Projekt Management Plan Vortex-Tunnel

|  |  |  |
| --- | --- | --- |
| **Version** | **Beschreibung** | **Bearbeiter** |
| 1.0 | Erster Entwurf des Berichts | Marc Nussbaumer |
| 1.1 | Übertragung Risiken | Marc Nussbaumer |
| 1.2 | Übertragung Rahmenplanung und Projektübersicht | Marc Nussbaumer |
| 1.3 | Übertragung der Testfälle und Testprotokolle | Marc Nussbaumer |
| 1.4 | Vorgehensmodell und Projektziele | Marc Nussbaumer |

Inhaltsverzeichnis

[Projektübersicht 3](#_Toc502737153)

[Projektziele 3](#_Toc502737154)

[Organisation & Zuständigkeiten 3](#_Toc502737155)

[Projektführung 4](#_Toc502737156)

[Gewähltes Vorgehensmodell 4](#_Toc502737157)

[Rahmenplan 5](#_Toc502737158)

[Projektkontrolle 6](#_Toc502737159)

[Risikomanagement 6](#_Toc502737160)

[Risiken Meilenstein 1 6](#_Toc502737161)

[Risiken Meilenstein 2 7](#_Toc502737162)

[Risiken Meilenstein 3 8](#_Toc502737163)

[Risiken Meilenstein 4 9](#_Toc502737164)

[Risiken Meilenstein 5 10](#_Toc502737165)

[Definition of Done 11](#_Toc502737166)

[Projektabschluss 11](#_Toc502737167)

[Projekunterstützung 12](#_Toc502737168)

[Tools für Entwicklung 12](#_Toc502737169)

[Testplan 13](#_Toc502737170)

[System-Tests 13](#_Toc502737171)

[Testfälle 13](#_Toc502737172)

[Testprotokolle 17](#_Toc502737173)

[Sprint 1 Testergebnisse 17](#_Toc502737174)

[Sprint 2 Testergebnisse 17](#_Toc502737175)

[Sprint 3 Testergebnisse 18](#_Toc502737176)

[Anhang 19](#_Toc502737177)

[Sprintpläne 19](#_Toc502737178)

[Sprint 1 (26.10.2017 – 08.11.2017) 19](#_Toc502737179)

[Sprint 2 (09.11.2017 – 22.11.2017) 20](#_Toc502737180)

[Sprint 3 (30.11.2017 – 13.12.2017) 20](#_Toc502737181)

# Projektübersicht

## Projektziele

Es gilt mindestens einen Prototyp eines Vortex-Tunnels basierend auf der Aufgabenstellung (Siehe Anhang «Aufgabenstellung\_PAWI\_HS2017\_Gröbelbauer\_Nussbaumer») zu erstellen, welcher mit VR Hardware angesehen werden kann. Basierend auf diesem soll versucht werden mittels eines Versuches die Parameter für eine grössere Effekterzeugung herauszufinden. Abzugeben ist nebst einem lauffähigen Programm ein Benutzerhandbuch, als auch Tutorials bezüglich der Umsetzung von Projekten in Unity. Auch der durchgeführte Versuch mit Resultats Auswertung ist Teil der Abgabe.

## Organisation & Zuständigkeiten

**Teammitglieder**

**Marc Nussbaumer** **Philipp Gröbelbauer**marc.nussbaumer@stud.hslu.ch philipp.groebelbauer@stud.hslu.ch

**Verantwortungsbereiche im Projekt**

Marc Nussbaumer:

* Textur
* Auswertung Versuchsresultate
* Modell (UV-Mapping)
* XML-Mapping

Philipp Gröbelbauer:

* Lichter
* Versuchsszenarien / Aufbau
* Interaktive Doku
* Modell
* XML-Struktur

# Projektführung

## Gewähltes Vorgehensmodell

Die Besonderheit unseres PAWI-Projektes ist, dass es sowohl Komponenten einer Software-Entwicklung als auch einen Forschungs-Charakter enthält. Dadurch fällt die Wahl eines Vorgehensmodelles vergleichsweise schwer. Unser ursprünglicher Lösungsansatz war dabei das Projekt als ein klassisches Software-Projekt mittels SoDa umzusetzen.

Dieser Ansatz wurde am obligatorischen Input zum Vorgehensmodell vom 02.10.2017 den anwesenden Dozenten vorgestellt. Hauptkritik-Punkt war, dass wir «Sprints» hatten welche keine Softwarelösung zur Folge hatten und somit nicht der Definition entsprechen.

In der weiteren Diskussion mit Herrn Jud wurde uns aufgezeigt, dass es legitim ist mehrere Verfahren zu mischen. Hauptziel soll sein, dass Projekt möglichst planbar zu machen und nicht an einem Vorgehensmodell festzuhalten.

Basierend auf dieser Rückmeldung haben wir unser Vorgehensmodell aus einem Forschungsansatz mit Sprints definiert. So wird jegliche Software-/Produkterstellung in einem Sprint gehandhabt. Sämtliche Vorarbeiten werden jedoch fest strukturiert mit Überprüfung durch eine Meilenstein-Sitzung gehandhabt.

Als Besonders zu bezeichnen ist einerseits die Testdurchführungsperiode, welche zwischen zwei Sprints abgelaufen wird und andererseits, dass kein klassisches Sprintbacklog existiert.

