

Diplomarbeit



Unity Projectmanagement Software

Studiengang zum
Dipl. Informatiker HF Applikationsentwicklung
an der TEKO Bern

Klasse:
B-TIN-19-T-a

Eingereicht von
Sascha Dubois

24. Oktober 2022

Prüfungsexperte:

Herr Jürg Hess

Inhaltsverzeichnis

1	Lebenslauf	1
2	Management Summary	2
3	Initialisierung	3
3.1	Planung	4
3.1.1	Grobplanung	5
3.1.2	Initialisierung	5
3.1.3	Analyse	6
3.1.4	Konzept	6
3.1.5	Umsetzung	7
3.1.6	Deployment	8
4	Analyse	9
4.1	Ausgangslage	9
4.1.1	Analyse von bestehenden Lösungen	9
4.2	Stakeholder	14
4.2.1	Stakeholdergewichtung	15
4.3	Ziele	16
4.3.1	Projektmanagement Ziele	16
4.3.2	Projekt spezifische Ziele	16
4.3.3	Long-Term Ziele	16
4.4	Anforderungen	17
4.4.1	Applikation	17
4.4.2	In Unity integriertes Benutzerinterface	17
4.4.3	Nicht funktionale Anforderungen	17
4.5	Abgrenzung	18
4.6	Risikomanagement	19
4.6.1	Risikoerkennung	19
4.6.2	Risikogewichtung	19
4.6.3	Massnahmen	20
4.7	Wirtschaftlichkeit	21
5	Konzept	22
5.1	Use Case Diagramme	22
5.1.1	Use Case Übersicht	22
5.1.2	Projekt hinzufügen	23
5.1.3	Projekt verwalten	24
5.1.4	Includes von „Projekt verwalten“	25
5.1.5	Als Entwickler registrieren	29
5.1.6	Task hinzufügen (Applikation)	30
5.2	Klassendiagramm	31
5.2.1	Projekt Klassenbeschreibung	31
5.3	Sequenzdiagramme	32
5.3.1	Laden der lokalen Daten beim Starten der Applikation	32
5.3.2	Projekt Hinzufügen	33
5.3.3	Laden der Projekte	34
5.3.4	Als Entwickler registrieren	35
5.4	State Diagramme	36
5.5	Systemarchitektur	37
5.6	Mockup	38

6	Realisierung	39
6.1	Entwicklungsumgebung	39
6.2	Codeconventions	39
7	Anhang	41
7.1	Projektantrag Diplomarbeit	a

1 Lebenslauf

2 Management Summary

Themenbeschreibung

Entwicklung einer Open-Source Projektmanagement-Applikation für Videospiel-Einzelentwickler und kleine Entwicklerteams welche mit Unity3D arbeiten. Dabei sollen die Wichtigsten Funktionen direkt in Unity selber verfügbar sein. Kunde Potenzielle Kunden sind alle Einzelentwickler / Entwicklerteams, welche mit Unity3D arbeiten. Erfolgskriterien - Eine WPF-Applikation wurde den Anforderungen (siehe: Ziele und Anforderungen) entsprechend entwickelt. - Ein in Unity integriertes Benutzerinterface (Custom Editor) in welchem Aktivitäten zum aktuellen Projekt angesehen/bearbeitet werden können wurde den Anforderungen (siehe: Ziele und Anforderungen) entsprechend entwickelt. - Es Liegt eine vollständige Dokumentation des Projektes vor - Die Applikation ist als Open-Source Software lizenziert und zum Download verfügbar

3 Initialisierung

3.1 Planung

Es wird mit einer abgewandelten Form des Wasserfallmodells geplant. Dabei wird das Projekt in 5 Phasen aufgeteilt:

1. Initialisierung
2. Analyse
3. Konzept
4. Umsetzung / Realisierung
5. Veröffentlichung

In den einzelnen Phasen werden Tasks geplant, welche eine geschätzte Dauer zur umsetzung haben. Anhand dieser Schätzung wird die Dauer der einzelnen Phasen geplant. Zusätzlich wird pro Phase ein Buffer eingeplant.

Um das Projekt zu planen habe ich mir zuerst die einzelnen Phasen in einer Gantt-Diagramm-form ausgelegt. Anschliessend bin ich Phase für Phase durchgegangen un habe mir Tasks erstellt, anhand der Schätzung der Dauer habe ich überprüft, ob die Phasenaufteilung passt.

Die Tasks sind in anzahl Tagen geplant, die länge der Tasks ist jedoch auch von meinem Zeitkontingent an den entsprechenden Tagen abhängig, so kann z.B. ein 6h Task 3 Tage andauern, da ich nur 2h pro Tag verplanen konnte. in der Woche vom 17.10.2022 habe ich eine Intensiv Woche geplant. In dieser Woche habe ich die Möglichkeit vollzeit an der Diplomarbeit zu arbeiten. Daher ist in dieser Woche auch der Hauptteil der Umsetzung geplant. Nach dem ich mir die Gantt-Diagramme gemacht und die einzelnen Tasks tabellarisch aufgelistet habe. Habe ich mir in einer Zeit-Tracking App (Toggle) alle Tasks angelegt. In Toggle erfasse ich somit die geleistete Arbeitszeit zu jeweiligem Task.

Dies soll mir ermöglichen verzögerungen schnell erkennen zu können und entsprechende Massnahmen einzuleiten. Zum Controlling notiere ich mir die effektiv benötigte Zeit pro Task und kann somit das Zeitdelta berechnen.

3.1.1 Grobplanung

Um die Planung etwas übersichtlicher zu gestalten, hab ich eine Grobplanung mit den einzelnen Phasen erstellt. Jede Phase wird in den folgenden Abschnitten noch detaillierter geplant.

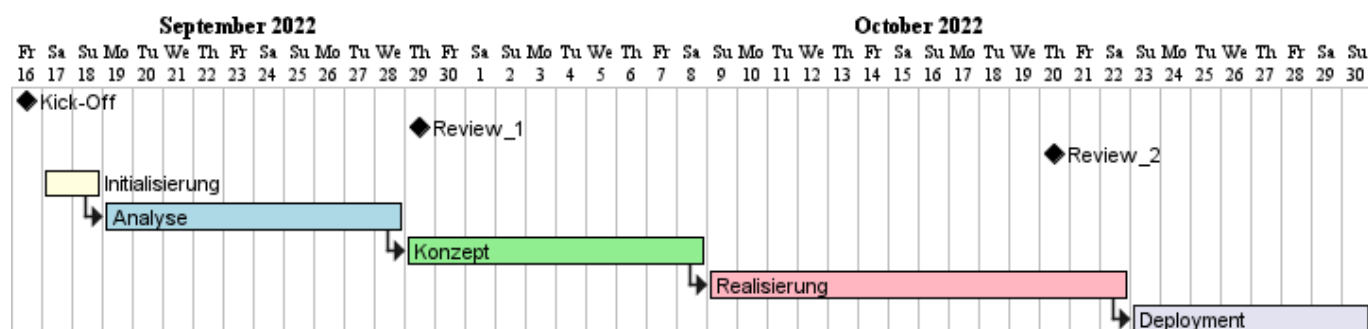


Abb. 1: GANTT Grobplanung

3.1.2 Initialisierung



Abb. 2: GANTT Initialisierung

Task	Beschreibung	Geplant	Ist	Delta
Dokumente Vorbereiten	LaTeX-Dokument aufsetzen, PlantUML Diagrammvorlagen erstellen	4h	3h	-1h
Repository Anlegen	GitHub Repos für Dokumentation und Projekt anlegen	1h	1h	0h
Themeneingabe	Themeneingabe in Dokumentation einfügen	1h	0.5h	-0.5h
Planung	Projektplanung Durchführen	6h	3.5h	-2.5
Total		12h	8h	-4h

Tabelle 1: Initialisierungs Tasks

3.1 Planung

3.1.3 Analyse

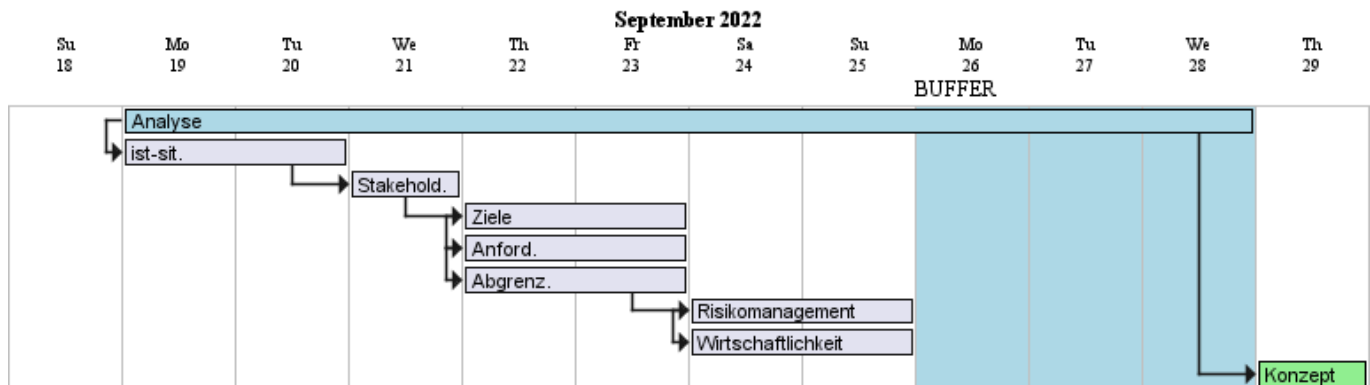


Abb. 3: GANTT Analyse

Task	Beschreibung	Geplant	Ist	Delta
Ist-Situation	Ausgangslage analysieren	4h	3h	-1h
Stakeholderanalyse	Stakeholder erfassen und gewichten	3h	2h	-1h
Ziele	Zielsetzungen aus Themeneingabe präzisieren	2h	3h	+1h
Anforderungen	Anforderungen aus Themeneingabe präzisieren	2h	1h	-1h
Abgrenzung	Abgrenzungen der Lösung definieren	3h	1h	-2h
Risikoanalyse	Risiken identifizieren und Massnahmen definieren	6h	2h	-4h
Wirtschaftlichkeitsanalyse	Wirtschaftlichkeit der Lösung eruieren	4h	1h	-3h
Total		24h	13h	-11h

Tabelle 2: Analyse Tasks

3.1.4 Konzept

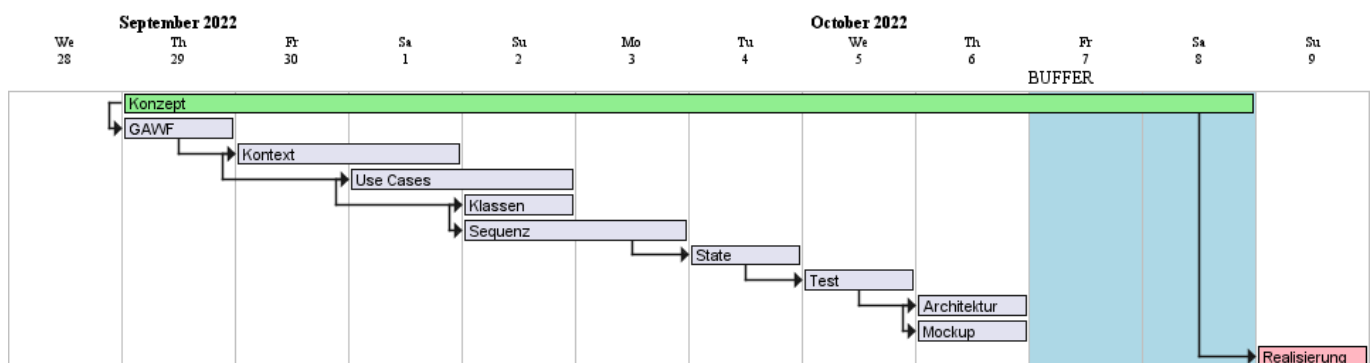


Abb. 4: GANTT Konzept

3.1 Planung

Task	Beschreibung	Geplant	Ist	Delta
GAWF	Geschäftsanwendungsfälle beschreiben	6h	0h	
Kontext	Kontextdiagramm erstellen	3h	0h	
Use Cases	Use-Case-Diagramme erstellen	5h	0h	
Klassen	Klassendiagramm erstellen	2h	0h	
Sequenz	Sequenzdiagramme erstellen	4h	0h	
State	Statediagramme erstellen	3h	0h	
Test	Testkonzept erstellen	4h	0h	
Architektur	Planung der Software Architektur	3h	0h	
Mockup	Mockup der Applikationen erstellen	5h	0h	
Total		35h	0h	0h

