

Prüfungsleistung im Modul

Softwareprojekt

vorgelegt der Fakultät Bau-Wasser-Boden
im
Studiengang Bachelor Angewandte Informatik
von
Frederik Grünewald
Matrikelnummer 70453004

Betreuer/in:

Prof. Dr.-Ing. Bernd-Uwe Rogalla

Sommersemester 2020

Inhaltsverzeichnis

Vorwort	3
Dokumente	3
Anforderungen	3
Entwicklungsmodell.....	4
Produktvision.....	4
Technologie	4
Architektur.....	5
Klassendiagramme	5
Use Cases.....	5
Mockups	5
Projektplan	6
Projektablauf	7
Projektmanagement.....	7
Aufwandseinschätzung.....	7
Stundenplanung	7
Benötigte Stunden.....	7
Kanban Ziele	7
Meilensteine.....	8
Produktboard	9
Testdurchführung.....	9
Testaufbau.....	9

Vorwort

In diesem Dokument finden Sie zusammenfassend die für das Projekt angefertigten Unterlagen. Dieses Dokument enthält nicht alle für das Projekt angefertigten Dokumente. Für die kompletten Dokumente folgen Sie bitte der Ordnerstruktur.

Dokumente

Anforderungen

Dies stellt die Prüfungsleistung Softwareprojekt dar. Da ich das Projekt allein entwickle, ist der Umfang dementsprechend angelegt. Dieses Projekt umfasst eine Applikation, mit der es möglich ist, verschiedene Suchalgorithmen mithilfe der GameEngine Unity Grafisch darzustellen.

A001: Menü

- Der User hat die Möglichkeit verschiedene Abteilungen der Applikation mithilfe eines Hauptmenüs zu erreichen.

A002: Settings

- Der User hat die Möglichkeit über Das Hauptmenü „Settings“ zu erreichen und dort grundlegende Einstellungen der Applikation zu verändern. (Fenstermodus, Qualität)

A003: HowItWorks

- Der User hat die Möglichkeit über das Hauptmenü „HowItWorks“ zu erreichen und dort grundlegende Informationen über die Suchalgorithmen angezeigt zu bekommen. (Wikipedia Artikel)

A004: SearchAlgorithms

- Der User hat die Möglichkeit über das Hauptmenü „SearchAlgorithms“ zu erreichen und dort zur Simulationsoberfläche zu gelangen, in der die Simulation ausgeführt wird.

A005: Simulationsdaten

- Der User hat die Möglichkeit in der Simulationsansicht Informationen über den Ablauf der Simulation ablesen zu können. (Zeit, Länge)

A006: Simulationseinstellungen

- Der User hat die Möglichkeit in der Simulationsansicht Einstellungen wie den Suchalgorithmus, die Geschwindigkeit oder weitere Anzeigeeinstellungen zu verändern.

Entwicklungsmodell

Als Entwicklungsmodell für die Unity_Pathfinder Anwendung wurde ein Agiles Verfahren auf der Basis von Kanban gewählt. Da dieses Projekt jedoch nur von einer Person durchgeführt wird fallen einige Dokumente, sowie die normalerweise stattfindenden Meetings weg.

Das Kanban Board wurde mithilfe von Trello realisiert. Dabei wurden die in Anforderungen formulierten Anforderungen in kleinere Schritte unterteilt und Schritt für Schritt umgesetzt.

Es wurde bei der Durchführung des Entwicklungsmodells darauf geachtet, dass Parallelarbeit die als Einzelperson schwer zu vermeiden ist, so gering wie möglich gehalten wurde.

Die Einzelschritte, die sich aus den Anforderungen ergeben haben, können den Trello Screenshots in dem Ordner Projektmanagement entnommen werden.

Produktvision

Für die Produktvision wurde ein „VisionBoard“ nach R. Pichler angefertigt. Sehen Sie sich dafür den Ordner Produktvision in dem Ordner „Documents“ an.

Technologie

Der Unity_Pathfinder wird in der Unity Engine 2019.3.14f1 entwickelt.

Als Programmiersprache wird C# verwendet.

Die Entwicklung bezieht sich nur auf das Betriebssystem Windows 10.

Als Erweiterung für die Unity Engine wird „TextMeshPro“ für bessere und schärfere Textdarstellung verwendet.

