

IPA-BERICHT

SKAP – Unity Spiel



Projektbezeichnung	SKAP	
Erstellt	19.11.2020	
Zuletzt geändert	08.01.2021	
Zustand	<input type="checkbox"/>	In Bearbeitung
	<input type="checkbox"/>	zur Prüfung
	<input checked="" type="checkbox"/>	fertig gestellt
Projektmitglieder	Jennifer Cicvarova	
Auftraggeber	Adrian Imboden	
Projektanfang	12.11.2020	
Projektabgabe	08.01.2021	

Inhaltsverzeichnis

1. Einleitung	2
1.1. Zweck und Ziel des Dokumentes.....	2
2. Ausgangslage	2
2.1. Ausgangslage im Allgemein	2
2.2. Rahmenbedingungen	2
3. Projektmanagement & Planung	3
3.1. IPERKA – Projektmethodik.....	3
3.2. Planung.....	4
4. Testkonzept.....	5
4.1. System	5
4.2. Testfälle	5
4.3. Voraussetzungen	7
4.4. Testmethode	7
5. Arbeitsjournal	8
6. Reflexion	15
7. Kurzfassung	16
7.1. Ausgangssituation.....	16
7.2. Umsetzung.....	16
7.3. Ergebnis.....	16

1. Einleitung

1.1. Zweck und Ziel des Dokumentes

Diese Dokumentation beschreibt die Arbeit und den Implementierungsprozess, die in diesem Modul 150-Projekt durchgeführt wurden.

2. Ausgangslage

2.1. Ausgangslage im Allgemeinen

Das Ziel war es ein IPA-Vorbereitung im Rahmen des Moduls 150 zu gestalten und durchzuführen.

2.2. Rahmenbedingungen

Die nachfolgenden Rahmenbedingungen und Restriktionen sind für das Projekt relevante Vorgaben, die zwingend eingehalten werden müssen:

Nr.	Beschreibung
1	Das Spiel wird mit Unity 3D entwickelt.
2	Das gesamte Spiel wird in C# entwickelt.
3	Das gesamte Spiel wird in der Entwicklungsumgebung von Visual Studio Code entwickelt.
4	Das Spiel funktioniert auf den Plattformen Linux, Windows und MacOSX.

3. Projektmanagement & Planung

3.1. IPERKA – Projektmethodik

Das Projekt ist nach der IPERKA-Methode aufgebaut, damit stets eine gegliederte Struktur im Projektablauf ersichtlich ist. Hierzu sind die 6 folgenden Schritte notwendig oder wie sie auch genannt werden «Phasen».

Informieren:

Der Auftrag und dessen Aufgabenstellung werden so gut wie möglich geklärt, die Informationen werden gewertet und sortiert und wesentliche Punkte so früh wie möglich erkannt sowie allfällige Fragen geklärt. Dies ist wichtig, um den Zeitplan fertigzustellen und zum Verständnis was überhaupt erwartet wird.

Planen:

Um einen guten und strukturierten Projektablauf zu garantieren muss das Projekt präzise geplant werden. Wenn möglich werden mehrere Lösungswege für komplexere Abläufe geplant. Dabei wird ein Testkonzept, sowie ein Realisierungskonzept erstellt.

Entscheiden:

Die unterschiedlichen Lösungswege werden miteinander verglichen und davon wird dann der bestmögliche ausgewählt. Dabei muss man sich überlegen ob die Ideen sinnvoll sind und ob man es dann auch wirklich so umsetzen kann.

Realisieren:

Erst wenn der Ablauf klar definiert ist und alle Entscheidungen getroffen sind, wird mit der Realisierung begonnen. Die Arbeitsabläufe werden stetig protokolliert und die Ist-Werte werden im Zeitplan eingetragen.

Kontrollieren:

Die Resultate müssen so getestet werden, wie es im Testkonzept steht, aber erst wenn die Entwicklung vollständig beendet ist. Fehlgeschlagene Test werden hierbei direkt korrigiert und behoben oder als Fehler vermerkt und dokumentiert.