Tabelle 3: Konzept Tasks

3.1.5 Umsetzung

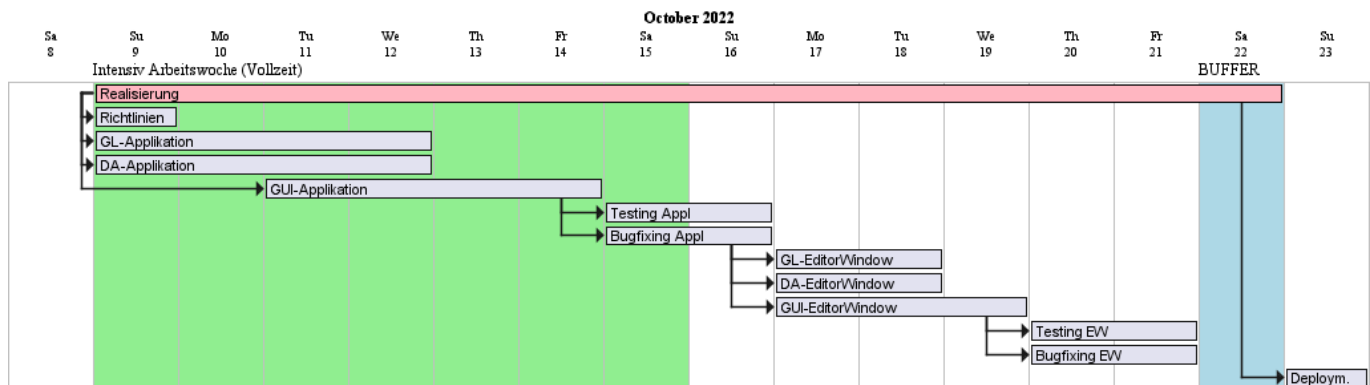


Abb. 5: GANTT Realisierung

Task	Beschreibung	Geplant	Ist	Delta
Richtlinien	Coding Richtlinien definieren	2h	0h	
GL-Applikation	Geschäftslogik der Applikation implementieren	11h	0h	
DA-Applikation	Data handling der Applikation implementieren	6h	0h	
GUI-Applikation	GUI der Applikation implementieren	25h	0h	
Testing Applikation	Testen der Applikation anhand Testkonzept	6h	0h	
Bugfixing Applikation	Allfällige Bugs fixen / Buffer	8h	0h	
GL-EditorWindow	Geschäftslogik des EditorWindows implementieren	4h	0h	
DA-EditorWindow	Data handling des EditorWindows implementieren	4h	0h	
GUI-EditorWindow	GUI des EditorWindows implementieren	8h	0h	
Testing EditorWindow	Testen des EditorWindows anhand Testkonzept	4h	0h	
Bugfixing EditorWindow	Allfällige Bugs fixen / Buffer	6h	0h	
Total		84h	0h	0h

Tabelle 4: Umsetzung Tasks

3.1 Planung

3.1.6 Deployment

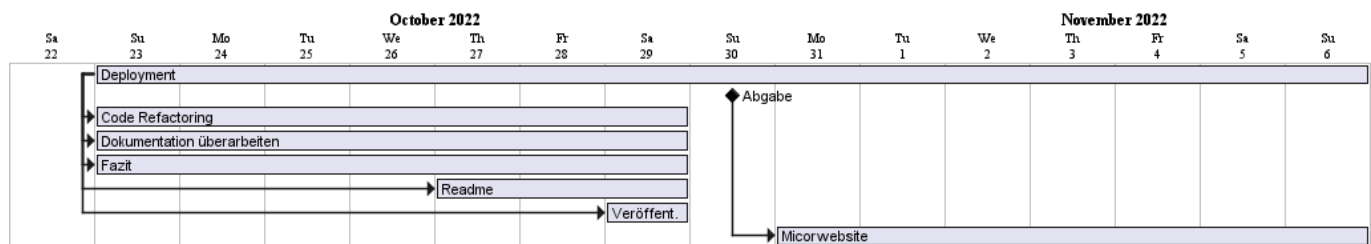


Abb. 6: GANTT Deployment

Task	Beschreibung	Geplant	Ist	Delta
Code Refactoring	Code cleanup / Optimisierungen	5h	0h	
Dokumentation Überarbeiten	Fehlerkorrektur, Format	5h	0h	
Readme	Readme für GitHub-Page erstellen	3h	0h	
Veröffentlichen	GitHub Projekt veröffentlichen	2h	0h	
Total		15h	0h	0h

Tabelle 5: Deployment Tasks

4 Analyse

In den folgenden Kapiteln, werden die punkte, welche bereits in der Themeneingabe definiert wurden, präzisiert.

4.1 Ausgangslage

Um als Indieentwickler Projekte zu planen, hat man heutzutage bereits eine relativ grosse Auswahl an Projektmanagement-Tools. Tools wie Azure DevOps sind jedoch hauptsächlich für grössere Teams gedacht und der Mehraufwand schreckt die meisten Einzelentwickler ab. Deshalb wird meist auf ein Projektmanagement-Tool verzichtet und man behilft sich mit Post-Its oder einem Whiteboard.

Viele Tools haben einen ähnlichen Aufbau, daher habe ich mich im Rahmen der Ist-Analyse dazu entschieden, bestehende Produkte etwas genauer anzusehen und Vor- und Nachteile im Bezug auf Indieentwickler rauszufiltern.

4.1.1 Analyse von bestehenden Lösungen



Abb. 7: TRELLO Logo

Trello ist ein Online-Tool zum verwalten von Aufgaben und gehört dem Unternehmen Atlassian. Das beliebte Online-Tool ist seit 2011 auf dem Markt und erfreut sich an milionen von Benutzern.

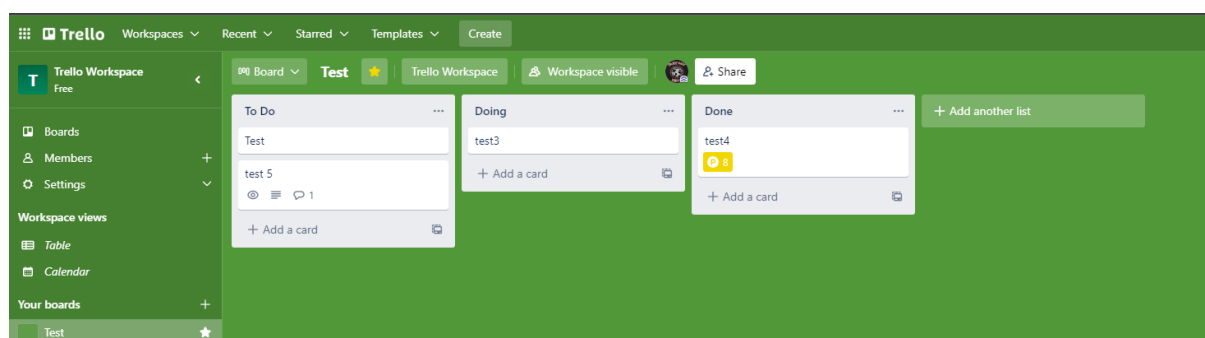


Abb. 8: TRELLO board

Das Layout entspricht einem klassischen Kanban-Board, welches jedoch nach belieben angepasst und modifiziert werden kann. Das Geschäftsmodell von Trello lässt sich am ehsten als "freemium" bezeichnen, die nötigsten Funktionen sind gratis, reicht dies nicht aus, so kann ein Premium-Abo für 10.- pro Monat abgeschlossen werden. Beim Testen des Dienstes ist mir das penetrante anbieten des 30-Days-Free-Trial-Angebotes, welches sich nach abschluss der 30 Tage automatisch verlängert besonders negativ aufgefallen.

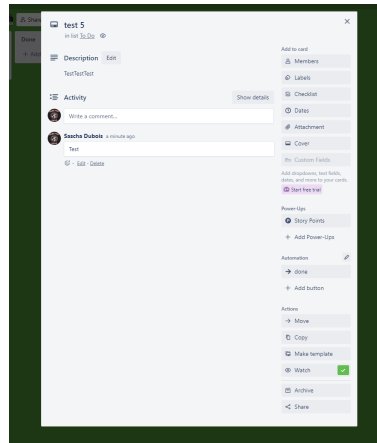


Abb. 9: TRELLO Card

Dem Trello-Board können sogenannte "Cards" hinzugefügt werden. Den "Cards" kann man standardmässig einen Titel, eine Beschreibung und Kommentare hinzufügen. Es lassen sich ausserdem Personen, Labels, Checklists, Covers und Anhänge anfügen.

Fängt man das erste Projekt mit Trello an, so scheinen die Funktionen relativ übersichtlich zu sein. Die grösste Stärke, meiner Meinung nach, liegt jedoch in der konfigurierbarkeit von Trello. Es können nämlich "PowerUps" eingefügt werden. PowerUps sind Plugins, mit denen die Funktionen von Trello beliebig erweitert werden können. Zum Beispiel gibt es standardmässig keine Möglichkeit den Cards irgendwelche StoryPoints oder Stunden zuzuweisen. Fügt man jedoch das entsprechende PowerUp hinzu so ist dies kein Problem mehr.

Und es gibt fast für alles ein entsprechendes PowerUp, so können auch Excel-Daten mittels PowerUp importiert werden. Dem Benutzer sind also nur wehnig Grenzen gesetzt, was die Personalisierung von Trello angeht.

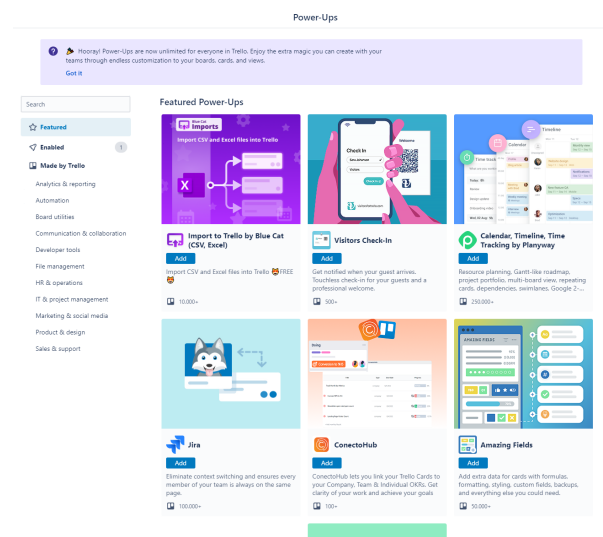


Abb. 10: TRELLO PowerUps

PowerUps können in einer Art App-Store durchsucht und hinzugefügt werden.

Da die PowerUps auch von Trello-Usern erstellt werden können, gibt es fast für jeden Use-Case ein entsprechendes PowerUp.

Besitzer	Atlassian
Gründung	2011
Plattform	Web
Layout	Kanban
Geschäftsmodell	Freemium

Tabelle 6: TRELLO Details

Bewertungspunkt	Bewertung	Begründung
Benutzerfreundlichkeit	****	Die Hohe konfigurierbarkeit führt zwangsläufig dazu, dass die Übersichtlichkeit der Funktionen etwas leidet
Darstellung	*****	Klassisches Kanbanboard, Backgrounds und Design können nach belieben angepasst werden
Usability	*****	Da Trello eine Webapplikation ist, ist sie auf allen webfähigen Geräten zugänglich
Funktionalität	*****	Durch die PowerUps kann die Funktionalität nach belieben erweitert werden
Preis	****	Die Gratisversion ist absolut ausreichend, jedoch wird man immer wieder dazu gedrängt ein Premium-Abo abzuschliessen welches nicht gerade günstig ist

Tabelle 7: TRELLO Bewertung

Punkte zur berücksichtigung in eigenem Projekt:

- Konfigurierbarkeit mittels anfügbarer Komponenten (PowerUps)
- Kanban Layout
- Design



Abb. 11: Monday.com Logo

Monday.com ist eine Online-Plattform zum erstellen von Anwendungen und Arbeitsverwaltungs Software. Die Plattform wurde 2014 veröffentlicht.

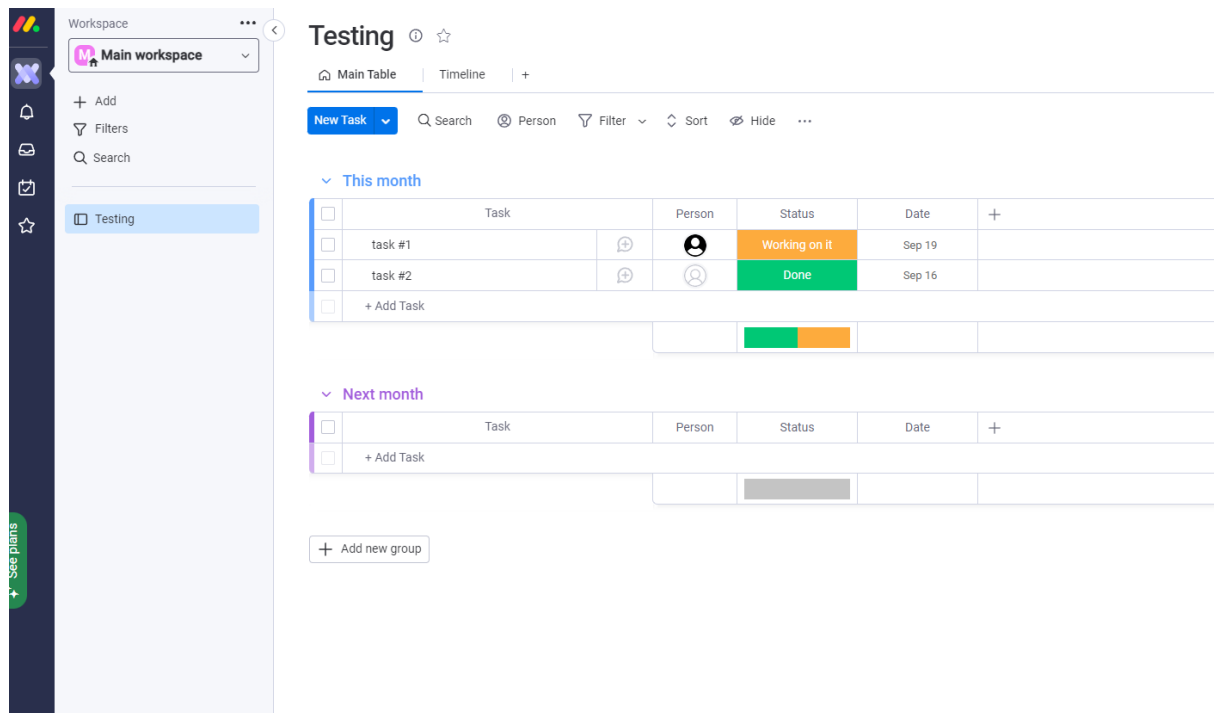


Abb. 12: TRELLO board

In der Standardkonfiguration werden Tasks aufgelistet und nicht als Kanbanboard dargestellt. Wird ein neuer Workspace erstellt, so klickt man sich erst durch eine Reihe fragen, deren Resultate anschliessend das Layout und die Funktionen des Workspaces vordefinieren.