Aufgrund der laufenden Änderungen an der Aufgabenstellung durch neue Forschungserkenntnisse wird dieser stattdessen basiert auf der letzten Meilensteinsitzung definiert.

## Rahmenplan

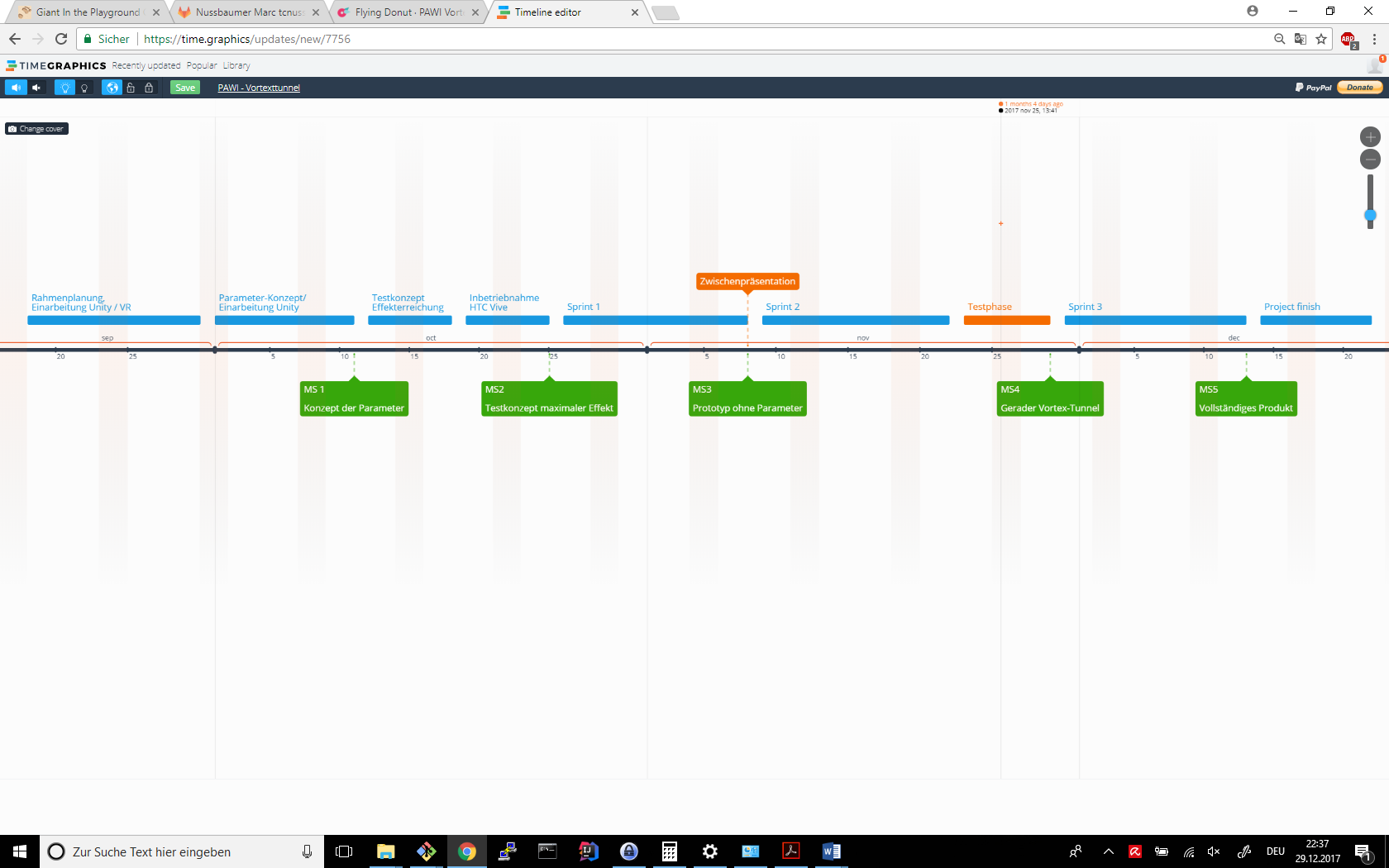


Abbildung Rahmenplan bis Zwischenpräsentation

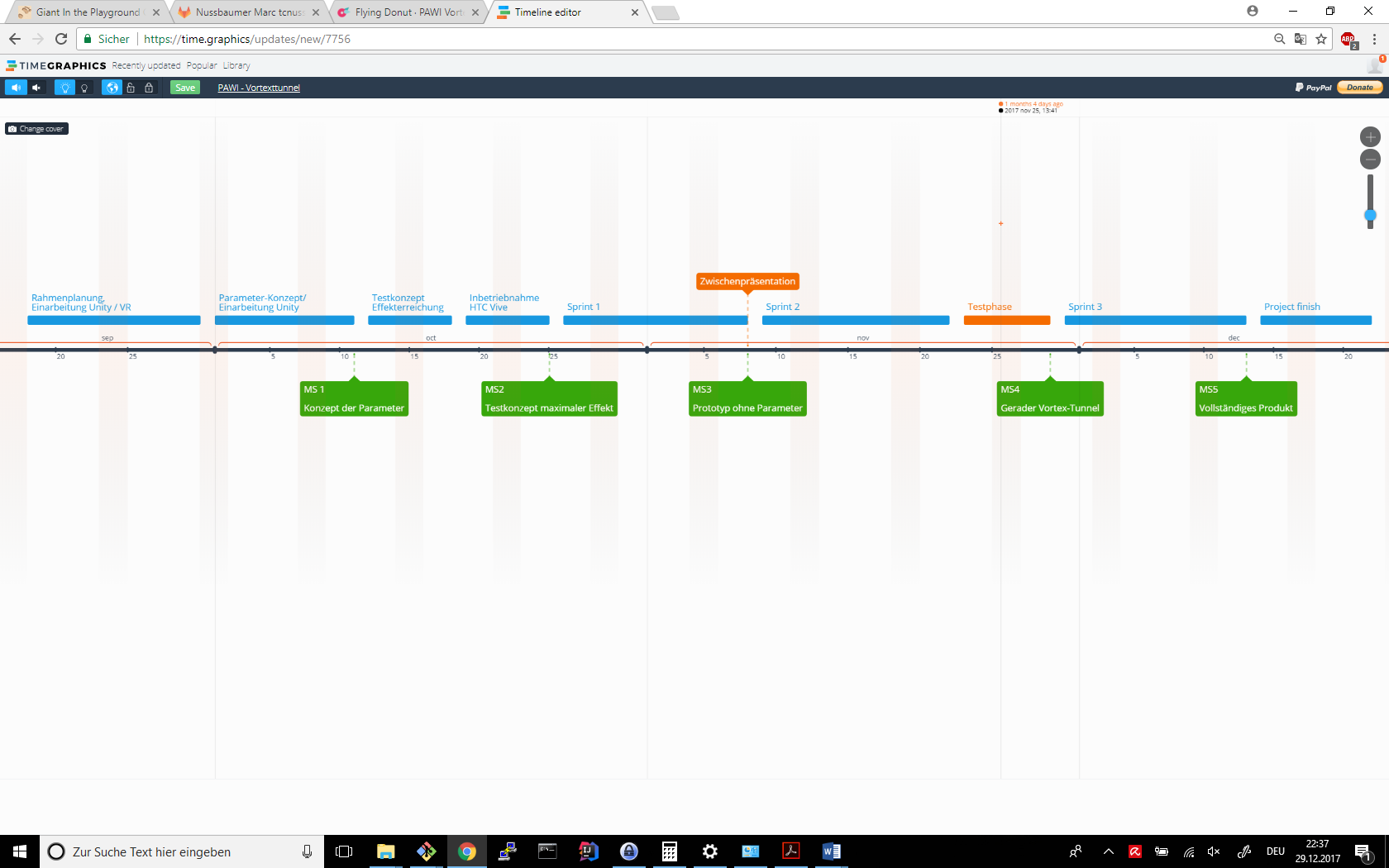


Abbildung 2 Rahmenplan ab Zwischenpräsentation

## Übersicht Meilensteine

|  |  |  |  |
| --- | --- | --- | --- |
| Meilenstein | Ziel | Deliverables | Datum |
| MS1: Konzept der Parameter | Vergleichen mehrerer möglicher Konzepte zur Parametrisierung. Auswahl der geeignetsten Lösung | Lösungskonzept zur Parameter-Anpassung | 11.10.2017 |
| MS2: Testkonzept maximaler Effekt | Ausarbeitung Fragekatalog für Testprobanden. Festlegung des maximalen Effektes. Erstellung Product Backlog | Test zur Überprüfung der Effektstärke Konzept zur Ausführung des Tests mit Probanden  Product Backlog | 25.10.2017 |