Architektur

Für die Umsetzung des Unity_Pathfinder wurde die MVC (Model View Controller) Architektur gewählt.

Die MVC Architektur ist eine sehr beliebte Architektur bei Projekten mit der Unity Engine und ermöglicht das schrittweise Erweitern der Software, um gegebenenfalls neue Features zu implementieren und eine hohe Wiederverwendbarkeit des Codes oder einzelner Codebausteine zu garantieren.

Der Aufbau der MVC Architektur im Unity_Pathfinder gestaltet sich wie folgt:

Model:

- GameData
 - o Node
 - o Graph
 - o MapData

View:

- UserInterface
 - o NodeView
 - o GraphView
 - o Menus

Controller:

- Controller
 - o Game Controller

Klassendiagramme

Die Klassendiagramme finden Sie im Ordner „Documents“ unter „ClassDiagram“

Use Cases

Die UseCases finden Sie im Ordner „Documents“ unter „UseCase“

Mockups

Die Mockups finden Sie im Ordner „Documents“ unter „Mockups“

Projektplan

Projektziel:

Das Ziel des Projektes ist es, eine Anwendung zu schaffen, die grafisch aufzeigt wie Suchalgorithmen in einem Koordinatensystem suchen. Die Applikation soll zum Verständnis dieser Algorithmen beitragen, die sonst relativ abstrakt theoretisch erklärt werden.

Projektumfang:

Zum Verwirklichen dieses Projektes muss eine Applikation in Unity geschrieben werden, die auf einer Fläche in einem dreidimensionalen Projekt eine TileMap bzw. ein Koordinatensystem anlegt. Jede einzelne Tile, jeder Punkt in dem Koordinatensystem erhält Hintergrundvariablen, um zu wissen ob der Punkt begehbar ist oder nicht. Dafür muss Unity grundlegend erlernt werden und dessen Funktionen so weitgehend verstanden werden, dass diese für die Verwirklichung des Projekts genutzt werden können. Des Weiteren müssen preFabs und Grafiken für die Menüs und das Overlay angefertigt werden.

Terminierung:

Das Projekt endet offiziell mit der Abgabe am 30.07.2020. Genauere Einteilungen in Meilenstein-Phasen sind dem Timeline Diagramm zu entnehmen.

Details aller Relevanten Projektkomponenten:

Kommunikation:

- Da es sich um eine Einzelarbeit handelt muss intern nicht kommuniziert werden. Für die Unterredungen mit dem Betreuer werden E-Mail für Schriftverkehr und Discord für Voice und Videoübertragungen verwendet. Die Kommunikation findet zur Absprache und bei Problemen statt.

Ressourcen:

- Die verwendete Hardwareressource ist der Computer des ausführenden Entwicklers. Die Softwareressourcen sind die Unity Engine 19.04. Für das Projekt selbst Visual Studio 2019, für das Editieren des Programmcodes und Photoshop CC für das Anfertigen von Grafiken.

Risiken, Annahmen, Abhängigkeiten und Einschränkungen:

- Da es sich um ein Einzelprojekt handelt, mussten keine Absprachen mit Gruppenteilnehmer abgehalten werden. Es besteht keine Abhängigkeit von Fremdkräften, außer die Zustimmung des Betreuers.

Qualitätskontrolle:

- Es wird nach Clean Code Regeln gearbeitet und nach jedem Schritt des Kanban Planes vor dem Commit auf Github eigenständig auf Funktion getestet.

Genehmigung:

- Der Betreuer muss das Projekt vorab genehmigen.

Projektablauf

Der Projektablauf kann in dem Ordner „Documents“ in dem Unterordner „Projektplan“ in dem Timeline.pdf betrachtet werden, jedoch geben auch die Github Commits Auskunft über den Ablauf des Projekts.

Projektmanagement

Aufwandseinschätzung

Stundenplanung

Die Stundenplanung finden Sie im Ordner „Projektmanagement“ im Unterordner „Aufwand“

Benötigte Stunden

Die benötigten Stunden finden Sie im Ordner „Projektmanagement“ im Unterordner „Aufwand“

Kanban Ziele

Die Kanban Ziele wurde vor Beginn des Projekts wie folgt gewählt. Die Reihenfolge sollte nach Möglichkeit eingehalten werden, Parallelarbeit wurde so gut es geht vermieden.