Auswerten:

Zum Schluss gibt es eine Reflexion über die Arbeit und die Erfahrungen, die man gesammelt hat. Dabei geht man durch alle Schritte, die man innerhalb des Projekts gemacht hat und es wird untersucht was gut war und wo Verbesserungen möglich sind. Das ist auch praktisch für die Umsetzung zukünftiger Projekte.

3.2. Planung

[illegible]

4. Testkonzept

4.1. System

Getestet wurde auf einem Windows 10, Linux und MacOSX System. Die Unity Engine wird von über 25 verschiedenen Plattformen unterstützt. Für uns war allerdings nur Windows relevant.

4.2. Testfälle

Laufen	
Testfall: Der Charakter kann mit den WASD Tasten laufen.	Erwartetes Resultat: Wenn man die WASD Tasten drückt, bewegt sich der Charakter in die Respektive Richtung.
Resultat: Der Charakter kann sich mit den WASD Tasten bewegen.	

Springen mit Space	
Testfall: Der Charakter kann mit den Space Taste springen.	Erwartetes Resultat: Wenn man die Space Taste drückt, der Charakter springt.
Resultat: Der Charakter kann mit den Space Taste springen.	

Sprinten mit Left Shift	
Testfall: Der Charakter kann mit den Left_Shift Taste sprinten.	Erwartetes Resultat: Wenn man die Left_Shift Taste drückt, der Charakter sprintet.
Resultat: Der Charakter kann mit den Left_Shift Taste sprinten.	

Hocken mit Left Ctrl	
Testfall: Der Charakter kann mit den Left_Control Taste hocken.	Erwartetes Resultat: Wenn man die Left_Control Taste drückt, der Charakter hockt.
Resultat: Der Charakter kann mit den Left_Control Taste hocken.	

Waffen wechseln mit Zifferntasten	
Testfall: Der Charakter kann mit den Zifferntasten die Waffen wählen.	Erwartetes Resultat: Wenn man eine von die sechs Zifferntasten drückt, der Charakter wechselt die Waffen.
Resultat: Der Charakter kann mit den Zifferntasten Waffen wechseln.	

Schiessen mit LMB (Linke Maustaste)	
Testfall: Der Charakter kann mit den linke Maustaste schiessen.	Erwartetes Resultat: Wenn man die linke Maustaste drückt, der Charakter schiesst.
Resultat: Der Charakter kann mit den linke Maustaste schiessen.	

Zoomen mit RMB (Rechte Maustaste)	
Testfall: Der Charakter kann mit den rechte Maustaste zoomen.	Erwartetes Resultat: Wenn man die rechte Maustaste drückt, der Charakter zoomt.
Resultat: Der Charakter kann mit den rechte Maustaste zoomen.	

Feindpatrouille	
Testfall: Feinde patrouillieren auf dem Spielfeld.	Erwartetes Resultat: Wenn das Spiel beginnt, patrouillieren Feinde frei auf dem Spielfeld herum.
Resultat: Die Feinde beginnen zu patrouillieren, wenn das Spiel gestartet wird.	

Feind verfolgt	
Testfall: Der Feind verfolgt den Spieler.	Erwartetes Resultat: Wenn der Spieler dem Feind zu nahe kommt, beginnt der Feind, den Spieler zu verfolgen.
Resultat: Der Feind beginnt, den Spieler zu verfolgen, sobald er zu nahe kommt.	

Feindliche Angriffe	
Testfall: Der Feind greift den Spieler an.	Erwartetes Resultat: Wenn sich der Feind vor dem Spieler befindet, greift er an.
Resultat: Der Feind greift den Spieler an, wenn er sich vor dem Spieler befindet.	

