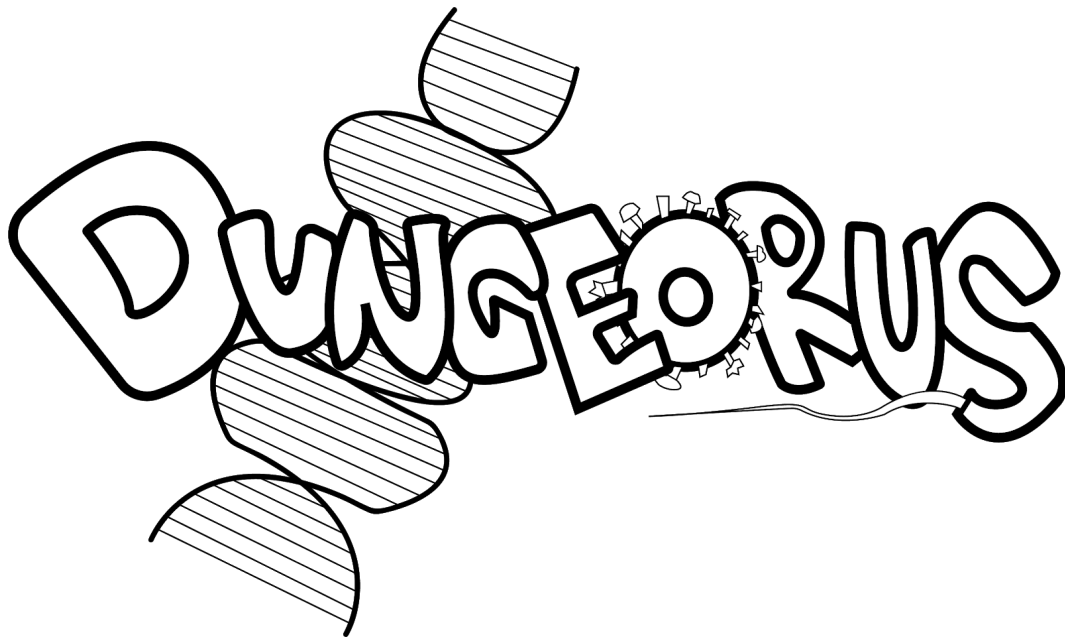


Pflichtenheft



Ein 2D Top-Down Pixel Art Spiel

Autoren des Dokuments	: Adib Ghassani Waluya Minh Hoang Do Michael Reno
Letzte Änderung	: Berlin, 21.05.2020
Dateiname	: Pflichtenheft - Dungeorus
Version	: 1.0

Inhaltsverzeichnis

1	Überblick	3
2	Hauptziele	5
3	Annahmen und Abgrenzungen	5
4	Workflow	6
5	Funktionalitäten	7
5.1	Überblick	7
5.2	F1 - Neues Spiel starten	10
5.3	F2 - Pause-Menü aufrufen	11
5.4	F3 - Charakter auswählen	12
5.5	F4 - Charakter bewegen	13
5.6	F5 - Fähigkeiten einsetzen	14
5.7	F6 - Schlagen	15
5.8	F7 - Schießen	16
5.9	F8 - Charaktere / Items / Gegner / Levels designen	17
5.10	F9 - Charaktere / Items / Gegner / Levels anlegen	18
5.11	F10 - Item aufsammeln	19
5.12	F11 - Charakter heilen	20
5.13	F12 - Gegner erzeugen	21
5.14	F13 - Highscore berechnen	22
6	Backlog	23
7	Aufteilung der Kapitel	23

Versionshistorie

Version	Datum	Autoren	Bemerkung
0.1	06.05.2020	Michael Reno	Initiale Dokumenterstellung
0.2	12.05.2020	Minh Hoang Do, Michael Reno, Adib Ghassani Waluya	Erweiterung (Überblick, Ziele, Workflow, Funktionalitäten)
0.3	16.05.2020	Michael Reno, Minh Hoang Do, Adib Ghassani Waluya	Erweiterung (Funktionalitäten)
1.0	21.05.2020	Adib Ghassani Waluya, Minh Hoang Do, Michael Reno	Erweiterung, letzte Veränderung und Fertigung