| MS3: Prototyp ohne Parameter | Es soll ein Proof-of-Concept Model des Vortex-Tunnels in Unity erstellt werden. Dieses soll als Gesprächsgrundlage für die folgenden Sprints dienen. | Unity-Modellierung des Vortex-Tunnels ohne Parameter-Einstellungen | 08.11.2017 |
| MS4: Gerader Vortex-Tunnel | Das Konzept zur Anpassung der Parameter soll mit dem Proof-of-Concept Model verschmolzen werden um eine parametrisierbare Applikation zu erhalten. | Unity-Modellierung des Vortex-Tunnes mit Einstellungsparametern | 29.11.2017 |
| MS5: Vollständiges Produkt | Abschluss der Entwicklung der Applikation | Vollständige Unity-Applikation | 13.12.2017 |

Tabelle Risikomanagemen

## Projektkontrolle

Mittels FlyingDonouts (siehe Anhang Sprintpläne)

## Risikomanagement

Sämtliche Risiken wurden an jedem Meilenstein neu beurteilt und als eigenes Kapitel in den Meilensteinberichten aufgeführt. Fokus wurde dabei vor allem auf mögliche Beeinflussung von Drittparteien und technischen Problemen gelegt.

Zur einfacheren Lesbarkeit wurden sämtliche Änderungen gegenüber dem letzten Meilenstein immer mit Gelb hinterlegt. Zusätzlich werden sämtliche Änderungen an den Risiken begründet.