Feindliche Angriffe	
Testfall: Der Feind greift den Spieler an.	Erwartetes Resultat: Wenn sich der Feind vor dem Spieler befindet, greift er an.
Resultat: Der Feind greift den Spieler an, wenn er sich vor dem Spieler befindet.	

Gesundheit UI	
Testfall: Der Gesundheit sinkt, wenn der Spieler angegriffen wird.	Erwartetes Resultat: Wenn der Spieler angegriffen wird, sinkt ein Zehntel der Gesundheitsanzeige.
Resultat: Ein Zehntel der Gesundheitsanzeige sinkt, wenn der Spieler angegriffen wird.	

Ausdauer UI	
Testfall: Die Ausdauerleiste sinkt beim Sprinten.	Erwartetes Resultat: Wenn der Spieler mit Left_Shift sprintet, sinkt die Ausdauer in der Ausdauerleiste.
Resultat: Die Ausdauer in der Ausdauerleiste sinkt, wenn der Spieler sprintet.	

Tod des Spielers – Spiel zurückgesetzt	
Testfall: Das Spiel wird zurückgesetzt, wenn der Spieler stirbt.	Erwartetes Resultat: Wenn der Spieler stirbt, wird das Spiel auf den Beginn zurückgesetzt.
Resultat: Das Spiel wird zurückgesetzt, wenn der Spieler stirbt.	

Sound ein-/ausschalten	
Testfall: Man kann den Sound ein-/ausschalten.	Erwartetes Resultat: Im Menu gibt es eine Soundtaste, welche man Sound ein- oder ausschalten kann.
Resultat: Man kann im Settings die Soundtaste klicken und die Sound ein- oder ausschalten.	

4.3. Voraussetzungen

Man kann das Spiel testen mit einem Windows 10, Linux oder MacOSX Computer. Es braucht keine Netzwerkverbindung.

4.4. Testmethode

Als Testmethode haben wir uns für manuelles Testen entschieden. Das Schreiben von Unit Tests hätte mich zu viel Zeit gekostet. Zudem habe ich noch nie Unity Unittests geschrieben.

5. Arbeitsjournal

Name: Jennifer Cicvarova		Thema: SKAP – Singleplayer Survival FPS Game
Datum	Was gemacht?	Reflexion (Schwierigkeiten, Lösungen, was gelernt...?)
12.11.2020	Moduleinführung	Einführung in den M150. Ich habe alle Informationen zum Projekt bekommen.
12.11.2020	Ideen-Brainstorming	Ich habe ein Ideen-Brainstorming gemacht, somit hatte ich eine Themaauswahl.
12.11.2020	Gruppenfindung	Ich habe zu viel Zeit mit Gruppen und Gesprächen mit anderen verschwendet. Einige Leute waren sich nicht sicher, ob sie in Gruppen sein wollten oder ob sie allein arbeiten wollten. Aus diesem Grund musste ich ein wenig warten, bevor ich eine Antwort bekam.
12.11.2020	Thema wählen	Ich habe aus meinen Themaauswahl ein Thema ausgesucht.
12.11.2020	Projekteinführung	Ich habe meine Github Repository erstellt.
12.11.2020	Backlog erstellen	Ich habe ein Backlog auf Github erstellt.
19.11.2020	Todos erstellen, Spiele definieren	Ich habe eine TODO-Liste mit allen Aufgaben erstellt und mein Spiel genauer definiert.
19.11.2020	Unity Projekt erstellen (Projekt (Level) vorbereitung)	Ich habe ein Unity-Projekt erstellt und angefangen, Sachen vorzubereiten, die ich für das Projekt benötigen würde. (verschiedene Assets: Soundeffekte, Menschmodell, Tiermodell usw.)
19.11.2020	Beginnen mit der Einstellung von CI/CD	Ich habe begonnen, CI / CD für mein Unity-Projekt festzulegen (in Github Actions). Ich habe auf youtube ein hilfreiches Tutorial gefunden, mit dem ich das gemacht habe. Ich habe es heute nicht geschafft, also werde ich morgen weitermachen.
20.11.2020	Einstellung von CI/CD fertig machen	Ich habe die Einstellung von CI / CD für mein Unity-Projekt fertig gemacht und es funktioniert, wie es sollte.
26.11.2020	Nutzbare Assets gefunden	Ich habe mehrere Websites durchsucht (z.B. Unity-Assetstore), um Assets zu finden, die ich in meinem Projekt verwenden kann. Ich habe einige Zeit gebraucht, um gutaussehende Assets zu finden, die kostenlos waren. Nach einiger Zeit fand ich einige nützliche Assets (z.B. Texturen, Modelle, Soundeffekte...).
26.11.2020	Beginnen mit dem Projekt (Level-Map erstellen)	Heute habe ich mit der Erstellung des Terrains begonnen. (Kartentexturen, Beleuchtung ...)