Vorhandene Dokumente

Dokumente	Autoren	Datum
Lastenheft	Adib Ghassani Waluya, Michael Reno, Minh Hoang Do	29.04.2020
Lastenheft + Kommentare	Prof. Dr.-Ing. Mohammad Abuosba	02.05.2020
Projektplan	Prof. Dr.-Ing. Mohammad Abuosba, Adib Ghassani Waluya, Minh Hoang Do, Michael Reno	21.05.2020

1 Überblick

Anhand des Lastenheftes zum Projekt “Dungeorus” wird ein 2D-Shooter Spiel realisiert. Es soll folgende Anforderungen erfüllen:

Bei der Grafik des Spiels soll ein 2D-Pixel Stil verwendet werden, das heißt Objekte sowie Charaktere in der Spielwelt können sich nur in der vertikalen sowie horizontalen Achse bewegen. Es werden eigens konzipierte 2D-Assets genutzt für Charaktere, Items sowie Gegner und frei zugängliche Assets für das Leveldesign.

Im Hauptmenü kann der Benutzer wählen, ob er sich entweder das Scoreboard anschaut oder ein neues Spiel startet und daraufhin einen Charakter auswählt. Sobald dieser in das Level geladen wird steuert der Benutzer ihn aus der Top-Down Perspektive.

Im Spiel hat der Benutzer einen Lebensbalken sowie einen Score, der mit besiegten Gegnern zunimmt. Der Benutzer kann Items aufheben wie zum Beispiel Schwerter, Laserpistolen, Schutzmasken, Desinfektionsmittel oder Seife um entweder seinen Lebensbalken aufzufüllen oder Fähigkeiten freizuschalten.

Beim Aufheben von Hygieneartikeln soll ein Dialogfenster erscheinen mit der Meldung, warum dieses Item während einer Pandemie wichtig ist und wie es zu benutzen ist.

Das Spiel endet, wenn der Benutzer von Gegnern besiegt wurde und der Lebensbalken aus null gesunken ist. Daraufhin soll der erreichte Score angezeigt werden und der Benutzer soll entscheiden, ob er ein neues Spiel starten will oder nicht.



Abbildung 1: Skizze des Spiels (In-Game)

2 Hauptziele

#	Ziel	Beschreibung der Implementation	Front- / Back-End
1	Entwicklung der Spiellogik	Unity 2019, C# mit Visual Studio 2019	Back-End
2	Designen der Charaktere, Items, Gegner	Adobe Photoshop, Illustrator, After Effects, Aseprite	Front-End
3	Designen der Level	Unity 2019	Front-End
4	Vermittlung der Hygieneregeln	Unity 2019	Front-End
5	Aktualisierung des Highscores	Unity 2019, C# mit Visual Studio 2019	Back-End

Tabelle 1: Hauptziele

3 Annahmen und Abgrenzungen

#	Annahmen (fachliche und technische Annahmen)
1	Es ist ein Offline-Spiel, daher wird keine Internetverbindung benötigt.
2	Die Highscores werden lokal gespeichert.
3	Die minimale Systemvoraussetzung ist Android 7.0 (Nougat).

Tabelle 2: Annahmen

#	Abgrenzungen (was ist in dieser Lösung nicht enthalten bzw. abgedeckt)
1	Es gibt kein Multiplayer Feature.
2	Der Spielstand wird nicht gespeichert.