Die Ansicht des Workspaces ist nach belieben konfigurierbar, so können Tasks nach Monaten gruppiert werden oder aber auch zB. nach Sprints oder Thema. Monday.com setzt ebenfalls auf das Freemium Geschäftsmodell, für ein Abo wird jedoch nicht so penetrant geworben wie bei anderen Anbietern.

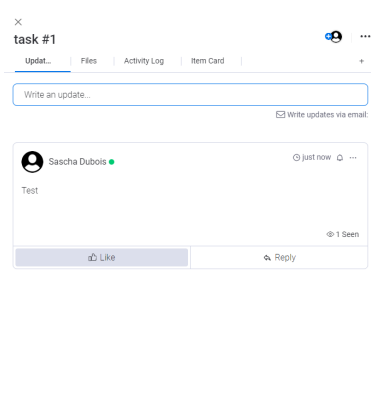


Abb. 13: Monday.com Task

4.1 Ausgangslage

Monday.com können Tasks erstellt werden, im gegensatz zu anderen Anbietern sind diese Tasks sehr minimalistisch gehalten und enthalten lediglich einen Titel und eine Beschreibung.

Den Tasks können jedoch weitere Views angefügt werden, in deren weiter felder und Funktionen zu verfügung stehen. Vom Prinzip her gleicht es zwar den PowerUps von trello, dort finde ich jedoch die Anbindung um einiges intuitiver und effizienter.

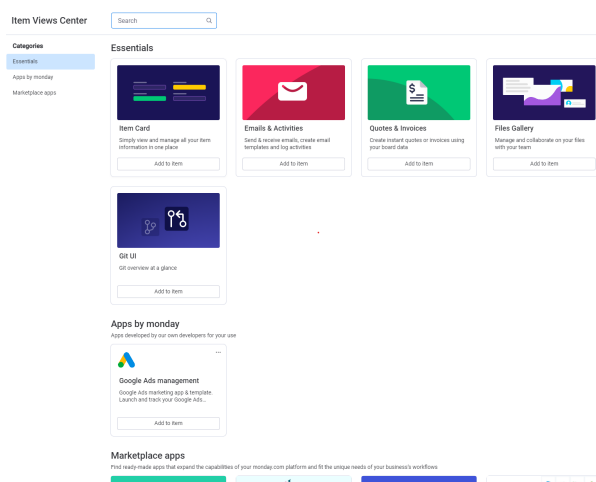


Abb. 14: Monday.com Views

Views können im Item-View-Center durchstöbert und hinzugefügt werden. Die hinzugefügten Views erscheinen dann in einem separaten Tab im Task. In den Views ist es ausserdem möglich Widgets einzufügen und das generelle Layout zu konfigurieren.

Im Item-View-Center findet man native Monday.com Views aber auch ausreichend andere Views von drittanbietern.

Besitzer	monday.com
Gründung	2012
Plattform	Web
Layout	Liste
Geschäftsmodell	Freemium

Tabelle 8: Monday.com Details

Bewertungspunkt	Bewertung	Begründung
Benutzerfreundlichkeit	***	—
Darstellung	****	Listenansichten werden schnell unübersichtlich
Usability	*****	Da monday.com eine Webapplikation ist, ist sie auf allen webfähigen Geräten zugänglich + gut optimiert für Mobilgeräte
Funktionalität	****	Durch Views können Funktionalitäten hinzugefügt werden jedoch ist die auswahl deutlich kleiner als z.B. bei Trello
Preis	*****	Gratis version reicht vollkommen

Tabelle 9: Monday.com Bewertung

Punkte zur berücksichtigung in eigenem Projekt:

- Lieber keine Listenansicht verwenden

4.2 Stakeholder

Im folgenden abschnitt werden die wichtigsten Stakeholder ermittelt. Dabei werden deren Interessen an das Projekt analysiert und daraus eine entsprechende Gewichtung vorgenommen. Die erkenntnisse aus der Stakeholderanalyse fließen anschliessend in die Definition der Ziele und Anforderungen ein.

Folgende Stakeholder werden berücksichtigt:

- TEKO
- Indieentwickler
- Konsumenten von Indiespielen
- Ich
- GitHub
- Unity

Auf der nächsten Seite werden die identifizierten Stakeholder nach Ihrer jeweiligen Macht und ihrem Einfluss gewichtet.

Dabei zeigt sich folgendes Bild:

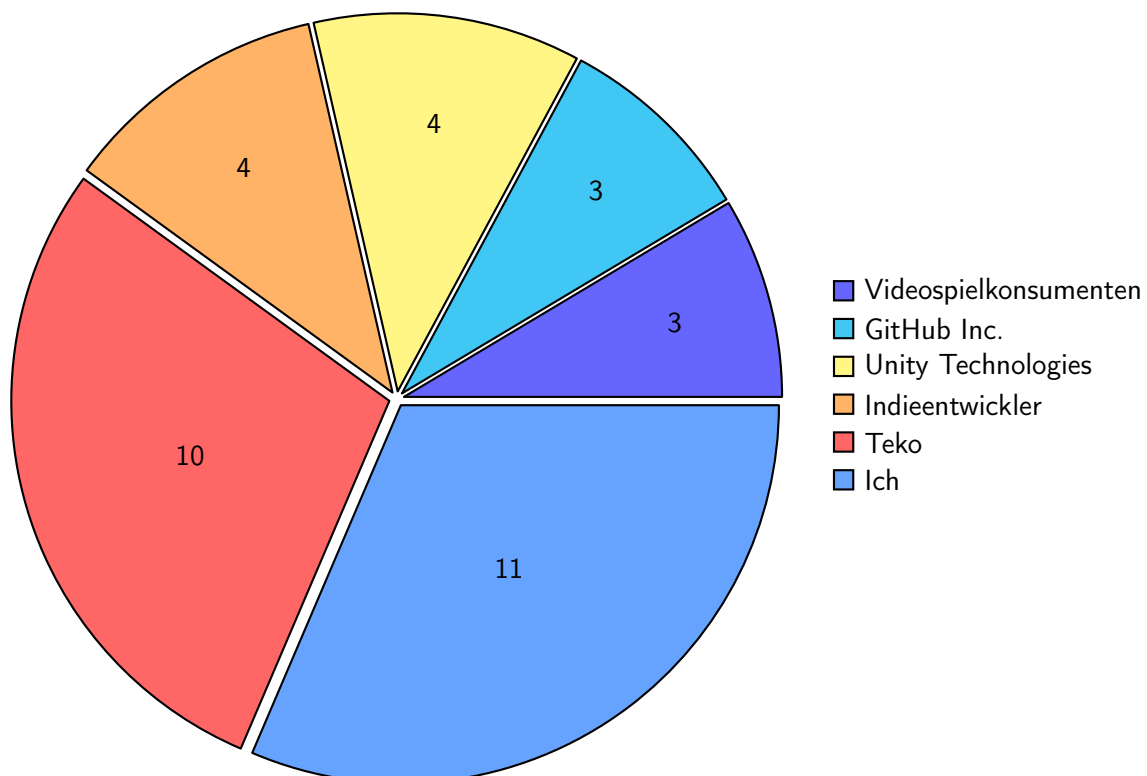


Abb. 15: Stakeholder Anteile

4.2 Stakeholder

4.2.1 Stakeholdergewichtung

	Stakeholder	Direkt				Indirekt			
		Ich	Indienteilnehmer	Teko	Konsumenten von Videospielen	Unity Technologies	GitHub Inc. (Microsoft)		
Allgemeines	Beschreibung	Projektleiter / Entwickler und Indienteilnehmer ->	Unabhängige Videospieler Entwickler (Einzelpersonen / Kleine Teams)	Abnehmer Diplomarbeit	Konsumenten von Videospielen	Anbieter der Unity3D Game Engine	Anbieter des gleichnamigen Online-Dienstes für die Versionsverwaltung von Software Produkten		
	Organisatorische Schlagkraft	Bestimmt Art und Umfang des Projektes innerhalb der vorgegebenen Richtwerte	Können vor Veröffentlichung nur indirekt Einfluss nehmen	Kontrolliert Art und Umfang des Projektes und kann direkten Einfluss darauf nehmen	Kann keinen direkten Einfluss nehmen	Kann keinen direkten Einfluss nehmen. Durch Änderungen von Unity kann jedoch indirekt Einfluss genommen werden	Kann Einfluss durch bestmögliche Bestimmung der Richtlinien zur Verwaltung und Veröffentlichung des Projektes nehmen		
	Personelle Schlagkraft	1 MA	N	1	N	N	N		
	Fachliche Kompetenz (0-10)	8	8	7	2	2	3		
	Rechtliche Einflusspotentiale	Kennt die Rechtlichen Voraussetzungen		Kennt die Rechtlichen Voraussetzungen		Kann rechtliche Schritte bei Verstoß der eigenen Vorgaben einleiten	Kann rechtliche Schritte bei Verstoß der eigenen Vorgaben einleiten		
Erwartungen	Erfahrung mit ähnl. Projekten Intern / Extern	Ja	Ja	Ja	Nein	Ja	Ja		
	vom Projekt	Siehe "Indienteilnehmer" ->	Qualitätssteigerung der Projekte durch besseres und einfacheres Planen	Microwebsite, Dokumentation und Präsentation	Höhere Qualität von Indiespielen				
	an das Projekt	Einfache Wartung, Saubere Dokumentation, Einfache Erweiterbarkeit	Gute Dokumentation, Erweiterbarkeit	Gute Dokumentation, Gute Planung und Umsetzung derer					
	Faktor	4	1	4	1	1	1		
	Faktor	4	2	3	1	2	1		
Erwartungen und Konflikte	Faktor	3	1	3	1	1	1		
	Bemerkung	Es wird hohe Qualität angestrebt, jedoch muss aufgrund des Zeitfensters auf unnötige Features verzichtet werden können	Da das Produkt im Vorfeld nicht öffentlich vermarktet wird, sind keine hohen Erwartungshaltungen zu erwarten						
	Gewichtung (max. 15)	11	4	10	3	4	3		

Abb. 16: Stakeholder Analyse

4.3 Ziele

Nachfolgend definiere ich die Ziele meiner Diplomarbeit, da das Projekt keinen direkten Wirtschaftlichen auslöser hat, beziehen sich die Ziele mehr auf

4.3.1 Projektmanagement Ziele

Mein Ziel ist es, ...

- ... die Deadline bezüglich der Diplomarbeit einzuhalten, dazu werde ich anhand der Planung ein Controlling durchführen um etwaige Komplikationen oder Verzögerungen frühzeitig erkennen und entsprechende Massnahmen einleiten zu können.
- ... zwei Reviews vor Projektende durchzuführen

4.3.2 Projekt spezifische Ziele

Mein Ziel ist es, ...

- ... die Lösung nach Abschluss der Diplomarbeit für alle Entwickler zugänglich zu machen, dazu wird die Lösung als Open-Source Projekt entwickelt.
- ... dass die Lösung, nach Abschluss der Diplomarbeit, für alle zugänglich und verständlich dokumentiert wird. Dazu werde ich eine Readme-Datei via Git-Hub veröffentlichen.

4.3.3 Long-Term Ziele

Mein Ziel ist es, ...