03.12.2020	Level Map hinzugefügt, Beleuchtung fertig	Ich habe den Gameplay-Bereich aus dem Prefabs-Ordner in Assets importiert. Die Beleuchtung war nicht schwierig, aber ich musste suchen, wie es genau funktioniert, da ich zum ersten Mal mit dem gerichteten Licht (Lichteffekte) arbeitete.
03.12.2020	Beginnen mit dem Spieler (Player)	Ich habe ein GameObject erstellt ("Player" umbenannt). Dann habe ich eine "Character Component" hinzugefügt (und die Values dort angepasst). Ich habe auch einige Dinge erstellt, wie die Person_Camera und Person_weapons (zeigt Hände mit Waffen). Im Moment arbeite ich an den Modellen und Animationen (assets/models/firstperson). Es wird wahrscheinlich etwas mehr Zeit dauern, da ich noch nie mit solchen Modellen gearbeitet habe. (Die Spiele, die ich in der Vergangenheit gemacht habe, waren 2D)
10.12.2020	Fertig mit dem Spieler Modellen	Ich habe die Spielermodelle (Waffen und Hände) fertig eingerichtet. Ich habe die Positionen angepasst und die Texturen hinzugefügt.
10.12.2020	Beginnen mit der Spielerbewegung	Ich fing an, die Spielerbewegung zu codieren. Es gab keine grossen Probleme, weil ich es zuvor getan habe. Ich habe die Spielerbewegung (WASD-Tasten) und die Mausbewegung (nach oben, unten, links, rechts) gemacht.
13.12.2020	Spielerbewegung fertig	Ich habe die Spielerbewegung fertiggestellt. Ich habe die Sprint-, Jump- und Crouchfunktionen zu der Bewegung hinzugefügt.
16.12.2020	Animieren von Händen und Waffen	Ich habe die Animationen zu den Modellen hinzugefügt und Animationsschritte (Weapon_Draw, Weapon_Idle, Weapon_Aim, Weapon_Attack) im "Animator" erstellt. Animationen waren etwas Neues für mich, daher war es wirklich interessant, online darüber zu lesen und zu sehen, wie sie funktionieren.
21.12.2020	Füge die Schritte des Spielers hinzu	Ich habe die Schritte des Players hinzugefügt. Ich habe zuerst mein PlayerFootsteps-Skript erstellt und diese Funktionen dann im PlayerAbilities-Skript hinzugefügt. Es gibt ein minVolume und ein maxVolume, denn wenn du rennst, bist du lauter und wenn du dich hockst, bist du leiser.