Tabelle 3: Abgrenzungen

4 Workflow

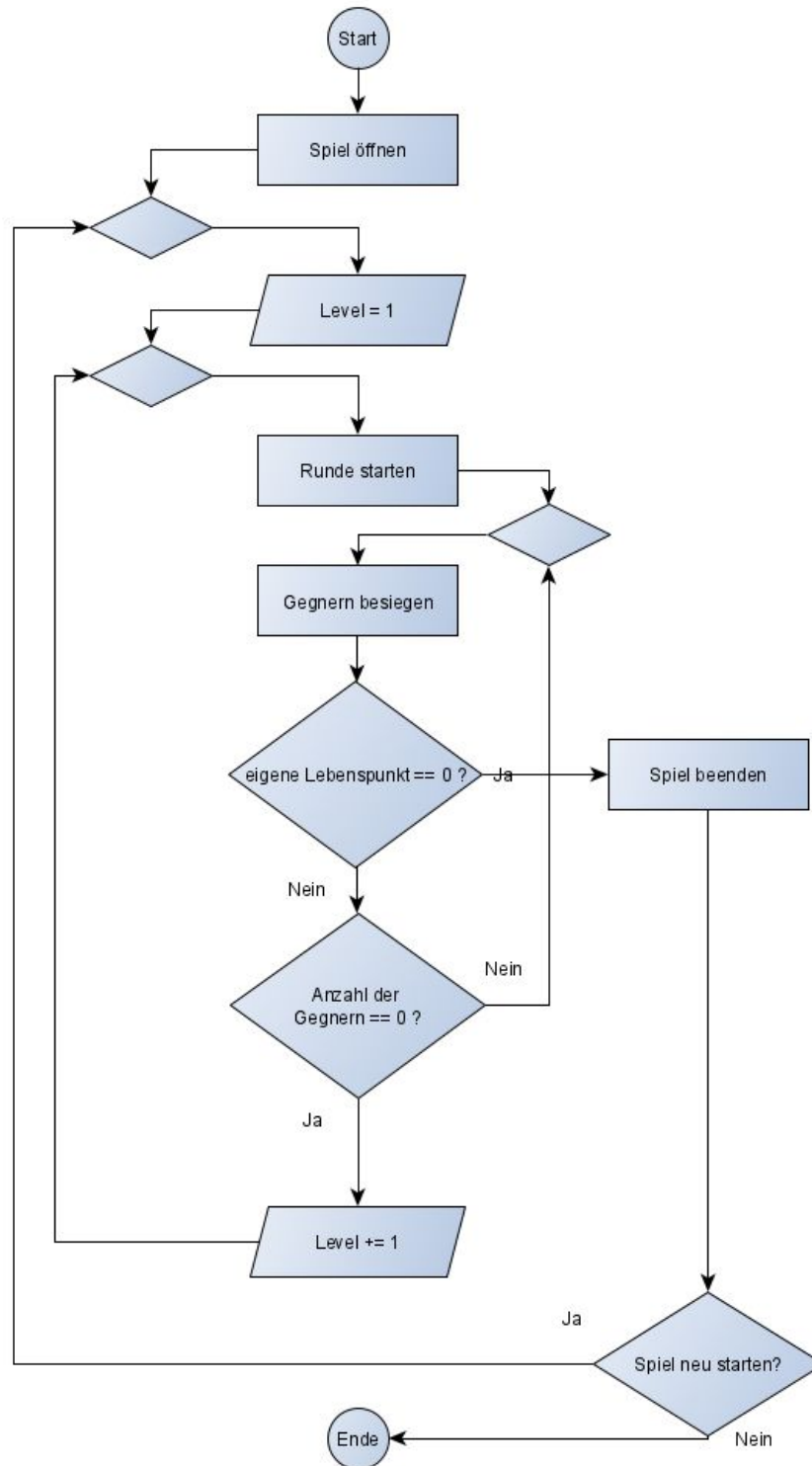


Abbildung 2: Workflow-Diagramm

5 Funktionalitäten

5.1 Überblick

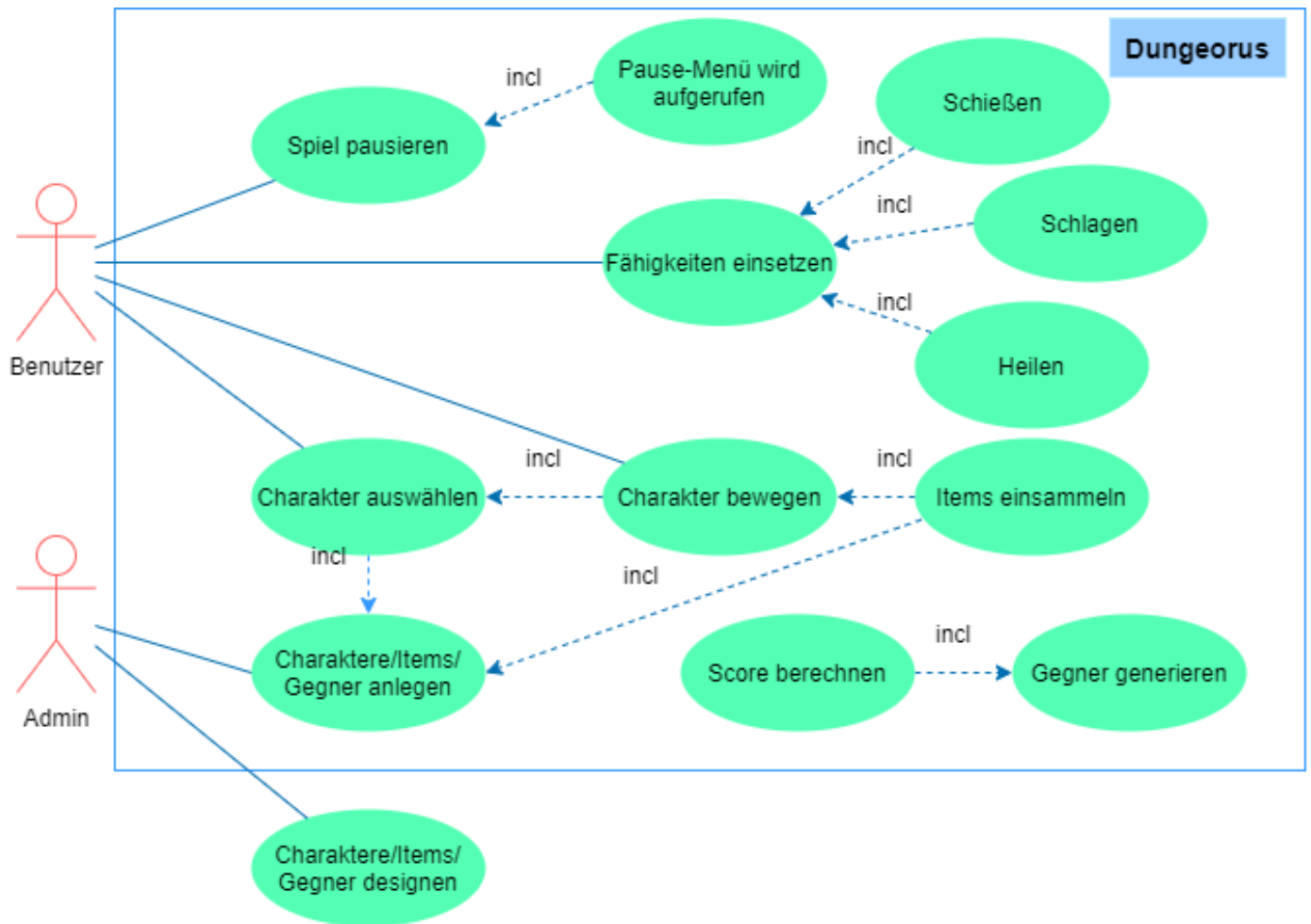


Abbildung 3: Use-Case Diagramm

Layer 1	Layer 2	Layer 3	Layer 4
Game's Design (Dungeorus Logo)	Start Game	Charakter auswählen	Gameplay
	Einstellungen	Audio konfigurieren	
	Scoreboard	Scores anzeigen	
		Scoreboard zurücksetzen	
	Quit		

Tabelle 4: Layer des Spiels

Neues Spiel starten

Der Benutzer kann im Hauptmenü oder im Pause-Menü während des Spiels eine neue Runde starten und wird daraufhin in die Charakterauswahl weitergeleitet.

(Siehe 5.2 "Neues Spiel starten")

Charakterauswahl

Der Benutzer kann sich in diesem Menü einen von mehreren Charakteren auswählen. Diese unterscheiden sich sowohl vom Aussehen als auch von den Fähigkeiten und Eigenschaften wie Lebenspunkte und Bewegungsgeschwindigkeit.

(Siehe 5.4 "Charakter auswählen")

Charakter bewegen

Um den Charakter in X- und Y-Richtung sich bewegen zu können, kann ein programmierter mobiler Analog-Joystick verwendet werden.