- ... mit Hilfe, der in diesem Projekt erstellten Lösung, ein Videospiel-Projekt bis Ende 2023 abschliessen zu können

4.4 Anforderungen

4.4.1 Applikation

- Bestehende Unity-Projekte können der Applikation hinzugefügt werden
- Aktivitäten können in der Applikation erstellt, bearbeitet und gelöscht werden
- Aktivitäten können terminiert werden (Start -und Enddatum)
- Aktivitäten können kommentiert werden
- Aktivitäten können priorisiert werden
- Aktivitäten können punkte zugewiesen werden (z.B. Story Points oder Stunden)
- Aktivitäten können gruppiert werden (z.B. zu Phasen)
- Gruppen von Aktivitäten können kommentiert werden*
- Gruppen können terminiert werden (Start- und Enddatum)
- Gruppen können priorisiert werden
- Gruppen können punkte zugewiesen werden (z.B. Story Points oder Stunden)
- Meilensteine können in der Applikation geplant werden
- Aktivitäten können Benutzern zugewiesen werden
- Wird ein Unity-Projekt der Applikation hinzugefügt, so wird das entsprechende Custom-Editor-Script dem Projektordner hinzugefügt

4.4.2 In Unity integriertes Benutzerinterface

- Der User kann eigene Aktivitäten zum aktuell geöffneten Projekt erstellen
- Der User kann seine Aktivitäten zum aktuell geöffneten Projekt bearbeiten
- Projektinformationen zum aktuell geöffneten Projekt (Beschreibung, Meilensteine usw.) können angesehen werden

4.4.3 Nicht funktionale Anforderungen

- Projektdaten werden beim Projekt selbst hinterlegt
- Features sollen als Addon hinzugefügt werden können (Analog den PowerUps von Trello)
- Projektdaten können mittels versionsverwaltung (z.B. GIT) synchronisiert werden

4.5 Abgrenzung

- Projektdaten (Aktivitäten, Gruppen usw.) werden nicht auf einer Datenbank persistiert
- Die Softwarelösung ist ausschliesslich auf Windows Geräten verfügbar
- Synchronisation der Daten z.B. über einen Webserver ist nicht möglich

4.6 Risikomanagement

4.6.1 Risikoerkennung

Risiko-Art	Mögliche Risiken
Technologisch	<ul style="list-style-type: none"> Das Speichern der Projektdaten im Projektordner ist zu wehnig performant
Personell	<ul style="list-style-type: none"> Krankheitsausfall
Werkzeuge	<ul style="list-style-type: none"> Änderungen durch Unity in Form von Updates, könnten den Einsatz von Custom Editor Windows erschweren oder verunmöglichen
Anforderung	<ul style="list-style-type: none"> Nachträgliche Änderungen / Erweiterungen werden verlangt
Schätzung	<ul style="list-style-type: none"> Die Entwicklungsdauer wurde unterschätzt Die Dauer der Dokumentationen wurde unterschätzt

Tabelle 10: Risikoerkennung

4.6.2 Risikogewichtung

Nachfolgend werden die erkannten Risiken nach ihrer Eintrittswahrscheinlichkeit (0-5) und den möglichen Auswirkungen (0-5) bewertet.

Risiko	Eintrittswarscheinlichkeit	Auswirkung
Das Speichern der Projektdaten im Projektordner ist zu wehnig performant	3	2
Krankheitsausfall	3	2
Änderungen durch Unity in Form von Updates, könnten den Einsatz von Custom Editor Windows erschweren oder verunmöglichen	1	4
Nachträgliche Änderungen / Erweiterungen werden verlangt	2	3
Die Entwicklungsdauer wurde unterschätzt	3	3
Die Dauer der Dokumentationen wurde unterschätzt	3	3

Tabelle 11: Risikogewichtung

4.6.3 Massnahmen

Risiko	Massnahme
Das Speichern der Projektdaten im Projektordner ist zu wenig performant	Im Rahmen der Diplomarbeit werden keine direkten Massnahmen geplant. Performance Optimierungen, respektive Änderungen die daraus folgen werden nach Veröffentlichung der Applikation vorgenommen
Krankheitsausfall	Bei Eintritt eines krankheitsbedingten Ausfalls wird der Diplomlehrer sofort informiert und abgeklärt ob Anforderungen gekürzt werden können
Änderungen durch Unity in Form von Updates, könnten den Einsatz von Custom Editor Windows erschweren oder verunmöglichen	Eintrittswahrscheinlichkeit während der Projektdauer sehr klein, bei Eintritt wird mit der alten Unity-Version weitergearbeitet
Nachträgliche Änderungen / Erweiterungen werden verlangt	Werden zusätzliche Anforderungen verlangt, so muss der Mehraufwand berechnet werden und deren Folgen auf den Zeitplan.
Die Entwicklungsdauer wurde unterschätzt	Mit dem Diplomlehrer direkt über mögliche Kürzungen der Anforderungen diskutieren
Die Dauer der Dokumentationen wurde unterschätzt	Mit dem Diplomlehrer direkt über mögliche Kürzungen der Anforderungen diskutieren

Tabelle 12: Massnahmen

4.7 Wirtschaftlichkeit

Dadurch, dass die Applikation als Open-Source-Software zu Verfügung gestellt wird, ist eine direkte Überprüfung der Wirtschaftlichkeit kaum möglich.

Für mich als Entwickler wird die Software nicht Wirtschaftlich sein, denn die theoretischen Entwicklungskosten belaufen sich auf rund 20'000 Fr.- (ca. 200h Arbeitsaufwand * 100.- Stundensatz).

Die Effektiven Entwicklungskosten sind jedoch eher bei 0 Fr.- anzusiedeln, denn die einzigen Kosten die ich ausser meiner Zeit investiere sind Heiz, Strom und Internetkosten, welche jedoch so oder so anfallen.

Dies bedeutet aber nicht, dass Open-Source-Software nicht Wirtschaftlich ist. Aus Entwickler und Firmensicht stimmt das zwar, jedoch können die Nutzer (Mich eingeschlossen) der Software einen erheblichen Wirtschaftlichen Nutzen erhalten. Denn die Benutzung der Software ist gratis, es fallen keine zusätzlichen Kosten an und zudem können Entwicklungsdauer und Kosten bei der Entwicklung von Videospielen verkleinert werden.

5 Konzept

5.1 Use Case Diagramme

Im folgenden Abschnitt werde ich die Use-Cases identifizieren und beschreiben.

Dies dient zum einen zum Erfassen der benötigten Klassen und zum anderen um ein gutes Testkonzept zu erstellen.

5.1.1 Use Case Übersicht

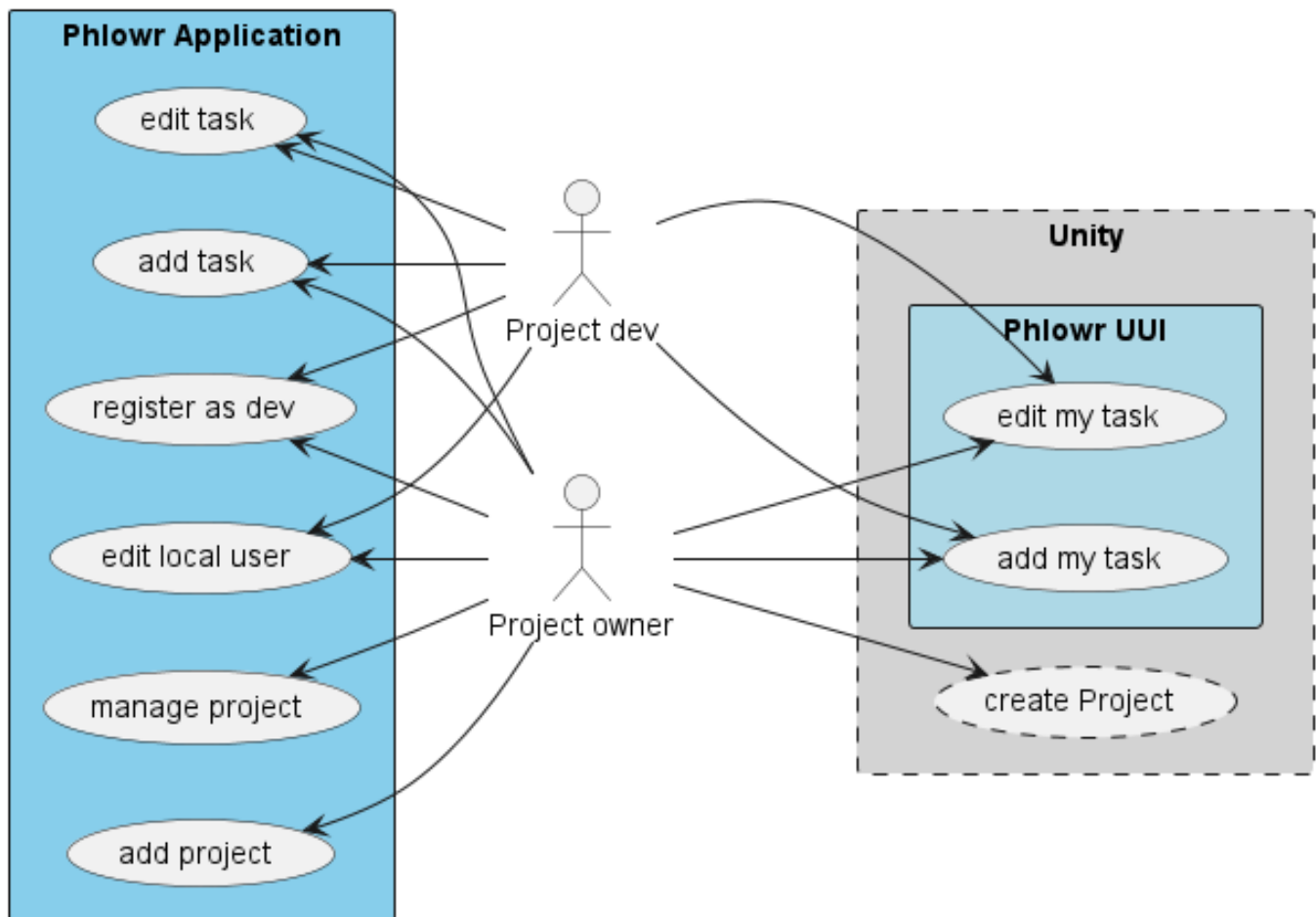


Abb. 17: Use Case Diagramm <Übersicht>

5.1.2 Projekt hinzufügen

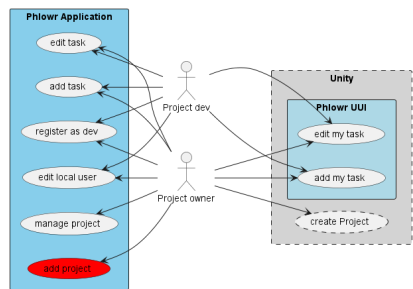


Abb. 18: Use Case Diagramm <add project>

Use Case	ADD PROJECT
Beschreibung	Ein Unity-Projekt wird der Applikation hinzugefügt
Includes	<ul style="list-style-type: none"> keine
Akteure	<ul style="list-style-type: none"> Project Owner
Auslöser	<ul style="list-style-type: none"> Ein bestehendes Unity-Projekt soll mit Phlowr verwaltet werden
Vorbedingungen	<ul style="list-style-type: none"> Das hinzuzufügende Unity-Projekt wurde bereits erstellt
Abschlussbedingungen	<ul style="list-style-type: none"> Das gewünschte Projekt ist in der Applikation erfasst
Ablauf	<ol style="list-style-type: none"> Applikation öffnen <Projekt Hinzufügen> wählen Den entsprechenden Projekt-Ordner auswählen Speichern
Zu Beachten / Notizen	<ul style="list-style-type: none"> Projekt sollte nur einmal hinzugefügt werden können Projekt Ordner-Struktur sollte überprüft werden

Tabelle 13: Use Case: ADD PROJECT

5.1.3 Projekt verwalten

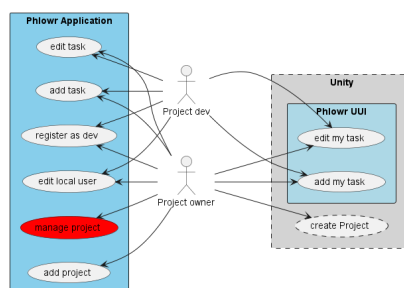


Abb. 19: Use Case Diagramm <manage project>

Use Case	MANAGE PROJECT
Beschreibung	Ein Projekt wird in der applikation verwaltet
Includes	<ul style="list-style-type: none"> • Beschreibung anpassen (edit description) • Name anpassen (edit name) • User-Slots anpassen (edit user-slots) • Projekt-Owner ändern (change PO)
Akteure	Projekt-Owner
Auslöser	Ein Projekt soll angepasst werden
Vorbedingungen	Das Projekt wurde der Applikation hinzugefügt und der ausführende User ist der Project-Owner
Abschlussbedingungen	Die gewünschten Änderungen wurden gemacht und das Projekt wurde gespeichert
Ablauf	<ol style="list-style-type: none"> 1. Applikation öffnen 2. <Projekt Editieren> wählen 3. Gewünschte Änderungen vornehmen (siehe Includes) 4. Speichern
Zu Beachten / Notizen	<ul style="list-style-type: none"> • Projekte dürfen nur vom PO editiert werden können