### Risiken Meilenstein 1

|  |  |  |  |
| --- | --- | --- | --- |
| Risiko | Eintrittswahrscheinlichkeit | Ausmass | Gegenmassnahmen |
| VR-Ausrüstung ist defekt | 3 | 4 | Simulation kann an PC ausgeführt werden |
| Unity-Framework unterstützt Funktionen nicht | 2 | 8 | Früher Prototyp zur Reduktion der Unbekannten |
| Nicht genügend Testprobanden vorhanden | 5 | 7 | Rekrutierung von Probanden läuft bereits / Test an HSLU |
| Unity-Updates beeinträchtigen Projekt | 6 | 10 | PAWI auf einer bestimmten Unity Version ausführen, welche nicht aktualisiert wird |
| Nicht genügend Know-How für Prototyp | 2 | 9 |  |
| Der Effekt des Gleichgewichts-verlusts tritt nicht ein | 1 | 10 | Früher Prototyp, im Eintrittsfall andere Szenarien zur Täuschung des Gleichgewichtssinnes |
| Legende | 0 = tief; 10 = garantiert | 0 = kein Problem,  10 = PAWI nicht  erfolgreich |  |

Tabelle 2 Risiken Meilenstein 1

### Risiken Meilenstein 2

|  |  |  |  |
| --- | --- | --- | --- |
| Risiko | Eintrittswahrscheinlichkeit | Ausmass | Gegenmassnahmen |
| *VR-Ausrüstung geht kaputt* | ***2*** | *4* | *Simulation kann an PC ausgeführt werden* |
| Unity-Framework unterstützt Funktionen nicht | 2 | 8 | Früher Prototyp zur Reduktion der Unbekannten |
| Nicht genügend Testprobanden vorhanden | 5 | 7 | Rekrutierung von Probanden läuft bereits / Test an HSLU |
| Unity-Updates beeinträchtigen Projekt | ***3*** | 10 | PAWI auf einer bestimmten Unity Version ausführen, welche nicht aktualisiert wird |
| Nicht genügend Know-How für Prototyp | 2 | 9 |  |
| Der Effekt des Gleichgewichts-verlusts tritt nicht ein | 1 | 10 | Früher Prototyp, im Eintrittsfall andere Szenarien zur Täuschung des Gleichgewichtssinnes |
| Legende | 0 = tief; 10 = garantiert | 0 = kein Problem,  10 = PAWI nicht  erfolgreich |  |

Tabelle 3 Risiken Meilenstein 2

**Anmerkungen Risiken**

* Das Risiko, dass die Ausrüstung defekt ist fällt weg, da wir diese am 18.10.2017 erfolgreich in Betrieb nehmen konnten. Es wird ersetzt durch das Risiko, dass die Ausrüstung während unserer PAWI beschädigt wird.
* Das Risiko, dass Unity-Updates das Projekt beeinträchtigen wurde minimiert, indem auf den Entwickler-PCs als auch auf dem VR-PC das gleiche Unity-Framework (Version 2017.2.0f3) installiert wurde.

### Risiken Meilenstein 3

|  |  |  |  |
| --- | --- | --- | --- |
| Risiko | Eintrittswahrscheinlichkeit | Ausmass | Gegenmassnahmen |
| VR-Ausrüstung geht kaputt | 2 | 4 | Simulation kann an PC ausgeführt werden |
| Unity-Framework unterstützt Funktionen nicht | 2 | 8 | Früher Prototyp zur Reduktion der Unbekannten |
| Nicht genügend Testprobanden vorhanden | 5 | 7 | Rekrutierung von Probanden läuft bereits / Test an HSLU |
| Unity-Updates beeinträchtigen Projekt | 3 | 10 | PAWI auf einer bestimmten Unity Version ausführen, welche nicht aktualisiert wird |
| Nicht genügend Know-How für Prototyp | 2 | 9 |  |
| *Es kann nicht auf Ressourcen des Enterpriselab zugegriffen werden* | ***4*** | ***2*** | Änderungen können auf Entwickler-Maschine bestehen bis Problem gelöst wurde |
| ~~Der Effekt des Gleichgewichts-verlusts tritt nicht ein~~ | ***~~5~~*** | ~~10~~ | ~~Früher Prototyp, im Eintrittsfall andere Szenarien zur Täuschung des Gleichgewichtssinnes~~ |
| Legende | 0 = tief; 10 = garantiert | 0 = kein Problem,  10 = PAWI nicht  erfolgreich |  |

Tabelle 4 Risiken Meilenstein 3

**Anmerkungen Risiken**

* Aufgrund der aufgetretenen GIT-Quota-Probleme nehmen wir den Zugriffsverlust auf die Schulressourcen als zusätzliches Risiko auf, welches erneut eintreten könnte
* Der Effekt konnte erfolgreich an der Zwischenpräsentation vorstellt werden und wurde von allen Präsentationsteilnehmern wahrgenommen.