23.12.2020	Waffenhandhabungsfehler	Ich habe heute am Waffenhandler gearbeitet. Nachdem ich alles beendet und festgeschrieben hatte, wurde ein Fehler in meinem CI-Workflow angezeigt. Ich habe versucht, meine Schritte zurückzuverfolgen, um festzustellen, welcher Teil von dem, was ich hinzugefügt habe, falsch ist, aber ich konnte heute keine Lösung finden.
24.12.2020	Waffenhandhabungsfehler - Player-Prefab löschen	Heute habe ich daran gearbeitet, den Fehler von gestern zu korrigieren. Zuerst habe ich nicht verstanden, warum es einen Fehler gab. Der Code war in Ordnung und es gab keinen Grund für Unity, ein Problem mit dem Hinzufügen von AudioSources zu GameObjects zu haben. Dann habe ich mir den Fehler noch einmal angesehen. Über der Fehlermeldung stand "Nicht behandelte NULL-Ausnahme empfangen" und ich hatte die Idee, dass mein Player-Prefab das Problem sein könnte. Das Problem ist, dass das von Unity ausgeführte Skript den Pfad / die ID nicht findet, um das Basis-Prefab zum Aktualisieren zu veranlassen. Das Löschen des Fertighauses hat den Fehler erloscht.
27.12.2020	"added weapon-handler" und "deleted player prefab" commits gepushed	Ich hatte einige Probleme mit dem Branch, also habe ich einen neuen erstellt. Deshalb ist der Push am 27./28. Und nicht am 23./24.
28.12.2020	Waffenmanager hinzugefügt	Ich habe ein WeaponManager-Skript erstellt, in dem Sie mit den Zifferntasten zwischen den verschiedenen Waffen wechseln können. Es war ziemlich einfach. Keine Schwierigkeiten.
30.12.2020	Spielerangriff hinzugefügt	Ich habe ein PlayerAttack-Skript erstellt, in dem ich die Grundlagen für die Schiessen-Funktionen geschaffen habe, die ich später hinzufügen werde. Im Moment werden nur die Animationen abgespielt, wenn Sie mit der linken Maustaste (LMB) klicken.
30.12.2020	Waffenanimationen aktualisiert	Ich habe die Waffenanimationen aktualisiert. Sie sollten nicht warten, bis die Animation beendet ist, und dann zur Angriffsanimation gehen. Die Idle-Animation sollte anhalten und zur Angriffsanimation wechseln, sobald Sie mit der LMB (Linke Maustaste) klicken. Ich habe die Exit-Zeit deaktiviert und eine neue TransitionDuration festgelegt, je nachdem, wie schnell der Übergang erfolgen soll.

30.12.2020	Crosshair mit Animationen hinzugefügt	Ich habe ein Crosshair erstellt, indem ich eine Canvas mit einem Bild und einem Sprite-Bild des Crosshairs hinzugefügt habe. Ich habe auch Animationen hinzugefügt, wenn Sie zoomen oder mit einer Waffe zielen. Ich fühle mich bei Animationen jetzt viel besser als zuvor. Am Anfang habe ich zum ersten Mal mit Animationen gearbeitet, daher war ich mir nicht sicher, wie es genau funktioniert.
30.12.2020	Zoomfunktion hinzugefügt	Ich habe das Crosshair und die Zoomfunktion zum PlayerAttack-Skript hinzugefügt. Wenn Sie nun auf die RMB (rechte Maustaste) klicken und diese gedrückt halten, können Sie mit Ihren Waffen zoomen / zielen.
01.01.2021	Munition Prefabs erstellt	Ich habe einen Ammo-Prefabs-Ordner für die Arrow- und Spear-Prefabs erstellt. Ich werde diese später verwenden, wenn ich mit dem Arrow- und Spearskript beginne.
01.01.2021	Weapon_Arrow_Spear Skript erstellt	Da Bogen und Speer anders funktionieren als die anderen Waffen, habe ich ein WeaponArrowSpear-Skript erstellt, in dem ich die von mir erstellten Prefabs verwende. Ich habe auch die ShootArrowOrSpear-Funktion erstellt, bei der ich eine Kopie des Pfeils oder des Speer-Fertighauses erstelle und in Richtung von Crosshair starte.
02.01.2021	Menschlichen Feind hinzufügen	Ich habe heute angefangen, an den Feinden zu arbeiten. Ich habe mit dem menschlichen Feind angefangen. Ich begann mit den Animationen, indem ich einen Animations-Controller erstellte und nur auf spätere Sachen vorbereitete (Skripte usw.).
02.01.2021	Tierfeind hinzugefügt	Genau wie beim menschlichen Feind, nur um die Dinge vorzubereiten.
02.01.2021	Release Alpha_1.0	Ein Alpha_1.0 Release erstellt. (GitHub -> Releases)
02.01.2021	EnemyAnimation-Skript erstellen	Ich habe ein EnemyAnimationsskript mit nur einigen Animationsfunktionen für Gehen, Laufen, Angriff und Tod erstellt. Keine Schwierigkeiten. Im Moment konzentriere ich mich darauf, das Projekt rechtzeitig abzuschliessen.