Tabelle 14: Use Case: MANAGE PROJECT

5.1.4 Includes von „Projekt verwalten“

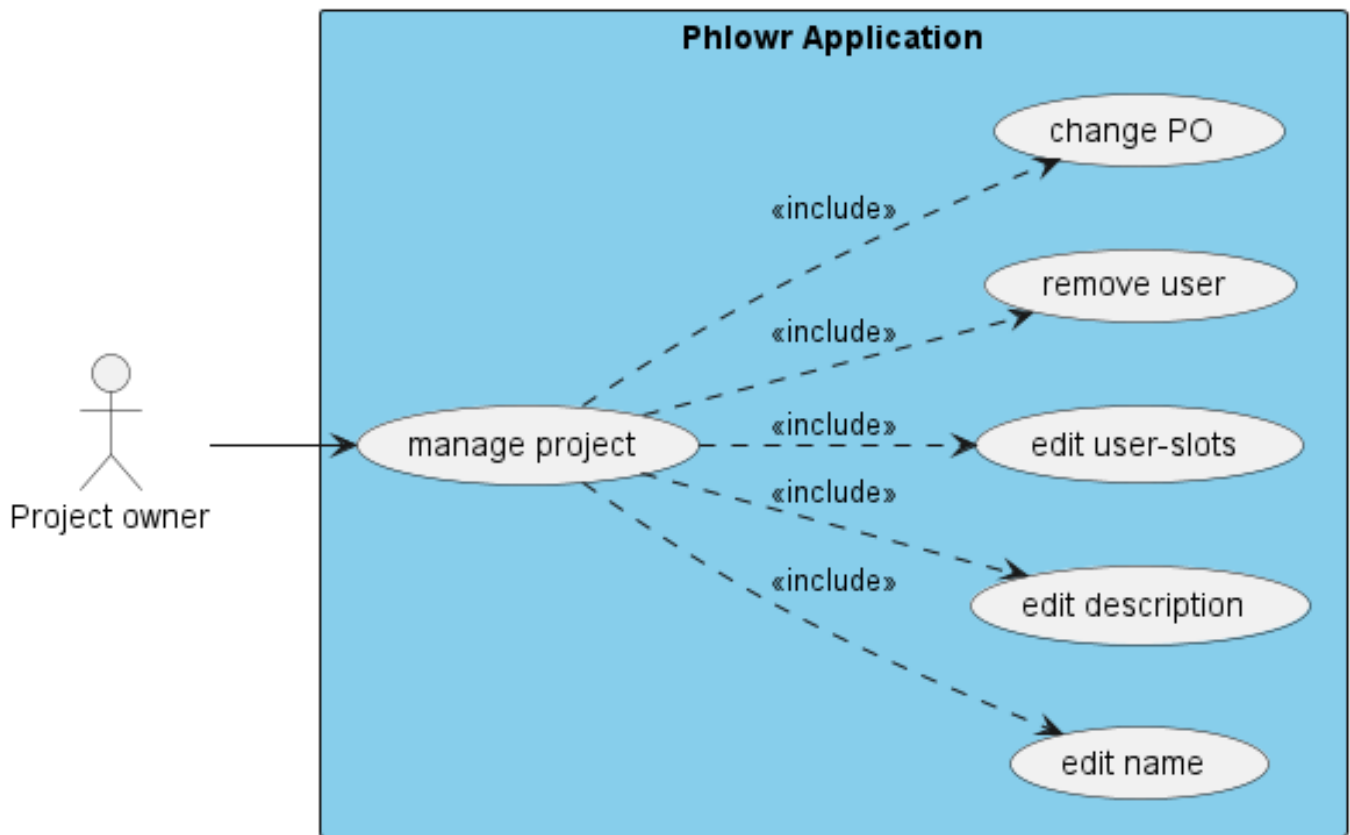


Abb. 20: Use Case Diagramm <manage project>

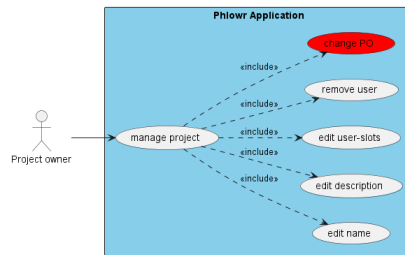


Abb. 21: Use Case Diagramm <change PO>

Use Case	CHANGE PO
Beschreibung	Der Projekt-Owner des Projektes wird geändert
Includes	<ul style="list-style-type: none"> keine
Akteure	Projekt-Owner
Auslöser	Der Projekt-Owner soll geändert werden
Vorbedingungen	<ul style="list-style-type: none"> Vorbedingungen vom <MANAGE PROJECT>-Use-Case Der neue Projekt-Owner muss als User im Projekt registriert sein
Abschlussbedingungen	Der Projekt-Owner wurde geändert und die Änderungen sind gespeichert
Ablauf	<ol style="list-style-type: none"> 1. Applikation öffnen 2. <Projekt Editieren> wählen 3. <Change Project Owner> wählen 4. Neuer PO aus liste der registrierten User auswählen 5. Speichern
Zu Beachten / Notizen	<ul style="list-style-type: none"> Wird ein PO inaktiv, so kann das Projekt nicht mehr bearbeitet werden, der PO muss dann manuell im JSON-File geändert werden

Tabelle 15: Use Case: manage project -> CHANGE PO

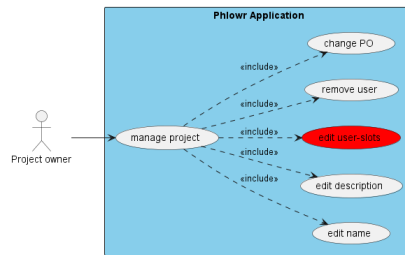


Abb. 22: Use Case Diagramm <edit user-slots>

Use Case	EDIT USER-SLOTS
Beschreibung	User-Slots des Projektes werden editiert
Includes	<ul style="list-style-type: none"> keine
Akteure	Projekt-Owner
Auslöser	<ul style="list-style-type: none"> ein Slot für einen neuen Entwickler wird benötigt ein Slot für einen neuen Entwickler wird nicht mehr benötigt
Vorbedingungen	<ul style="list-style-type: none"> Vorbedingungen vom <MANAGE PROJECT>-Use-Case
Abschlussbedingungen	Die Anzahl User-Slots wurde im Projekt angepasst und entsprechend gespeichert
Ablauf	<ol style="list-style-type: none"> 1. Applikation öffnen 2. <Projekt Editieren> wählen 3. + / - unter dem abschnitt <User-Slots> wählen 4. Speichern
Zu Beachten / Notizen	<ul style="list-style-type: none"> keine

Tabelle 16: Use Case: manage project -> EDIT USER-SLOTS

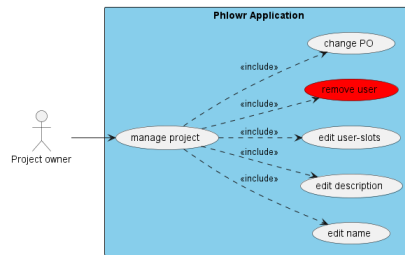


Abb. 23: Use Case Diagramm <remove user>

Use Case	REMOVE USER
Beschreibung	Ein User wird aus einem Projekt entfernt
Includes	<ul style="list-style-type: none"> keine
Akteure	Projekt-Owner
Auslöser	<ul style="list-style-type: none"> ein Entwickler arbeitet nicht mehr am Projekt ein Entwickler hat fälschlicherweise beim Projekt registriert
Vorbedingungen	<ul style="list-style-type: none"> Vorbedingungen vom <manage project>-Use-Case Der zu entfernende User hat sich als Entwickler beim Projekt registriert Die vorhandenen Tasks des Users wurden vorgängig bereinigt
Abschlussbedingungen	Der User ist nicht mehr im Projekt registriert
Ablauf	<ol style="list-style-type: none"> 1. Applikation öffnen 2. <Projekt Editieren> wählen 3. unter dem Abschnitt <Entwickler> wird der entsprechende User geöffnet 4. <Entwickler entfernen> wird gewählt 5. Speichern
Zu Beachten / Notizen	<ul style="list-style-type: none"> wird ein User entfernt, so müssen allfällige Kommentare des Users in den Tasks usw. entsprechend berücksichtigt werden

Tabelle 17: Use Case: manage project -> remove user

5.1.5 Als Entwickler registrieren

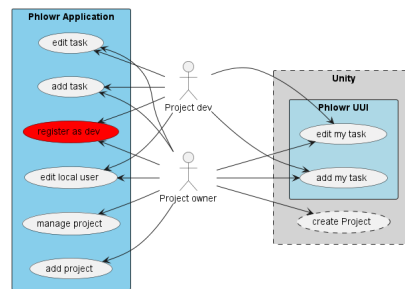


Abb. 24: Use Case Diagramm <register as dev>

Use Case	REGISTER AS DEV
Beschreibung	Ein User registriert sich bei einem Projekt. Dies ist nötig, damit der User zur Planung im Projekt zu Verfügung steht.
Includes	<ul style="list-style-type: none"> keine
Akteure	Entwickler
Auslöser	<ul style="list-style-type: none"> ein Entwickler möchte an einem Projekt mitarbeiten
Vorbedingungen	<ul style="list-style-type: none"> es ist ein freier Slot im Projekt verfügbar der Entwickler ist im Projekt noch nicht registriert das Projekt ist auf dem aktuellsten Stand
Abschlussbedingungen	Der User ist im Projekt zur plan verfügbar Der User kann für sich selbst Tasks im Projekt erstellen
Ablauf	<ol style="list-style-type: none"> 1. Applikation öffnen 2. Projekt öffnen 3. <Als Entwickler Registrieren> klicken 4. Speichern 5. falls nötig Projekt synchronisieren
Zu Beachten / Notizen	<ul style="list-style-type: none"> das Projekt muss auf dem neusten Stand sein -> es könnten Merge-Konflikte auftreten

Tabelle 18: Use Case: register as dev

5.1.6 Task hinzufügen (Applikation)

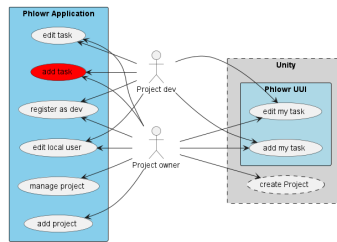


Abb. 25: Use Case Diagramm <add Task>

Use Case	ADD TASK
Beschreibung	Ein Task wird dem Projekt hinzugefügt
Includes	<ul style="list-style-type: none"> keine
Akteure	<ul style="list-style-type: none"> Entwickler Projekt-Owner
Auslöser	<ul style="list-style-type: none"> Ein Task soll dem Projekt hinzugefügt werden
Vorbedingungen	<ul style="list-style-type: none"> Das Projekt ist der Applikation hinzugefügt Der Ersteller ist Projekt-Owner oder beim Projekt als Entwickler registriert das Projekt ist auf dem aktuellsten Stand
Abschlussbedingungen	Ein neuer Task ist im Projekt verfügbar
Ablauf	<ol style="list-style-type: none"> 1. Applikation öffnen 2. Projekt öffnen 3. <Task hinzufügen> klicken 4. Details des neuen Tasks ausfüllen (Name, Beschreibung usw.) 5. Speichern 6. Projekt synchronisieren
Zu Beachten / Notizen	<ul style="list-style-type: none"> Tasks können ebenfalls bei bereits existierenden Tasks hinzugefügt werden, dazu wird einfach der entsprechende Task geöffnet.

Tabelle 19: Use Case: add task

5.2 Klassendiagramm

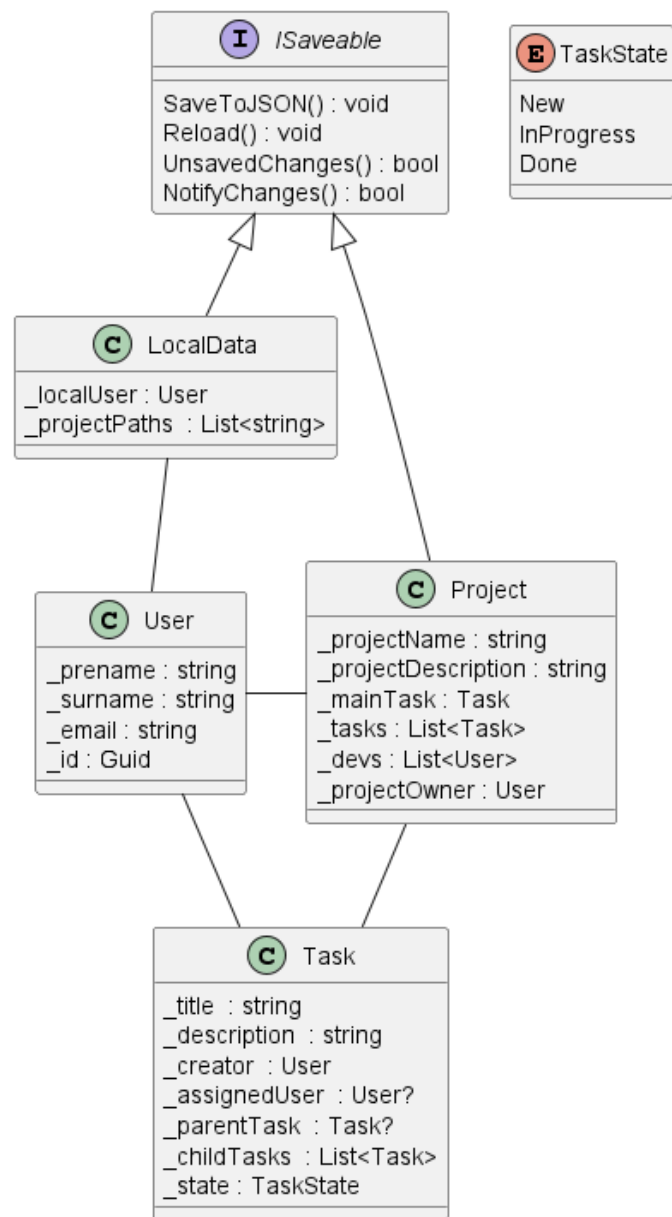


Abb. 26: Klassendiagramm

5.2.1 Projekt Klassenbeschreibung

Klasse	Projekt
Attribute	

Tabelle 20: Klassenbeschreibung: Projekt

5.3 Sequenzdiagramme

5.3.1 Laden der lokalen Daten beim Starten der Applikation

Da alle lokal zu persistierenden Daten in einem JSON-Dokument abgespeichert werden, müssen diese Daten beim Starten der Applikation geladen werden. Dabei gibt es zwei Szenarien:

1. Die Applikation wird zum ersten Mal gestartet und es existiert noch kein JSON-Dokument
2. Es existieren bereits gespeicherte Daten in Form eines JSON-Dokuments

In der Klasse „LocalData“ wird die statische Methode „LoadOrCreate“ ausgeführt, in dieser Methode wird über den Data-Access-Layer (DAL) geprüft, ob ein solches JSON Dokument existiert. Existiert dieses, so wird das Objekt über den DAL ausgelesen, dabei enthält das Objekt ebenfalls eine Instanz der Klasse „LocalUser“ welcher somit ebenfalls geladen wurde.