(Siehe 5.5 "Charakter bewegen")

Fähigkeiten einsetzen

Jeder Charakter verfügt über eine einzigartige Fähigkeiten, die er im Spiel nutzen kann. Außerdem können anderen freigeschaltet werden, indem der Spieler Items sammelt.

(Siehe 5.6 "Fähigkeiten einsetzen"; 5.11 "Items aufsammeln")

Schlagen/Schießen/Heilen

Der Charakter kann mit den Buttons "Schlagen" sowie "Schießen" Gegner besiegen sowie mit dem "Heilen"-Button seinen Lebensbalken erhöhen.

(Siehe 5.7 "Schlagen"; 5.8 "Schießen"; 5.12 "Heilen")

Charaktere / Items / Gegner / Levels designen und anlegen

Charakter-, Item- sowie Gegner-Assets müssen gezeichnet, animiert und daraufhin in das Spiel implementiert werden.

Die Levels werden aus vorhandenen Assets erstellt.

(Siehe 5.9 “Charakter/Items/Gegner/Levels designen”; 5.10 “Charakter/Items/Gegner/Levels anlegen”)

Pause Menü

Innerhalb des Levels kann der Benutzer über einen Button das Pause-Menü aufrufen.

Darin kann der Benutzer entscheiden, ob er das Spiel fortsetzt, beendet oder ein neues startet.

(Siehe 5.3 “Pause-Menü aufrufen”)

Gegner generieren

Innerhalb des Levels sollen verschiedene Gegner generiert werden, die der Benutzer zu besiegen hat. Anhand dessen soll der Highscore berechnet werden.

(Siehe 5.13 “Gegner generieren”)

5.2 F1 - Neues Spiel starten

Zweck / Ziel	Mit dieser Funktion wird eine neue Runde gestartet.
Akteur / Auslöser	Spieler
Vorbedingung	-
Daten-Input	Eingabe
Verarbeitungsschritte	<ol style="list-style-type: none"> 1. Prozess starten. 2. Fall 1: Spieler wählt "Neues Spiel / Spiel Starten" im Hauptmenü. Fall 2: Spieler wählt "Neues Spiel / Spiel Starten" im Pause-Menü.
Ergebnis	Der Spieler wird in die Charakterauswahl weitergeleitet.
Plausibilitäten / Test Cases	Nachdem tippen auf den "Neues Spiel" Button, muss das Charakterauswahl-Menü geöffnet werden.
Fehlerbehandlung	Im Fehlerfall soll das Hauptmenü wieder aufgerufen werden.
Folgeprozess	F3 - Charakter auswählen

