselect | Mehrere Charaktere müssen über ein Menü zur Verfügung stehen | Hoch |
| fMovement | Charaktere müssen sich in mehrere Richtungen bewegen können | Hoch |
| fJump | Charaktere müssen springen können | Hoch |
| fLightAtk | Charaktere müssen einen Standardangriff haben | Hoch |
| fHeavyAtk | Charaktere sollen einen schweren Angriff haben | Standard |
| fBlastzones | Level müssen eine spatiale Limitierung haben | Hoch |
| fDamage | Rückstoßreichweite muss in Abhängigkeit von Schaden sein | Hoch |
| fStocks | Spielerleben müssen angezeigt werden | Hoch |
| fEnd | Eine Bedingung für ein Spielende muss hinzugefügt werden | Hoch |
| fStageselect | Mehrere Level sollen über ein Menü zur Verfügung stehen | Hoch |
| fModes | Verschiedene Modi sollen zur Verfügung stehen | Niedrig |
| fDash | Charaktere sollen in der Lage sein, Sprints zu initiieren | Niedrig |
| fSprint | Charaktere sollen in der Lage sein, Sprints auszuführen | Niedrig |

## 3.2 Randbedingungen

|  |  |  |
| --- | --- | --- |
| Name | Erklärung | Priorität |
| qEngine | Unity | Hoch |
| qProgrammiersprache | C# | Standard |
| qVersioncontrolsystem | Github | Hoch |
| qDokumentation | PDF | Hoch |
| qPräsentation | Produktvorführung | Hoch |

• Niedrig: **Optionale** Funktion

• Standard: Funktion **soll** in das Spiel eingebaut werden

• Hoch: Funktion **muss** in das Spiel eingebaut werden

# 4. Aufwandsplanung

Zwecks Übersicht über den Aufwand wird jener unterteilt in einen geschätzten und einen

tatsächlichen, im Nachhinein ermittelten, Aufwand. Diese tabellarischen

Aufwandsdarstellungen sollen zeigen, mit welchem zeitlichen Aufwand das Team für die

Produktion des digitalen Mediums rechnet, wie der Aufwand Zustande kommt und wie stark

er vom tatsächlichen Aufwand sich unterscheidet.

## 4.1 Geplanter Aufwand

|  |  |  |  |
| --- | --- | --- | --- |
| Aktivität | Opt. | Real. | Pess. |
| Planung | 50 | 55 | 65 |
| Recherche | 40 | 45 | 50 |
| Character Design | 50 | 60 | 70 |
| Character Animation | 90 | 100 | 120 |
| Level Design | 40 | 45 | 55 |
| Menüentwicklung | 25 | 30 | 35 |
| Implementierung | 40 | 45 | 60 |
| Dokumentation | 40 | 45 | 50 |
| Tests | 35 | 45 | 55 |
| Präsentation | 40 | 45 | 55 |
|  |  |  |  |
| **Summe** | 450 | 515 | 615 |
| **Schätzwert** |  | 526.67 |  |

## 4.2 Tatsächlicher Aufwand

|  |  |  |  |
| --- | --- | --- | --- |
| Aktivität | Opt. | Real. | Pess. |
| Planung |  |  |  |
| Recherche |  |  |  |
| Character Design |  |  |  |
| Character Animation |  |  |  |
| Level Design |  |  |  |
| Menüentwicklung |  |  |  |
| Implementierung |  |  |  |
| Dokumentation |  |  |  |
| Tests |  |  |  |
| Präsentation |  |  |  |
|  |  |  |  |
| **Summe** |  |  |  |
| **Schätzwert** |  |  |  |

# 5. Aufgabenverteilung

Programmierung

• Timo Peters

• Christian Dirks

• Benjamin Schesler

Grafik/Leveldesign/Mapping

• Benedict Ertelt

Sound

• Kim Yuan Kan

Animationen

• Benedict Ertelt

• Benjamin Schesler

Dokumentation / Präsentation

• Kim Yuan Kan

TBE

# 6. Projektablauf

## 6.1 Projektziele

1. TBD

## 6.2 Projektstrukturplan (Tabellarisch)

1. Vorarbeit

1.1 Projekt erfassen

1.2 Team organisieren

1.3 Kommunikation einrichten

2. Recherche

2.1 Feedback seitens Kurses wahrnehmen

2.2 Analysieren bestehender Spiele

2.3 Einarbeitung in Unity

2.4 Einarbeitung in Cinema4D/Blender

3. Planung

3.1 Anforderungsanalyse durchführen

3.2 Aufgabenverteilung erstellen

3.3 Zeitaufwandsplan erstellen

3.4 Projektziele / Meilensteine erfassen

3.5 Projektablaufplan ermitteln

3.6 Projektstrukturplan ermitteln

4. Entwurf

4.1 Character Entwürfe erstellen

4.2 Level Design erstellen

4.3 Soundmedium einbinden

4.4 Menüs erstellen

5. Implementierung

5.1 Gamelogik einbinden

5.2 Funktionen einbinden

6. Finalisierung

6.1 Tests durchführen

6.1.1 gegebenenfalls Änderungen / Korrekturen vornehmen

6.2 Endpräsentation vorbereiten

6.1.2 Endpräsentation halten

6.3 Projekt abschließen

## 6.3 Projektablaufplan

## 