Existiert kein JSON-Dokument so wird erst ein neuer User als „LocalUser“ erstellt, dabei wird auch dessen Guid generiert (Das Generieren der Guid eines Users darf nur hier erfolgen!). Anschliessend wird eine neue Instanz der Klasse „LocalData“ mit dem neuen „LocalUser“ erstellt. Diese wird über den DAL als JSON-Dokument abgespeichert und kann beim nächstenmaligen Starten der Applikation geladen werden.

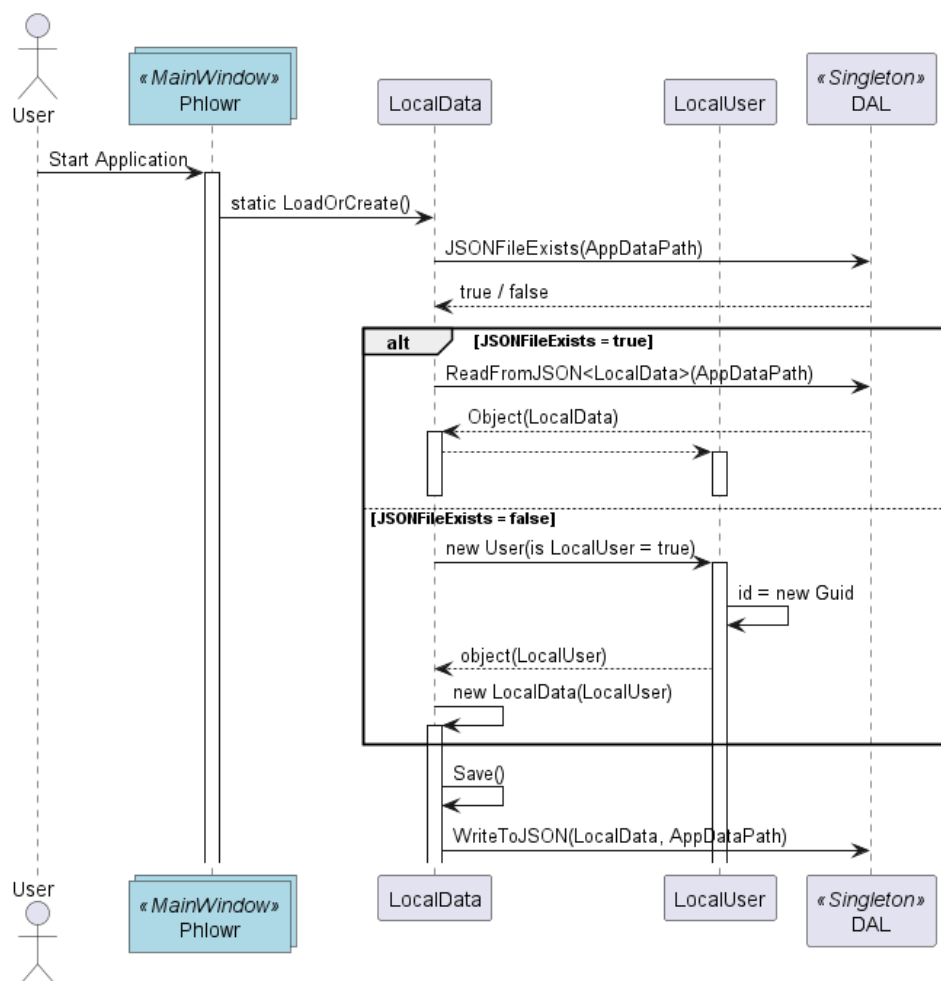


Abb. 27: Sequenzdiagramm: Load LocalData

5.3.2 Projekt Hinzufügen

Das Hinzufügen eines Projektes ist sehr einfach gehalten, denn bei diesem Vorgang werden keine Projekt-Objekte erzeugt oder geladen. Durch einen Klick auf „Projekt hinzufügen“ wird der Verzeichnis-Explorer angezeigt, in diesem navigiert der Benutzer zum gewünschten Unity-Projektordner und wählt diesen aus. Anschliessend wird die Methode „AddProjectPath“ in der LocalData-Instanz ausgeführt, ist der Pfad nicht bereits vorhanden, so wird dieser in die Liste aufgenommen.

Zum Schluss wird die LocalData-Instanz wieder abgespeichert.

Das Erzeugen der Projekte wird im Sequenzdiagramm „Laden der Projekte“ behandelt.

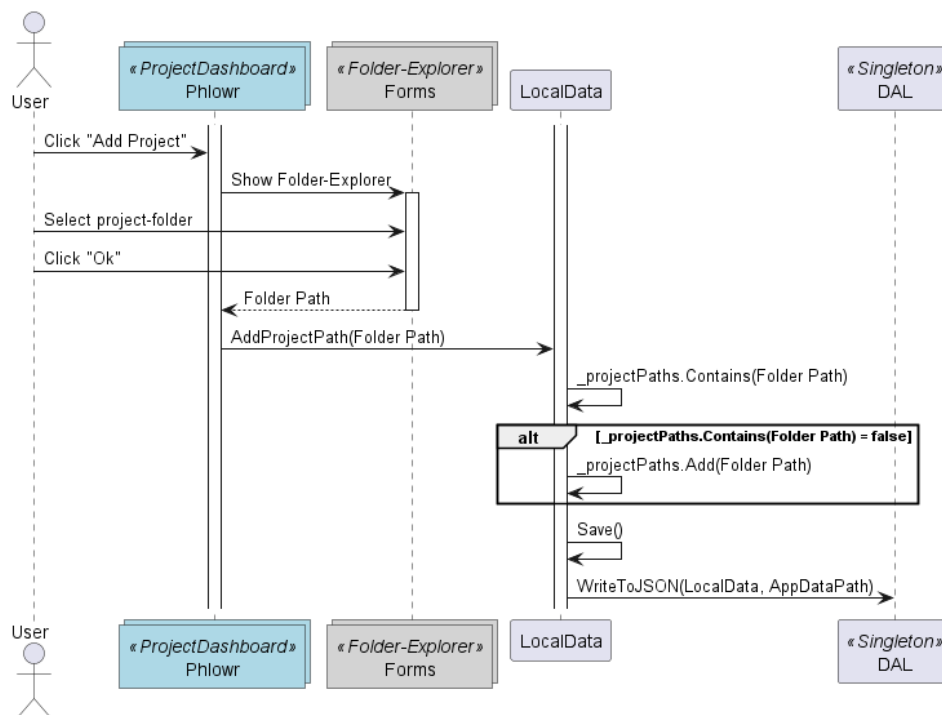


Abb. 28: Sequenzdiagramm: Add Project

5.3.3 Laden der Projekte

Beim Öffnen des Projekt-Dashboards werden alle hinzugefügten Projekte geladen. Dafür wird durch alle Projekt-Pfade, welche in der LocalData-Instanz vorhanden sind, geloopt. Für jeden Pfad wird die statische Methode „LoadOrCreate“ in der Projekt-Klasse ausgeführt. Diese Methode überprüft über den DAL, ob bereits ein JSON-Dokument unter dem angegebenen Pfad vorhanden ist. Ist ein JSON-Dokument vorhanden, so wird dieses über den DAL geladen und zurückgegeben. Ist noch keine Projektdatei vorhanden, so wird eine neue Projekt-Instanz erzeugt, der Projekt-Owner der neu erzeugten Instanz wird automatisch der LocalUser. Die Instanz wird anschliessend abgespeichert, als JSON-Dokument im angegebenen Pfad abgelegt und zurückgegeben. Das Projekt-Dashboard fügt die erhaltene Projekt-Instanz zur Projekt-Liste hinzu.

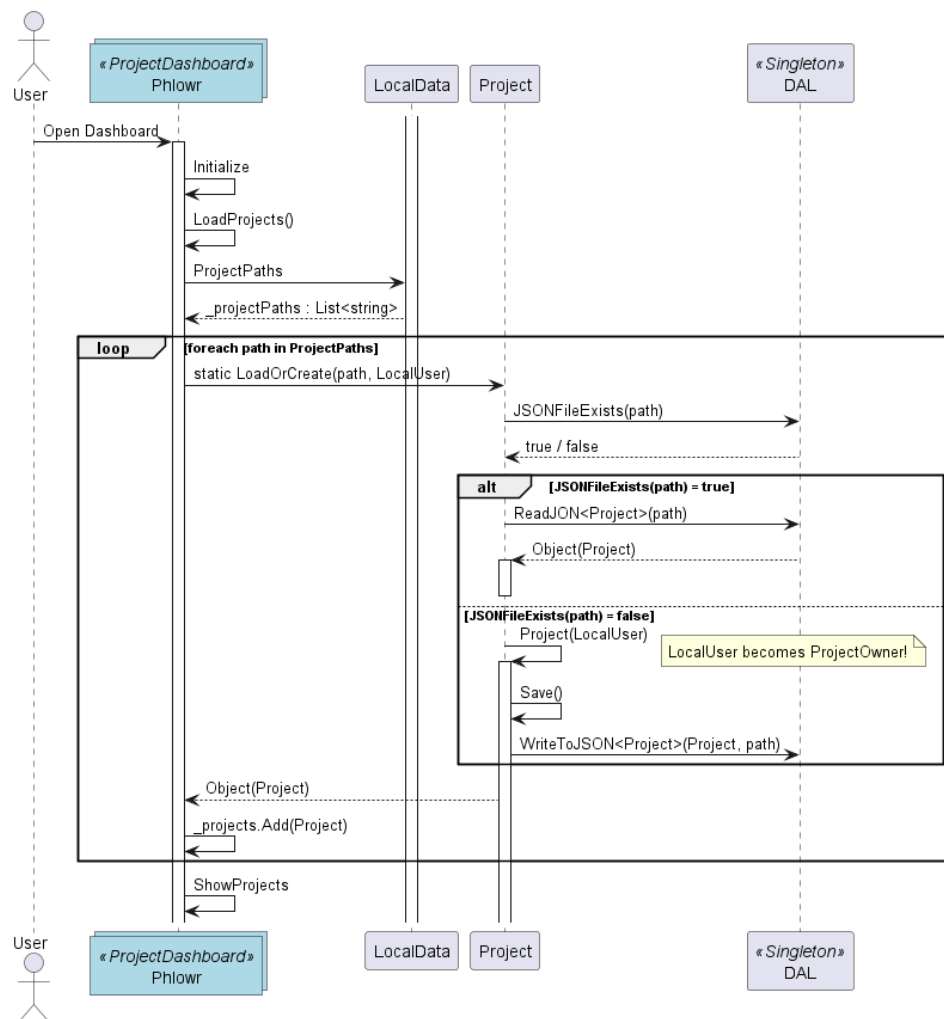


Abb. 29: Sequenzdiagramm: Load Projects

5.3.4 Als Entwickler registrieren

Damit ein User als Entwickler in einem Projekt geplant werden kann, muss er sich zuerst beim Projekt registrieren. Dafür navigiert er zum entsprechenden Projekt und wählt „Als Entwickler registrieren“ aus. Dadurch wird auf der entsprechenden Projekt-Instanz die Methode „RegisterUser“ mit dem LocalUser als Übergabeparameter ausgeführt. In dieser Methode wird überprüft, ob noch ein freier Slot verfügbar ist und ob sich der User nicht schon registriert hat. Für letzteres wird über alle registrierten User zusätzlich nach der Guid des neuen Users gesucht.