03.01.2021	EnemyController erstellt	Heute habe ich das EnemyController-Skript erstellt. Wo ich das Patrol, Chase, Attack-System gemacht habe. Die Feinde werden "patrollieren" und um das Spiel herumgehen. Wenn wir sie erschiessen oder wenn wir ihnen nahe sind, werden sie uns verfolgen.
03.01.2021	Feinde und der Spieler können Damage verursachen und erkennen	Ich habe das PlayerDetectDamage-Skript erstellt. An diesem Punkt können Feinde und der Spieler Damage verursachen und erkennen. Ich habe das mit AttackPoints gemacht. Der AttackPoint des menschlichen Feindes befindet sich auf seinem rechten Arm. Schweine sind auf der Schnauze und der Spieler ist auf der Axt. Zusammen mit Animationen habe ich genau festgelegt, wann die AttackPoints aktiv werden.
03.01.2021	Release Alpha_1.1	Ein Alpha_1.1 Release erstellt. (GitHub -> Releases)
04.01.2021	Health Skript hinzufügen	Ich habe das Health-Skript erstellt. Nachdem der menschliche Feind getötet wurde, wird dieses Objekt deaktiviert. Wenn der Spieler stirbt, wird das Spiel neu gestartet. Der Grund, warum der Tod des menschlichen Feindes nicht so gut aussieht wie der des Tieres, liegt darin, dass ich keine Todesanimation für den menschlichen Feind in meinem Assets habe.
04.01.2021	Stamina und Health UI erstellt	Zuerst habe ich die UI-Objekte erstellt und dann die Spielerstatistiken in den Code eingefügt. Ich habe das PlayerStats-Skript erstellt. Ich habe auch einige Funktionen aktualisiert.
04.01.2021	EnemyManager Skript hinzugefügt	Ich habe das EnemyManager-Skript erstellt. Dieses Skript stellt sicher, dass ein weiterer Feind erzeugt wird, nachdem wir einen getötet haben, damit wir weiterhin Feinde auf dem Spielfeld haben können. Ich habe auch ein paar GameObjects erstellt. (z.B. "SP_1") Diese Objekte sind die EnemySpawnPoints.
04.01.2021	Die Position der Spawnpunkte wurde geändert	Ich habe die Position der Spawn-Punkte geändert, damit die Feinde näher am Spieler sind.
05.01.2021	Angepasster Damage	Ich habe den Damage angepasst, weil ich das Gefühl hatte, die Feinde seien zu stark. Diese kleinen Änderungen, von denen ich weiss, dass sie keinen Fehler verursachen, werden an den Haupt-Branch übertragen. Ich habe nur einige Zahlen geändert.