### Risiken Meilenstein 4

|  |  |  |  |
| --- | --- | --- | --- |
| Risiko | Eintrittswahrscheinlichkeit | Ausmass | Gegenmassnahmen |
| VR-Ausrüstung geht kaputt | 6 | *4* | *Simulation kann an PC ausgeführt werden* |
| Unity-Framework unterstützt Funktionen nicht | 2 | 8 | Früher Prototyp zur Reduktion der Unbekannten |
| Nicht genügend Testprobanden vorhanden | 5 | 7 | Rekrutierung von Probanden läuft bereits / Test an HSLU |
| Unity-Updates beeinträchtigen Projekt | 1 | 10 | PAWI auf einer bestimmten Unity Version ausführen, welche nicht aktualisiert wird |
| ~~Nicht genügend Know-How für Prototyp~~ | ~~2~~ | ~~9~~ |  |
| *Es kann nicht auf Ressourcen des Enterpriselab zugegriffen werden* | ***2*** | ***2*** | Änderungen können auf Entwickler-Maschine bestehen bis Problem gelöst wurde |
| Legende | 0 = tief; 10 = garantiert | 0 = kein Problem,  10 = PAWI nicht  erfolgreich |  |

Tabelle 5 Risiken Meilenstein 4

**Anmerkungen Risiken**

* Der Computer für VR stürzte zweimal ab während Prototyp-Vorbereitung. Daher wurde das Risiko defekter Ausrüstung als erhöht eingestuft.
* Unity-Updates werden nicht automatisch ausgeführt und beide Entwickler arbeiten mit derselben Version des Programms.
* Prototyp ist nach Zwischenpräsentation abgenommen, daher fällt das Risiko weg
* Beim Enterpriselab wurden mehr Ressourcen für das Git beantragt und genehmigt. Daher minimiertes Risiko.

### Risiken Meilenstein 5

|  |  |  |  |
| --- | --- | --- | --- |
| Risiko | Eintrittswahrscheinlichkeit | Ausmass | Gegenmassnahmen |
| VR-Ausrüstung geht kaputt | 6 | 4 | Simulation kann an PC ausgeführt werden |
| ~~Unity-Framework unterstützt Funktionen nicht~~ | ~~2~~ | ~~8~~ | ~~Früher Prototyp zur Reduktion der Unbekannten~~ |
| ~~Nicht genügend Testprobanden vorhanden~~ | ~~5~~ | ~~7~~ | ~~Rekrutierung von Probanden läuft bereits / Test an HSLU~~ |
| Unity-Updates beeinträchtigen Projekt | 1 | 10 | PAWI auf einer bestimmten Unity Version ausführen, welche nicht aktualisiert wird |
| *Es kann nicht auf Ressourcen des Enterpriselab zugegriffen werden* | ***2*** | ***10*** | Änderungen können auf Entwickler-Maschine bestehen bis Problem gelöst wurde |
| Legende | 0 = tief; 10 = garantiert | 0 = kein Problem,  10 = PAWI nicht  erfolgreich |  |

Tabelle Risiken Meilenstein 5

**Anmerkungen Risiken**

* Da der Versuch erfolgreich durchgeführt wurde fällt das Risiko mit zu wenig Testprobanden weg
* Da sämtliche Funktionen umgesetzt wurden fällt das Risiko einer Funktion, welche nicht in Unity vorhanden ist weg.
* Da die Abgabe per GitLab beschlossen wurde würde ein Ausfall des Enterpriselabs die Abgabe der PAWI verunmöglichen, daher wurden die Auswirkungen erhöht

## Definition of Done

Ein Sprint gilt für uns als erledigt, wenn nebst den Abnahme-Kriterien der einzelnen Stories folgende Punkte funktional sind:

* Ausserhalb der Programmierumgebung lauffähig  
  Die Simulation soll als \*.exe-Datei gestartet werden können und den Vortex-Tunnel oder ein Anfangs-Szenario darstellen.
* Default-Kamera zeigt korrekte Szene an  
  Beim Start der Simulation soll die Kamera auf den Tunnel oder Anfangsszenario ausgerichtet sein, es sollen keine grafischen Fehler sichtbar sein.
* Darstellung auf HTC Vive  
  Bei Start der Simulation muss diese an einer angeschlossenen HTC Vive dargestellt werden.
* Erfüllung der Best Practices gemäss Occulus Rift  
  Zur Minimierung der «VR-Sickness» muss bei jedem Sprint die Erfüllung der Best Practices nach Occulus-Rift erfüllt werden. (Yao, Heath, Davies, Forsyth, & Mitchell, 2014)

## Projektabschluss

Das finale Produkt wurde den betreuenden Dozenten bereits demonstriert und als funktional bezeichnet. Aus Sicht des Projekt-Teams wurde ist das Projekt somit erfolgreich abgeschlossen. So konnte der Effekt bereits in den ersten Prototypen nachgewiesen werden. Aufgrund der Versuchsdurchführung konnten die Parameter weiter optimiert werden.

# Projekunterstützung

## Tools für Entwicklung

Aufgrund der Warnung von Herr Diehl bezüglich Kompatibilitätsproblemen zwischen verschiedenen Versionen von Unity in bereits durchgeführten Projekten haben wir uns entschlossen nur auf einer vereinbarten Version zu arbeiten um Probleme zu verhindern.

Dies haben wir bei allen eingesetzten Werkzeugen verwendet und die Kompatibilität ist mit den angegebenen Versionen garantiert.