Tabelle 5: F1 - Neues Spiel starten

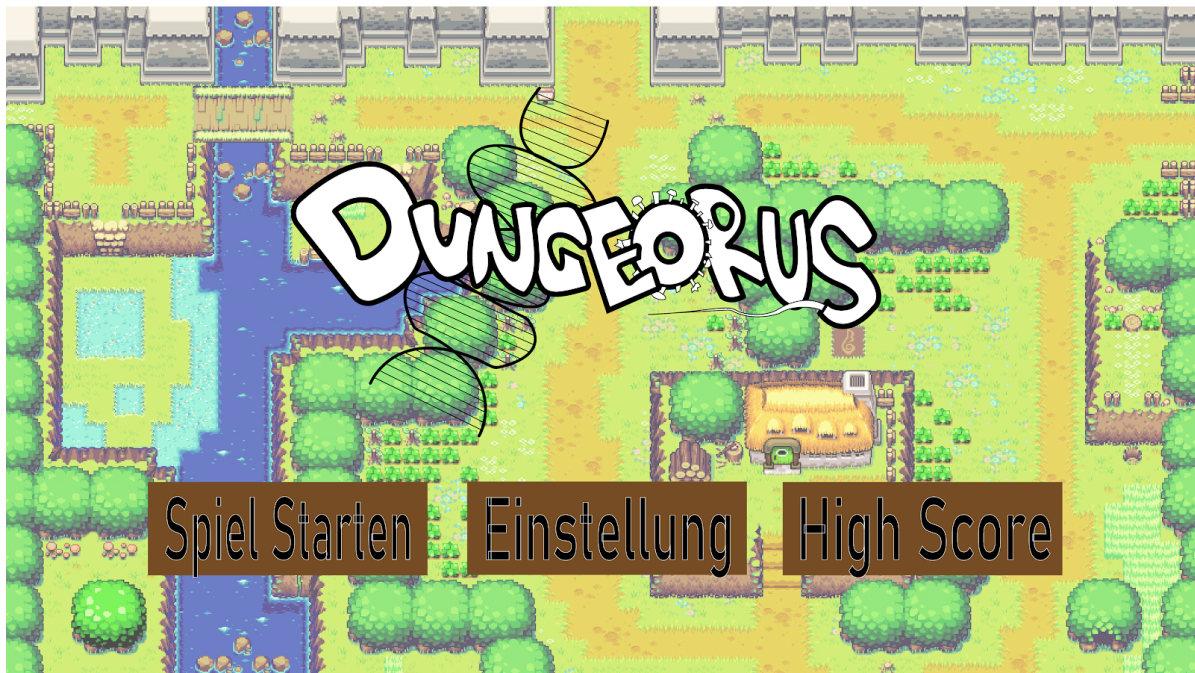


Abbildung 4: Skizze von Hauptmenü

5.3 F2 - Pause-Menü aufrufen

Zweck / Ziel	Mit dieser Funktion wird das Pause-Menü aufgerufen.
Akteur / Auslöser	Spieler
Vorbedingung	-
Daten-Input	Eingabe
Verarbeitungsschritte	<ol style="list-style-type: none"> 1. Prozess starten. 2. Menü anzeigen <ol style="list-style-type: none"> Fall 1: Spiel fortsetzen Das Spiel wird fortgesetzt. Fall 2: Spiel beenden Das Spiel wird beendet und das Programm schließt sich. Fall 3: Neues Spiel starten.
Ergebnis	Das Pause-Menü wird aufgerufen und im Bildschirm angezeigt.
Plausibilitäten / Test Cases	Nach dem Pausieren sollte das Menü angezeigt.
Fehlerbehandlung	Im Fehlerfall soll das Spiel fortlaufen.
Folgeprozess	-

Tabelle 6: F2 - Pause-Menü aufrufen

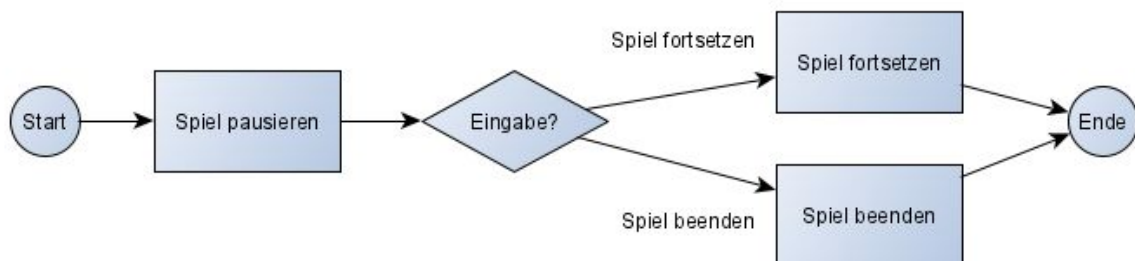


Abbildung 5: Flowchart F2 - Pause-Menü aufrufen

5.4 F3 - Charakter auswählen

Zweck / Ziel	Mit dieser Funktion kann der Spieler einen Charakter aus den Optionen auswählen.
Akteur / Auslöser	Spieler
Vorbedingung	F1 - Neues Spiel starten
Daten-Input	Charakter
Verarbeitungsschritte	<ol style="list-style-type: none"> 1. Beim Klick werden die Fähigkeiten des geklickten Charakters angezeigt. 2. Ausgewählter Charakter speichern.
Ergebnis	Der Charakter werden ausgewählt und für weitere Prozesse gespeichert.
Plausibilitäten / Test Cases	Es gibt mehr als ein Charakter mit verschiedenen Fähigkeiten.
Fehlerbehandlung	Im Fehlerfall wird der Spieler benachrichtigt.
Folgeprozess	F4 - Charakter bewegen