05.01.2021	Waffen Soundeffekte hinzugefügt	Ich habe den Waffen Soundeffekte hinzugefügt. Ich musste ein Skript für die Axt erstellen, da es 3 Axtangriffs-Audiodateien gibt. Es war einfach, hier keine Schwierigkeiten.
05.01.2021	Feinde Soundeffekte hinzugefügt	Ich habe das EnemySound-Skript mit den Audiofunktionen Schrei, Angriff und Tod erstellt. Dann habe ich einige andere Skripte aktualisiert, in denen ich den Kommentar "Audio abspielen" vorbereitet hatte.
05.01.2021	Bogen und Speer aktualisiert	Es gab ein Problem mit der Zoom- / Zielanimation. Sie würden zoomen, schiessen und dann funktionierte es nicht mehr. Ich habe es korrigiert, indem ich einen weiteren Übergang von Angriffsanimation zu Idle-Animation hinzugefügt habe.
05.01.2021	CI/CD Workflows	Das CI funktioniert aber nicht vollständig. Ich habe versucht, es zum Laufen zu bringen, aber es hat nicht funktioniert. Ich habe im Internet gesucht und auch andere Leute gefragt, aber niemand konnte mir helfen. Ich bin wütend, dass ich dadurch so viel Zeit verloren habe.
06.01.2021	Unsichtbare Wände hinzugefügt	Ich habe unsichtbare Wände hinzugefügt, um sicherzustellen, dass der Spieler nicht vom Rand des Spielfelds fällt.
06.01.2021	Assets hinzugefügt	Ich habe die Sound-, Bild- und Schriftarten-Assets zum Feature_Menu Branch hinzugefügt.
06.01.2021	Menu- und Settings Szene erstellen	Ich habe das Hauptmenü und die Einstellungsszene erstellt. Es war ein bisschen schwierig zu entscheiden, wie das Menü und die Einstellungsseite aussehen sollen, weil ich möchte, dass es gut aussieht. Die Menüseite sieht gut aus und passt zur Überlebensästhetik, und die Einstellungsseite enthält ein "How To Play" -Bild mit der "Musik ein- / ausschalten" Taste.
06.01.2021	Menu-Skripts erstellen	Ich habe einige Skripte für die Menü- und Einstellungsseiten erstellt. Einige waren für mehr Designzwecke (ButtonColor) und andere für Funktionszwecke (MenuBehavior).

08.01.2021	Settings Sound-Taste funktioniert jetzt	Ich habe die Sound-Taste korrigiert. Wenn Sie auf die Sound-Taste klicken, wird der Sound ausgeschaltet. Ich habe auch eine Funktion hinzugefügt, mit der Ihre Entscheidung gespeichert wird. Wenn ich die Musik ausschalte, wird die Musik beim nächsten Start des Spiels ausgeschaltet.
08.01.2021	Unnötigen Code / Dateien gelöscht	Ich habe eine Szene gelöscht, die ich nie verwendet habe, und Code, von dem ich dachte, ich würde ihn verwenden, aber nicht. Diese kleinen Änderungen, von denen ich weiss, dass sie keinen Fehler verursachen, werden an den Haupt-Branch übertragen. Ich habe nur einige Zahlen geändert.
08.01.2021	Mehr Spawnpunkte hinzugefügt	Ich habe mehr Spawnpunkte über das Spielfeld hinzugefügt, um mehr Feinde hinzuzufügen und damit das Spielfeld nicht so leer sind. Diese kleinen Änderungen, von denen ich weiss, dass sie keinen Fehler verursachen, werden an den Haupt-Branch übertragen. Ich habe nur einige Zahlen geändert.
08.01.2021	Release Final	Ein Alpha_1.1 Release erstellt mit Windows, Linux und MacOSX ZIP-Dateien. (GitHub -> Releases)
08.01.2021	Projekt Abgabe	Ich bin etwas nervös, aber ich denke, dass ich insgesamt gute Arbeit gemacht habe.