- Create Concept for Application
- Create UseCase
- Create Class Diagram
- Create Mockup
- Create Unity Project
- Create Map Grid
- Method(Draw in Map Grid)
- Method(Breadth First Search)
- Method(Best First Search)
- Method(Dijkstra Algorithm)
- Method(A* Search)
- Create Main Menu
- Create Settings Menu
- Create Wiki Menu
- Design Main Menu

- Design and Populate Settings Menu
- Design and Populate Wiki
- Create Ingame Overlay
- Functionality for Ingame Overlay
- Design Ingame Overlay
- Test Settings Menu
- Test Main Menu
- Test Wiki
- Test Ingame Overlay
- Test Search Algorithms
- Search for Bugs
- Documentation

Meilensteine

Da mit einem Kanban Ansatz gearbeitet wurde, gab es bereits viele kleine Ziele, die erreicht werden sollten. Jedoch wurden die kleinen Ziele noch zu größeren Meilensteinen zusammengefasst, die nach bestimmten Zeiträumen erreicht werden sollten, um ein Vorankommen der Entwicklung zu garantieren. Sollten die Meilensteine nicht erreicht worden sein, so dürfen keine neuen Ziele angelegt bzw. verschoben werden.

Meilenstein 1. Projektplanung und Anlegen des Projekts

- Create Concept for Application
- Create UseCase
- Create Class Diagram
- Create Mockup
- Create Unity Project
- Create Map Grid

Meilenstein 2. Entwickeln des Grids

- Method(Draw in Map Grid)

Meilenstein 3. Implementieren der Suchalgorithmen

- Method(Breadth First Search)
- Method(Best First Search)
- Method(Dijkstra Algorithm)
- Method(A* Search)

Meilenstein 4. Implementieren der Oberflächen und Menüs

- Create Main Menu
- Create Settings Menu
- Create Wiki Menu
- Design Main Menu
- Design and Populate Settings Menu
- Design and Populate Wiki
- Create Ingame Overlay
- Functionality for Ingame Overlay
- Design Ingame Overlay

Meilenstein 5. Testen der Applikation

- Test Settings Menu
- Test Main Menu
- Test Wiki
- Test Ingame Overlay
- Test Search Algorithms
- Search for Bugs

Produktboard

Das Produktboard finden Sie im Ordner „Projektmanagement“ im Unterordner „Produktboard“

Testdurchführung

Testaufbau

Als Testverfahren für den Unity_Pathfinder wurde ein Crowdfunding basierter Ansatz verfolgt, bei dem einzelne Tester einen Testbogen bekommen, den sie abarbeiten und dokumentieren was ihnen aufgefallen ist, welche Hindernisse sie behindert haben und welche Funktionen in welchem Rahmen erfolgreich abgelaufen sind.

Es wurde entschieden einen solchen Testansatz zu wählen, da dieser in der Spieleindustrie durchaus üblich ist und viele auf Unity beruhende Programmteile durch Unittests nicht abgedeckt werden können.

Die Testbögen umfassen folgende Tests und Abdeckungen:

- Die Tester führen Abläufe im Programm durch die mehrere bzw. alle Komponenten der Software umfassen und bei Versagen oder fehlerhafter Funktion keinen vollständigen Ablauf des Testszenarios zulassen.
- Die Tester führen Abläufe im Programm durch die mehrere bzw. alle Schnittstellen der Software umfassen und bei nicht korrekter Übermittlung von

Daten über die Schnittstellen keinen vollständigen Ablauf des Testszenarios zulassen.

- Die Tester führen Abläufe im Programm durch welche die gesamte Software mit all ihren Bausteinen benötigt, um ein positives Testergebnis zurückzuliefern.
- Die Tester installieren / öffnen die Software auf ihren eignen Rechnern und führen die Tests dort aus, um zusätzlich die Funktion auf Fremdsystemen zu garantieren.
- Die Tester geben ihr persönliches Feedback zur Applikation und können Verbesserungsvorschläge bezüglich UX/UI abgeben.

Informationen über die Tester und die ausgefüllten Testbögen finden Sie im Ordner „Testdurchführung“ im Unterordner „Tester“