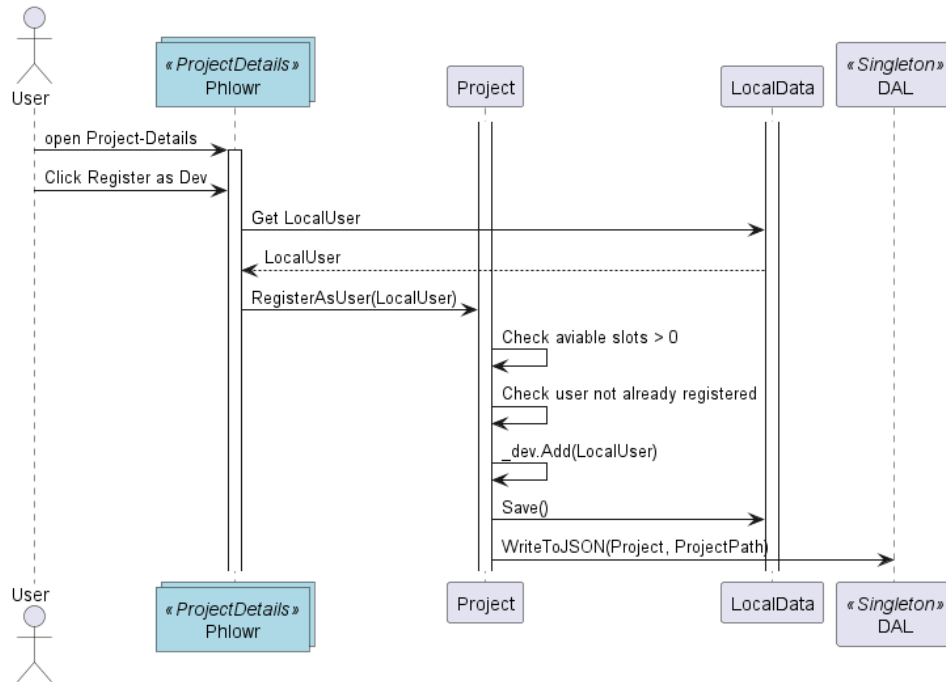


Abb. 30: Sequenzdiagramm: Register as user

5.4 State Diagramme

5.5 Systemarchitektur

5.6 Mockup

6 Realisierung

6.1 Entwicklungsumgebung

6.2 Codeconventions

Abbildungsverzeichnis

1	GANTT Grobplanung	5
2	GANTT Initialisierung	5
3	GANTT Analyse	6
4	GANTT Konzept	6
5	GANTT Realisierung	7
6	GANTT Deployment	8
7	TRELLO Logo	9
8	TRELLO board	9
9	TRELLO Card	10
10	TRELLO PowerUps	10
11	Monday.com Logo	12
12	TRELLO board	12
13	Monday.com Task	12
14	Monday.com Views	13
15	Stakeholder Anteile	14
16	Stakeholder Analyse	15
17	Use Case Diagramm <Übersicht>	22
18	Use Case Diagramm <add project>	23
19	Use Case Diagramm <manage project>	24
20	Use Case Diagramm <manage project>	25
21	Use Case Diagramm <change PO>	26
22	Use Case Diagramm <edit user-slots>	27
23	Use Case Diagramm <remove user>	28
24	Use Case Diagramm <register as dev>	29
25	Use Case Diagramm <add Task>	30
26	Klassendiagramm	31
27	Sequenzdiagramm: Load LocalData	32
28	Sequenzdiagramm: Add Project	33
29	Sequenzdiagramm: Load Projects	34
30	Sequenzdiagramm: Register as user	35

Tabellenverzeichnis

1	Initialisierungs Tasks	5
2	Analyse Tasks	6
3	Konzept Tasks	7
4	Umsetzung Tasks	7
5	Deployment Tasks	8
6	TRELLO Details	11
7	TRELLO Bewertung	11
8	Monday.com Details	13
9	Monday.com Bewertung	13
10	Risikoerkennung	19
11	Risikogewichtung	19
12	Massnahmen	20
13	Use Case: ADD PROJECT	23
14	Use Case: MANAGE PROJECT	24
15	Use Case: manage project -> CHANGE PO	26
16	Use Case: manage project -> EDIT USER-SLOTS	27
17	Use Case: manage project -> remove user	28
18	Use Case: register as dev	29
19	Use Case: add task	30
20	Klassenbeschreibung: Projekt	31

7 Anhang

Themeneingabe zur Diplomarbeit

Projektmanagement-Applikation für Unity3D

B-TIN-19-T-a

Standort: Bern

Eingereicht von

Dubois Sascha

Bergernschachenweg3, 3436 Zollbrück

saschadub@gmail.com

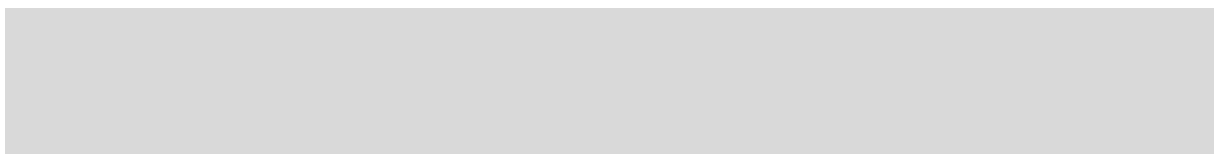
0794615088

Eingereicht am: 29.05.2022

Bewilligt: ☐ Ja / ☐ Nein

Datum: _____ / Unterschrift: _____

Begründung:



Änderungsverzeichnis

Datum	Änderung	Name
14.08.2022	Präzisierungen	Sascha Dubois

Management Summary

Themenbeschreibung

Entwicklung einer Open-Source Projektmanagement-Applikation für Videospiel-Einzelentwickler und kleine Entwicklerteams welche mit Unity3D arbeiten. Dabei sollen die Wichtigsten Funktionen direkt in Unity selber verfügbar sein

Kunde

Potenzielle Kunden sind alle Einzelentwickler / Entwicklerteams, welche mit Unity3D arbeiten

Erfolgskriterien

- Eine WPF-Applikation wurde den Anforderungen (siehe: [Ziele und Anforderungen](#)) entsprechend entwickelt.
- Ein in Unity integriertes Benutzerinterface (Custom Editor) in welchem Aktivitäten zum aktuellen Projekt angesehen/bearbeitet werden können wurde den Anforderungen (siehe: [Ziele und Anforderungen](#)) entsprechend entwickelt.
- Es liegt eine vollständige Dokumentation des Projektes vor
- Die Applikation ist als Open-Source Software lizenziert und zum Download verfügbar

Inhaltsverzeichnis

Änderungsverzeichnis	2
1 Zweck des Dokuments.....	6
2 Ausgangslage	6
2.1 Unity3D	6
2.2 Beschreibung	6
3 Aufgabenstellung.....	6
4 IST-Zustand.....	6
5 Abgrenzung	7
5.1 Generelle Abgrenzung der Arbeit	7
5.2 Geleistete Vorarbeiten.....	7
5.3 Verwendung Erzeugnisse Dritter	7
6 Rahmenbedingungen	7
6.1 Prozessbezogene Rahmenbedingungen	7
6.2 Produktbezogene Rahmenbedingungen	7
7 Ziele und Anforderungen	8
7.1 Ziele.....	8
7.2 Funktionale Anforderungen.....	8
7.2.1 Applikation	8
7.2.2 In Unity integriertes Benutzerinterface (Custom Editor)	8
7.3 Nicht-funktionale Anforderungen	8
8 Projektmanagement.....	9
8.1 Vorgehensmethodik.....	9
8.2 Grobe Zeitplanung	10
9 Nutzen	11
10 Skizze	12
11 Nachträgliche Präzisierungen.....	13
11.1 Umfang des Projektmanagements	13
11.2 Features von Unity3D	13

Glossar

Begriff	Bedeutung
Unity3D / Unity	Entwicklungsumgebung / Engine für Videospiele
Unity-Collaborate	Angebot von Unity zur verteilten Versionsverwaltung von Dateien / Projekten
Custom Editor	Mit «Custom Editor» ist eine Erweiterung des Unity-Editors gemeint. Unity bietet die Möglichkeit, solche Erweiterungen erstellen zu können, standartmässig an. Unity - Manual: Custom Editors (unity3d.com)
Git	Freie Software zur verteilten Versionsverwaltung von Dateien
GitHub	Netzbasierter Dienst zur Versionsverwaltung für Software-Entwicklungsprojekte
Open Source	Als Open Source wird Software bezeichnet, deren Quelltext öffentlich und von Dritten eingesehen, geändert und genutzt werden kann.

1 Zweck des Dokuments

Genaue Beschreibung des zu Bearbeitenden Themas für die Diplomarbeit.

2 Ausgangslage

2.1 Unity3D

«Unity ist eine Laufzeit- und Entwicklungsumgebung für Spiele (Spiel-Engine) des Unternehmens Unity Technologies mit Hauptsitz in San Francisco. Ziel-Spieleplattformen sind neben PCs (PC und Mac) auch Spielkonsolen, mobile Geräte und Webbrowser. Die Entwicklungsumgebung, die die Entwicklung von Computerspielen und anderer interaktiver 3D-Grafik-Anwendungen ermöglicht, gibt es für Windows, Linux (nur Beta) und macOS.» (Wikipedia Unity_(Spiel-Engine))

2.2 Beschreibung

Gamedesign unterscheidet sich in einigen Punkten von der herkömmlichen Softwareentwicklung und wird eher als ein kreativer Prozess angesehen. Vor allem bei kleinen Entwicklerteams und Einzelentwickler manifestiert sich dies besonders: Erfahrungsgemäss wird nur wenig Projektmanagement betrieben und vieles wird spontan entschieden, der Vergleich zum Künstler, der sein Bild nach Gefühl mahlt, ist hier sehr passend. Leider hat dieses Vorgehen einige Nachteile, so werden viele Projekte bekanntermassen nicht beendet oder unfertig publiziert, schwerwiegende Bugs liegen ebenfalls an der Tagesordnung. Zumindest ein grundlegendes Projektmanagement sollte jeder Entwickler führen, denn Videospiele können sehr schnell zur sehr komplexen Code-Suppe werden, bei welcher niemand mehr den Überblick behalten kann.

Ein Grund für das fehlende Projektmanagement ist vermutlich das fehlende Tool dafür. Es gibt zwar viele Projektmanagement-Tools, diese sind jedoch meist für grössere Teams ausgelegt und müssen je nachdem auch bezahlt werden. Der grösste Faktor ist, meiner Meinung nach, jedoch der Mehraufwand den ein solches Tool generiert, denn diese Tools müssen separat gestartet und verwaltet werden. Es wäre doch viel einfacher, wenn ich als Entwickler einfach Unity starten kann und ich alle meine Tasks zum geöffneten Projekt direkt in Unity selber verwalten könnte. Genau dies ist der Ansatz, den ich für meine Diplomarbeit verfolgen möchte.

3 Aufgabenstellung

Es wird eine eigenständige Applikation entwickelt, in welcher Unity-Projekte geplant und verwaltet werden können. Die Applikation soll als Open-Source Software lizenziert und auf GitHub veröffentlicht werden. Die Applikation soll dem Nutzer die Möglichkeit bieten, Projekte nach eigens ausgewähltem Vorgehensmodell zu planen, dass heisst, es wird ein generischer Aufbau der Projekte verlangt.

Ein in Unity integriertes Benutzerinterface (Custom-Editor), in welchem das geöffnete Projekt verwaltet werden kann, ist ebenfalls Teil der Aufgabenstellung.

Projektinformationen, welche von der Applikation erstellt werden, sollen direkt im Projektordner selbst abgespeichert werden, damit diese via Git oder Unity-Collaborate synchronisiert werden können.

Ebenfalls soll es eine Möglichkeit geben, Aufgaben bestimmten Benutzern zuweisen zu können und Meilensteine zu definieren.

4 IST-Zustand

Einzelentwickler und kleine Entwicklerteams verwenden zurzeit (wenn überhaupt) die herkömmlich verfügbaren Tools, um ihre Projekte zu verwalten.

5 Abgrenzung

5.1 Generelle Abgrenzung der Arbeit

Im Rahmen der Diplomarbeit, gibt es keine Möglichkeit Daten auf einer Datenbank zu persistieren. Daten werden lediglich serialisiert im entsprechenden Unity-Projektordner abgespeichert. Die Softwarelösung wird ausschliesslich für Windows entwickelt.

5.2 Geleistete Vorarbeiten

Vorab wurden Tests zwecks der Machbarkeitsüberprüfung durchgeführt. Für das Endprodukt selbst oder dessen Dokumentation werden jedoch keine Vorarbeiten durchgeführt.

5.3 Verwendung Erzeugnisse Dritter

Für die Integration in Unity, werden die von Unity bereitgestellten Bibliotheken verwendet. Verwendung von frei verfügbaren Bibliotheken (z.B. NuGets) für Darstellung und Design wird in der Projektdokumentation beschrieben.

6 Rahmenbedingungen

6.1 Prozessbezogene Rahmenbedingungen

- Das Projekt wird als Einzelarbeit durchgeführt
- Es wird eine vollumfängliche Dokumentation erstellt
- Da das Produkt anschliessend als Open-Source Software lizenziert und veröffentlicht werden soll, wird zudem eine englische Dokumentation des Quellcodes erstellt, diese Dokumentation soll anschliessend als Wiki in GitHub verfügbar sein.