6. Reflexion

Dieses Modul oder das Thema dieses Moduls war wirklich gut. Es ist wirklich gut, dieses Modul als Einführung und als Vorbereitung für die IPA zu haben. Ich bin froh, dass wir das Thema des Projekts auswählen konnten. Ich mag Videospiele und ich interessiere mich für sie. Die Gelegenheit, wieder mit Unity zu arbeiten, hat Spass gemacht. Es war manchmal ziemlich stressig. Anfangs habe ich nicht viel gearbeitet, weil ich mich darauf konzentriert habe, meine ABU-Vertiefungsarbeit zu beenden. Ich muss mehr an meinem Zeitmanagement arbeiten. Was mich sehr gestört hat, waren die CI / CD-Pipelines. Ich habe sehr viel Zeit verloren. Ein weiteres Problem war, dass einige Ziele nicht genau bedingt waren und leider mehr Stress gaben als ein Leitfaden.

In der Ansicht auf das Projekt bin ich sehr zufrieden mit dem Produkt. Ich bin froh, dass ich alles fertig habe, was ich wollte. Das Spiel sieht gut aus und die Leute, die es für mich getestet haben, sagten, dass es viel Spass macht.

7. Kurzfassung

7.1. Ausgangssituation

Erstelle ein Survival First-Person-Shooter (Überlebens-Ego-Shooter) Spiel, in dem du durch das Spielfeld gehst und Feinde tötest. Dieses Spiel sollte den folgenden obligatorischen Inhalt enthalten:

- **Spieler:** Der Spieler kann sich frei im Spielfeld bewegen. Der Spieler kann auch springen, sprinten und sich ducken.
- **Waffen:** Der Spieler kann aus mehreren Waffen wählen.
- **Feind:** Es gibt zwei Arten von Feinden (menschliche und tierische Feinde), die auf dem Spielfeld herumlaufen. Der Spieler kann sie mit den Waffen angreifen und töten.
- **Soundeffekte:** Gegner, Waffen und Spieler haben Soundeffekte.
- **UI:** In der oberen linken Ecke des Bildschirms befinden sich eine Gesundheitsleiste und eine Ausdauerleiste.

7.2. Umsetzung

Die Umsetzung verlief in der Projektmethode IPERKA. Das Spiel wurde mit dem Programm Unity¹ gemacht. Für die C# -Dateien habe ich VSCode² verwendet. Die Assets, die ich in diesem Spiel verwendet habe, stammen aus dem Unity Asset Store³. Das Produkt Backlog⁴ wurde auf GitHub unter Projects erstellt.

Nachdem ich nützliche und kostenlose Assets gefunden hatte, begann ich an dem Spiel zu arbeiten. Der Hauptfokus lag darauf, sicherzustellen, dass der obligatorische Inhalt voll funktionsfähig ist. Der Code wurde auf einfache und verständliche Weise mit Kommentaren zur Erklärung geschrieben.

7.3. Ergebnis

Das Ergebnis ist ein funktionierendes Spiel, das genau so funktioniert, wie es sollte. Zum Herunterladen müssen Sie nur die Zip-Datei einer dieser Plattformen herunterladen: Windows, Mac, Linux und die Exe-Datei öffnen.

Der Spieler kann laufen, springen, sprinten und sich ducken. Der Spieler hat 6 Waffen, aus denen er wählen kann. Die Schritte des Spielers sind hörbar und die Lautstärke ändert sich je nach Geschwindigkeit (Sprint, Hocke, Gehen).

Die Waffen haben Animationen, Soundeffekte und visuelle Effekte (Mündungsblitz). Es gibt auch ein Fadenkreuz, das jedes Mal erscheint, wenn Sie eine Schiesswaffe ausrüsten.

Insgesamt gibt es 10 Spawnpunkte, die die Feinde erscheinen lassen (4 Tiere, 6 Menschen). Feinde haben Animationen und Soundeffekte. Sie werden dich bemerken, sobald du nah genug bist und dich angreifen.

¹ Unity: <https://unity.com>

² Visual Studio Code: <https://code.visualstudio.com>

³ Asset Store: <https://assetstore.unity.com>

⁴ Produkt Backlog mit alle Todos: https://github.com/JennCA/M150_Unity_Game/projects/1