Entwicklungsumgebung:

* Unity
  + Version 2017.2.0f3
* Visual Studio Enterprise 2015
  + Version 14.0.25420.01 Update 3
* Webstorm
  + Version 2017.3

Verwaltung Source-Code:

* Enterpriselab der Hochschule Luzern
  + GitLab: <https://gitlab.enterpriselab.ch/tcnussba/pawi-vortext-tunnel.git>

Notwendige Frameworks:

* .NET-Framework
  + Version 4.7.02556

Projektplanung:

* Sprint-Planung
  + <https://www.flyingdonut.io/app/project/project-id=59f0a012e4b033c5fdd50c6d>
* Zeitstrahl
  + <https://time.graphics>

VR-Framework

* Steam-VR

Eine Besonderheit ist SteamVR, da das Programm nur den Zugriff erlaubt, falls es momentan auf dem neusten Stand ist. Es gibt keine Möglichkeit das Update zu unterbinden oder nur lokal zu arbeiten.

# Testplan

## System-Tests

Unity ist ein Tool zur Erstellung von virtuellen Welten, dabei können Objekten Programme hinzugefügt werden. Hierfür wurde C# verwendet, aufgrund des Unity-Spezifischen Aufbaus sind diese nicht selbständig lauffähig.

Unity erlaubt es nur Log-Nachrichten aufzunehmen um Probleme zu lösen, weitere Debugging-Werkzeuge werden nicht unterstützt.

Aus diesem Grund sind bei der Verwendung von Unity keine Unit-Test möglich, sämtliche Test mussten daher als Systemtests definiert und durchgeführt werden.

## Testfälle

|  |  |
| --- | --- |
| **Nr.** | 1 |
| **Name** | Darstellung des Tunnel-Modells in Unity |
| **Beschreibung** | SketchUp Modell soll in Unity dargestellt werden |
| **Story** | 1 |
| **Vorgehen** | 1. Unity mit Projekt starten 2. Neue Szene generieren 3. In \*.dae-Format exportiertes SketchUp-Modell per Drag&Drop in Asset-Order von Unity importieren 4. Das Modell per Drag&Drop in die neue Szene importieren 5. Modell in Sichtfeld der Kamera positionieren 6. Auf «Play»-Knopf drücken |
| **Ergebnis** | Das SketchUp-Modell soll auf der Game-Preview von Unity dargestellt werden |

|  |  |
| --- | --- |
| **Nr.** | 2 |
| **Name** | Bewegende Spot-Lichter |
| **Beschreibung** | Spot-Lichter sollen generiert und dargestellt werden |
| **Story** | 2 |
| **Vorgehen** | 1. Unity mit Projekt starten 2. Neue Szene generieren 3. Prefab «Pointlight» per Drag&Drop in die neue Szene importieren 4. In dem am Prefab angehängten Script die Anzahl Lichter und Distanz auf 250 setzen 5. Auf «Play»-Knopf drücken |
| **Ergebnis** | Es sollen 250 Lichter auf einer Distanz von 250 Meter dargestellt werden |

|  |  |
| --- | --- |
| **Nr.** | 3 |
| **Name** | Bewegende Textur |
| **Beschreibung** | Textur des Tunnels soll sich bewegen |
| **Story** | 3 |
| **Vorgehen** | 1. Unity mit Projekt starten 2. Szene «MS3\_Prototyp\_Texture» laden 3. Auf «Play»-Knopf drücken |
| **Ergebnis** | Es soll ein Modell eines Tunnels mit drehender Textur dargestellt werden |

|  |  |
| --- | --- |
| **Nr.** | 4 |
| **Name** | Darstellung einer Unity-Szene in VR-Hardware |
| **Beschreibung** | Unity-Szene soll mithilfe des SteamVR-Plugins auf HTC-Vive dargestellt werden |
| **Story** | 4 |
| **Vorgehen** | 1. VR-Hardware gemäss Hersteller konfigurieren 2. Unity mit Projekt starten 3. Neue Szene generieren mit Steam VR Prefabs [CameraRig], [SteamVR] 4. Würfel in Sichtfeld der Kamera vom Prefab [CameraRig] positionieren 5. Auf «Play»-Knopf drücken |
| **Ergebnis** | Der Würfel soll gleich auf der VR-Hardware dargestellt werden wie auf der Game-Preview von Unity |

|  |  |
| --- | --- |
| **Nr.** | 5 |
| **Name** | Bewegung von Point-Lichter |
| **Beschreibung** | Point-Lichter sollen sich in einem Tunnelmodell drehen |
| **Story** | 7 |
| **Vorgehen** | 1. Unity mit Projekt starten 2. Szene «MS3\_Prototyp» laden 3. Auf «Play»-Knopf drücken |
| **Ergebnis** | Es soll ein Modell eines Tunnels mit bewegenden Lichtern dargestellt werden |

|  |  |
| --- | --- |
| **Nr.** | 6 |
| **Name** | Teleportation im Intro-Szenario |
| **Beschreibung** | Benutzer soll an einem bestimmten Punkt im Intro-Szenario in den Vortex-Tunnel teleportiert werden |
| **Story** | 8 |
| **Vorgehen** | 1. VR-Hardware gemäss Hersteller einrichten 2. Unity mit Projekt starten 3. Szene «Intro\_Scene» laden 4. Auf «Play»-Knopf drücken 5. Mit VR-Hardware zum leuchtenden Punkt laufen |
| **Ergebnis** | Benutzer wird zum Vortex-Tunnel teleportiert |