6.2 Produktbezogene Rahmenbedingungen

- Die Softwarelösung wird als WPF-Applikation umgesetzt
- Als Entwicklungsumgebung wird VisualStudio verwendet

7 Ziele und Anforderungen

7.1 Ziele

- Die Lösung ist als Open-Source Software gratis verfügbar
- Die Applikation kann von Einzelentwicklern als auch Entwicklerteams verwendet werden
- Die Applikation kann lokal installiert werden, es wird kein Server benötigt

7.2 Funktionale Anforderungen

7.2.1 Applikation

- Unity-Projekte können in der Applikation geplant werden
- Aktivitäten können in der Applikation erstellt, bearbeitet und gelöscht werden
- Aktivitäten können terminiert werden (Start -und Enddatum) *
- Aktivitäten können kommentiert werden*
- Aktivitäten können priorisiert werden*
- Aktivitäten können punkte zugewiesen werden (z.B. Story Points oder Stunden) *
- Aktivitäten können gruppiert werden (z.B. zu Phasen)
- Gruppen von Aktivitäten können kommentiert werden*
- Gruppen können terminiert werden (Start- und Enddatum) *
- Gruppen können priorisiert werden *
- Gruppen können punkte zugewiesen werden (z.B. Story Points oder Stunden) *
- Bestehende Unity-Projekte können der Applikation hinzugefügt werden
- Meilensteine können in der Applikation geplant werden
- Aktivitäten können Benutzern zugewiesen werden
- Wird ein Unity-Projekt der Applikation hinzugefügt, so wird das entsprechende Custom-Editor-Script dem Projektordner hinzugefügt

7.2.2 In Unity integriertes Benutzerinterface (Custom Editor)

- Aktivitäten zum aktuell geöffneten Projekt können erstellt werden
- Aktivitäten zum aktuell geöffneten Projekt können bearbeitet werden
- Projektinformationen zum aktuell geöffneten Projekt (Beschreibung, Meilensteine usw.) können angesehen werden

7.3 Nicht-funktionale Anforderungen

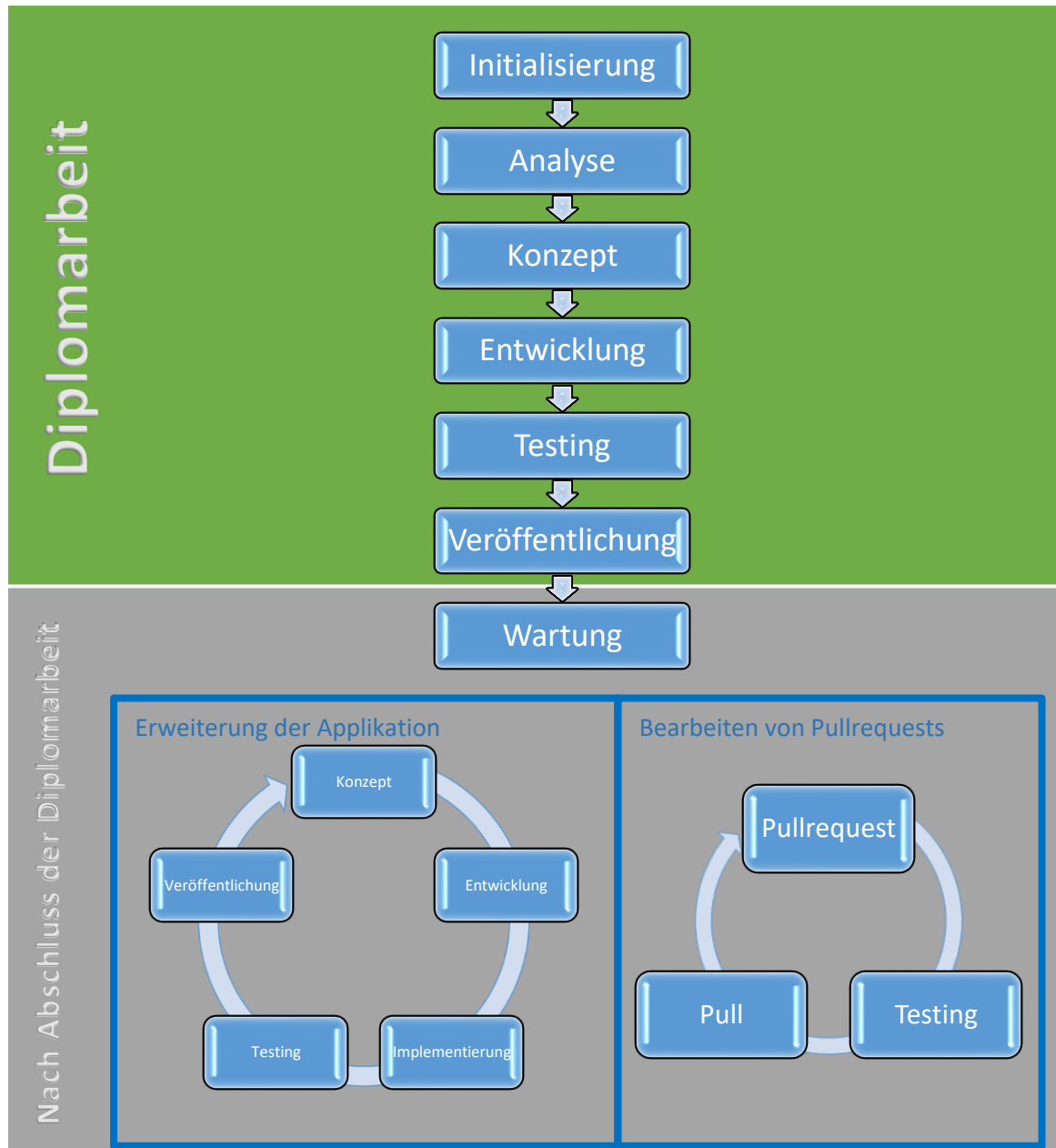
- Projektdaten werden mit dem Projekt zusammen abgespeichert und synchronisiert (z.B. via Git)

* Nachträglich ergänzt 14.08.2022

8 Projektmanagement

8.1 Vorgehensmethodik

Da ich schon einige Erfahrung mit der Entwicklung von WPF-Applikationen sowie mit dem Arbeiten mit Unity habe, entscheide ich mich für das Wasserfallmodell als Vorgehensmodell für das Entwickeln der Applikation im Rahmen der Diplomarbeit. Nach Abschluss der Diplomarbeit werden weitere Features und Verbesserungen in einem Agileren verfahren hinzugefügt.



8.2 Grobe Zeitplanung

Phase	Aktivitäten (Stichpunkte)	Zeitraum	Aufwand
Initialisierung	- Detaillierte Projektplanung	16.09.2022 – 18.09.2022	8-10h
Analyse	- Ziele spezifizieren - Anforderungen erfassen - Risikoanalyse - Dokumentation	19.09.2022 – 25.09.2022	24-28h
Konzept	- Modellierung mittels UML - GUI-Mockup für Applikation und Editor - Testkonzept erstellen - Dokumentation	26.09.2022- 05.10.2022	28-32h
Entwicklung	- Entwickeln der WPF-Applikation - Entwickeln des Custom-Editors - Dokumentation	06.10.2022- 23.10.2022	52-56h
Testing	- Testen nach Testkonzept - Bugfixing - Dokumentation	24.10.2022- 29.10.2022	24-28h
Veröffentlichung	- Readme-Datei erstellen - Open-Source-Lizenz importieren - GitHub Repo veröffentlichen - Dokumentation	30-31.10.2022	8-10h
Präsentation		31.10.2022 – 11.11.2022	

9 Nutzen

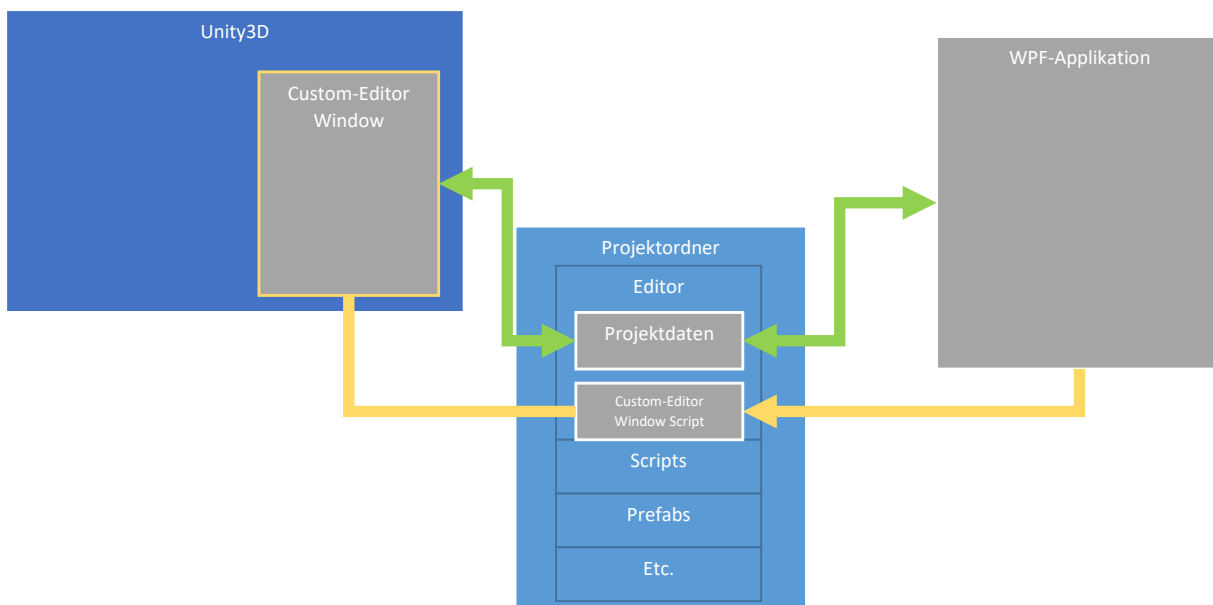
Persönlich:

In meiner Freizeit entwickle ich selbst Videospiele, leider ist das Management dieser Projekte immer sehr frustrierend, da es einfach keine gute Lösung dafür gibt. Ich erhoffe mir persönlich daher einen grossen Nutzen der Lösung und somit auch mehr Spass und Erfolg beim Entwickeln von Videospielen.

Allgemein:

In Online-Foren liest man immer wieder dasselbe Problem: Ein neues Projekt wird aufgrund einer eigentlich guten Idee begonnen aber nach einiger Zeit wird das Spiel so komplex, dass man die Übersicht verliert und das ganze Projekt begräbt. Das ist wohl die Geschichte, die die meisten Einzelentwickler kennen und somit erhoffe ich mir, dass meine Lösung zu einigen vollendeten Spielen und somit auch zu einigen Spielstunden beitragen wird.

10 Skizze



Die Applikation speichert alle Daten zum jeweiligen Projekt in dem entsprechenden Projektordner auf dem Dateisystem, in einem Custom-Editor in Unity können diese Daten ebenfalls angezeigt und bearbeitet werden. Das Custom-Editor-Script wird, beim Importieren des Projektes, von der Applikation im Ordner des Projektes (/Projektname/Assets/Editor/CustomEditorScript) abgelegt. Alle Scripts die in dem /Editor-Ordner abgelegt sind werden von Unity vom Build automatisch exkludiert.

11 Nachträgliche Präzisierungen

11.1 Umfang des Projektmanagements

«Was beinhaltet das Projektmanagement ausser Aktivitäten (und Gruppierung in Phasen), Meilensteine und Ressourcenallokation noch?»

Siehe Änderungen in [«Funktionale Anforderungen»](#)

- Aktivitäten können terminiert werden (Start -und Enddatum)
- Aktivitäten können kommentiert werden
- Aktivitäten können priorisiert werden
- Aktivitäten können punkte zugewiesen werden (z.B. Story Points oder Stunden)
- Gruppen von Aktivitäten können kommentiert werden
- Gruppen können terminiert werden (Start- und Enddatum)
- Gruppen können priorisiert werden
- Gruppen können punkte zugewiesen werden (z.B. Story Points oder Stunden)

11.2 Features von Unity3D

«Welche Features stellt Unity3D für die Integration der Applikation zur Verfügung?»

Unity bietet die Möglichkeit sogenannte «Custom Editors» und «Editor Windows» zu erstellen. «Custom Editors» können als Art Erweiterungen der normalen Editoroberfläche angesehen werden und werden komplett in C# geschrieben. Unity nennt dieses Verfahren zum Erstellen von GUI-Elementen «IMGUI» (Immediate Mode GUI system).

«Editor Windows» sind, wie der Name bereits sagt, eigene Editor Fenster welche innerhalb des Unity Editors geöffnet und andockt werden können.

Unity stellt die üblichen GUI-Elemente wie Buttons, Labels, TextBoxen usw. zu Verfügung. Jedes GUI-Element verfügt über einen GUI-Style welcher individuell angepasst oder erstellt werden kann, so sind die Möglichkeiten zur Erzeugung eigener Editor-Erweiterungen recht umfangreich.

Details zu IMGUI: [Unity - Manual: Immediate Mode GUI \(IMGUI\) \(unity3d.com\)](#)

[Unity - Manual: IMGUI Basics \(unity3d.com\)](#)

Details zu «Custom Editors»: [Unity - Manual: Custom Editors \(unity3d.com\)](#)

Details zu «Editor Windows»: [Unity - Manual: Editor Windows \(unity3d.com\)](#)

Details zu GUI-Styles: [Unity - Manual: GUI Style \(IMGUI System\) \(unity3d.com\)](#)

Des Weiteren verfügt Unity über recht umfangreiche Serialisations-Verfahren, welche das effiziente Speichern und Laden von serialisierten Daten zur Laufzeit ermöglicht.

Details zur Serialisation: [Unity - Manual: Script serialization \(unity3d.com\)](#)