|  |  |
| --- | --- |
| **Nr.** | 7 |
| **Name** | Laden von XML-Werten in Unity |
| **Beschreibung** | Tunnel soll gemäss XML-Konfiguration geladen werden |
| **Story** | 9 & 10 |
| **Vorgehen** | 1. Unity mit Projekt starten 2. Szene «MS4\_Parameter» laden 3. XML-Werte gemäss Benutzerhandbuch setzten, dabei genau 2 verschiedene Abschnitte definieren 4. Auf «Play»-Knopf drücken |
| **Ergebnis** | Es sind genau 2 verschiedene Abschnitte vorhanden, welche den eingegebenen Parametern entsprechen |

|  |  |
| --- | --- |
| **Nr.** | 8 |
| **Name** | Überprüfung der UV-Map |
| **Beschreibung** | Überprüfung ob unendliche Textur ohne grafische Fehler durchläuft |
| **Story** | 11 |
| **Vorgehen** | 1. Unity mit Projekt starten 2. Szene «MS4\_Parameter» laden 3. XML-Werte gemäss Standard-Szenario laden 4. Auf «Play»-Knopf drücken |
| **Ergebnis** | Textur dreht sich im Tunnel ohne ein Anfang oder Ende zu haben |

|  |  |
| --- | --- |
| **Nr.** | 9 |
| **Name** | Anpassung der Stegtransparenz |
| **Beschreibung** | Stegtransparenz soll aufgrund der XML-Konfiguration gesetzt werden |
| **Story** | 12 |
| **Vorgehen** | 1. Unity mit Projekt starten 2. Szene «MS4\_Parameter» laden 3. In der XML-Datei die Stegtransparenz auf 0.5 setzten 4. Auf «Play»-Knopf drücken |
| **Ergebnis** | Textur des Stegs sollte halbtransparent dargestellt werden |

|  |  |
| --- | --- |
| **Nr.** | 10 |
| **Name** | Dynamisches Stegmuster |
| **Beschreibung** | Stegmuster soll aufgrund der XML-Konfiguration gesetzt werden |
| **Story** | 13 |
| **Vorgehen** | 1. Unity mit Projekt starten 2. Szene «MS4\_Parameter» laden 3. In der XML-Datei den Wert für die Steg-Textur auf eine beliebige PNG-Datei ändern 4. Auf «Play»-Knopf drücken |
| **Ergebnis** | Steg sollte nun aus Wiederholungen der PNG-Datei bestehen |
| **Nr.** | 11 |
| **Name** | Stegdimensionen |
| **Beschreibung** | Stegdimensionen sollen aufgrund der XML-Konfiguration gesetzt werden |
| **Story** | 15 |
| **Vorgehen** | 1. Unity mit Projekt starten 2. Szene «MS4\_Parameter» laden 3. In der XML-Datei die Werte für Höhe auf -0.5 und die Stegbreite auf 0.5 setzen 4. Auf «Play»-Knopf drücken |
| **Ergebnis** | Steg sollte halb so breit wie normalerweise sein und näher an der Kamera dargestellt werden |

|  |  |
| --- | --- |
| **Nr.** | 12 |
| **Name** | Handgeländer-Konfigurationen |
| **Beschreibung** | Entfernen der Handgeländer über XML |
| **Story** | 16 |
| **Vorgehen** | 1. Unity mit Projekt starten 2. Szene «MS4\_Parameter» laden 3. In der XML-Datei den Wert für Geländer auf «off» setzten 4. Auf «Play»-Knopf drücken |
| **Ergebnis** | Es sollte der Steg ohne Handgeländer dargestellt werden |

## Testprotokolle

### Sprint 1 Testergebnisse

Testdurchführung mit Commit: c5b31ddfc4f9acbd3fdbb9ef450bc94ad627c826

|  |  |  |  |
| --- | --- | --- | --- |
| Test-Nummer | Datum | Status | Kommentar |
| 1 | 08.11.2017 | OK |  |
| 2 | 08.11.2017 | Nicht OK | Lichter verschwinden nach ca. 30 Unity-Metern |
| 3 | 08.11.2017 | OK |  |
| 4 | 08.11.2017 | OK |  |
| 5 | 08.11.2017 | OK |  |
| 6 | 08.11.2017 | OK |  |

Testdurchführung mit Commit: 34850ef3523da4382c77588168ebf9d47e7d1c23

|  |  |  |  |
| --- | --- | --- | --- |
| Test-Nummer | Datum | Status | Kommentar |
| 1 | 08.11.2017 | OK |  |
| 2 | 08.11.2017 | Nicht OK | Reproduzierbarer Unity-Bug mit den verschwindenden Lichtern wurde festgestellt |
| 3 | 08.11.2017 | OK |  |
| 4 | 08.11.2017 | OK |  |
| 5 | 08.11.2017 | OK |  |
| 6 | 08.11.2017 | OK |  |

Es wurde festgestellt, dass Unity Spot-Lights automatisch nach einer Distanz von ca. 30 Metern automatisch ausblendet. Das Problem wurde an der Präsentation aufgezeigt und es wurde beschlossen den Ansatz mit Spotlights nicht weiter zu verfolgen.

### Sprint 2 Testergebnisse

Testdurchführung mit Commit: 2c2b5ff6db87cd05fa6e126f8d2258c755e73e6e

|  |  |  |  |
| --- | --- | --- | --- |
| Test-Nummer | Datum | Status | Kommentar |
| 1 | 22.11.2017 | OK |  |
| 4 | 22.11.2017 | OK |  |
| 6 | 22.11.2017 | OK |  |
| 7 | 22.11.2017 | OK |  |
| 8 | 22.11.2017 | OK |  |
| 9 | 22.11.2017 | OK |  |

Die Testfälle 2;3 und 5 fallen aufgrund der Entscheidung der Verschmelzung der Prototypen weg und werden nicht mehr getestet.

### Sprint 3 Testergebnisse

Testdurchführung mit Commit: 7a7999dacdc8f5cbff02f9e16f8d59d166b7dcda

|  |  |  |  |
| --- | --- | --- | --- |
| Test-Nummer | Datum | Status | Kommentar |
| 1 | 13.12.2017 | OK |  |
| 4 | 13.12.2017 | OK |  |
| 7 | 13.12.2017 | OK |  |
| 8 | 13.12.2017 | OK |  |
| 9 | 13.12.2017 | OK |  |
| 10 | 13.12.2017 | OK |  |
| 11 | 13.12.2017 | OK |  |
| 12 | 13.12.2017 | OK |  |

Das Intro-Szenario wird besser durch das StandBy-Szenario von SteamVR gelöst, aufgrund dessen fällt der zugehörige Testfall 6 weg.

# Anhang

## Sprintpläne

Als Instrument für Projekt- und Produktmanagement kommt die Plattform «Flying Donut» zum Einsatz. Unter https://www.flyingdonut.io/app/project/project-id=59f0a012e4b033c5fdd50c6d können alle Artefakte zum Projekt angesehen werden.

### Sprint 1 (26.10.2017 – 08.11.2017)

In Sprint 1 steht die Erstellung drei verschiedener Prototypen Vordergrund. Dazu werden nebst der konkreten Implementation der drei verschiedenen Varianten auch ein Tunnelmodell und die Anbindung der VR-Hardware benötigt. Ziel ist es den Dozenten die möglichen Lösungsansätze zu präsentieren und anschliessend über das weitere Vorgehen zu entscheiden.

|  |  |  |
| --- | --- | --- |
| **Story ID** | **Beschreibung** | **Akzeptanzkriterien** |
| 1 | Erstellung eines Tunnel-Modells in einem Grafikprogramm | Modell soll in Unity importiert werden können Modell soll als Tunnel dargestellt werden |
| 2 | Variante 1: Bewegung im Tunnel durch drehende Spot-Lichter simulieren | Bewegung des Tunnels wird durch das Unity-Element «Spotlight» realisiert  Elemente werden durch Code generiert |
| 3 | Variante 2: Bewegung im Tunnel durch drehende Textur simulieren | Bewegung des Tunnels wird durch Verschiebung der Textur des Tunnel-Modells erreicht  Verschiebung findet durch Code statt |
| 4 | Unity-Projekt in VR zugänglich machen | HTC Vive stellt die Szene in Unity korrekt dar |
| 7 | Variante 3: Bewegung des Tunnels durch Point-Lichter simulieren | Bewegung des Tunnels wird durch das Unity-Element «Pointlight» realisiert  Elemente werden durch Code generiert |

### Sprint 2 (09.11.2017 – 22.11.2017)

Es geht in Sprint 2 vor allem darum die Entscheidung die Prototypen Pointlight und Textur zu verschmelzen. Zusätzlich soll eine Konfiguration des Tunnels durch XML-Dateien möglich sein um einen Versuchsaufbau zu ermöglichen. Besonders ist hierbei die Entscheidung den Tunnel aus mehreren Abschnitten zusammensetzen zu können.

|  |  |  |
| --- | --- | --- |
| **Story ID** | **Beschreibung** | **Akzeptanzkriterien** |
| 8 | Erstellung Intro-Szenario zur Angewöhnung an VR | Benutzer soll einen Lichtpunkt sehen  Betreten des Lichtpunktes soll an den Anfang des Vortex-Tunnels teleportieren |
| 9 | Laden XML-Werte in Unity | Werte können in Unity-Scripts verwendet werden  Datentypen werden korrekt interpretiert |
| 10 | Dynamische Tunnelgenerierung anhand XML-Konfiguration | In XML können mehrere Abschnitte definiert werden  Sämtliche XML-Parameter werden beachtet |
| 11 | Erstellung UV-Map für Tunnel-Modell | Wiederholende Textur hat keinen sichtbaren Anfangs- und Endpunkt |

### Sprint 3 (30.11.2017 – 13.12.2017)

Sprint 3 basiert auf den Erkenntnissen des durchgeführten Versuches vom 29. November 2017. Daher wurde der Fokus stark auf die Parametrisierung des Stegs, dem häufigsten Rückmeldepunkt, gelegt.

|  |  |  |
| --- | --- | --- |
| **Story ID** | **Beschreibung** | **Akzeptanzkriterien** |
| 12 | Anpassung der Stegtransparenz | Transparenz des Stegs soll in XML-Wert definiert werden können |
| 13 | Stegmuster dynamisch festlegen | Externe Bilddatei soll als Textur für den Steg verwendet werden können |
| 14 | Steg-Dimensionen per XML definieren | Abstand zur Kamera, Breite sollen in XML definiert werden können |
| 15 | Handgeländer per XML anpassen | Sichtbarkeit der Handgeländer soll in XML konfigurierbar sein |