Tabelle 7: F3 - Charakter auswählen

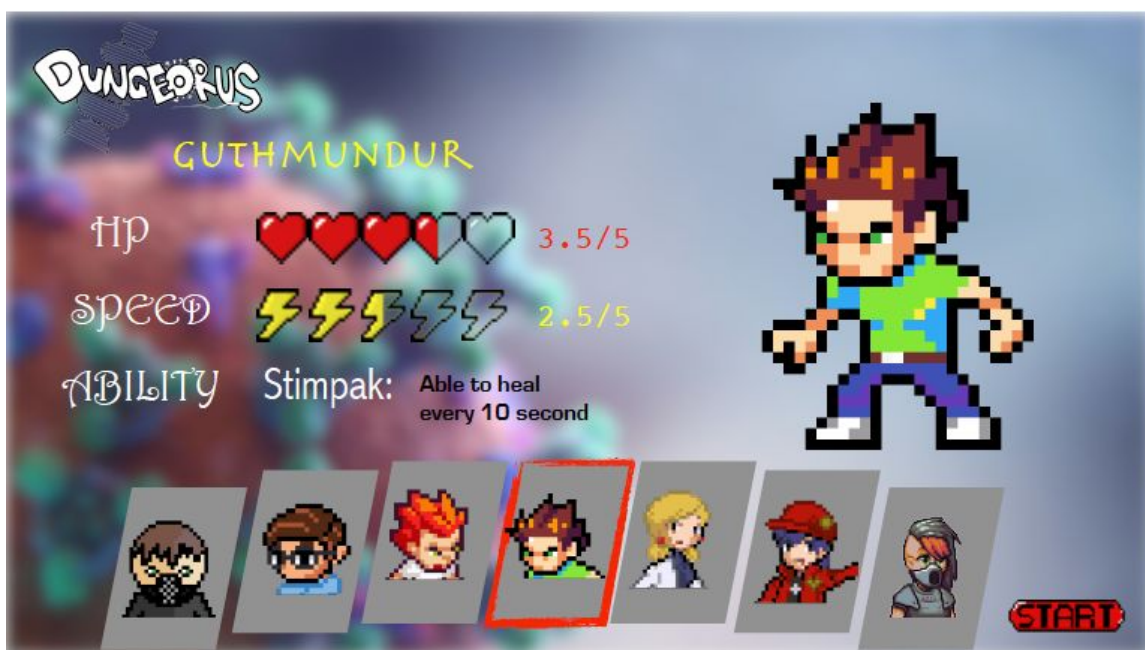


Abbildung 6: grobes Design bei Auswahl von Charakter

5.5 F4 - Charakter bewegen

Zweck / Ziel	Mit dieser Funktion kann der Spieler der Charakter in dem Spiel in die Mappe bewegen.
Akteur / Auslöser	Spieler
Vorbedingung	F3 - Charakter auswählen
Daten-Input	Eingabe von dem Spieler am virtuellen Analog-Joystick (Siehe: Abbildung 1: Skizze des Spiels)
Verarbeitungsschritte	<ol style="list-style-type: none">1. Prozess starten.2. Fall 1: Joystick nach oben bewegen Der Charakter bewegt sich nach oben. Fall 2: Joystick nach unten bewegen Der Charakter bewegt sich nach unten. Fall 3: Joystick nach rechts bewegen Der Charakter bewegt sich nach rechts. Fall 4: Joystick nach links bewegen Der Charakter bewegt sich nach links.
Ergebnis	Der Charakter bewegt sich nach der Eingabe.
Plausibilitäten / Test Cases	Der Charakter bewegt sich nach der Eingabe vom Spieler in die gewünschte Richtung.
Fehlerbehandlung	-
Folgeprozess	-

Tabelle 8: F4 - Charakter bewegen

5.6 F5 - Fähigkeiten einsetzen

Zweck / Ziel	Mit dieser Funktion kann der Spieler der Charakter in dem Spiel seine Fähigkeiten einsetzen.
Akteur / Auslöser	Spieler
Vorbedingung	F3 - Charakter auswählen
Daten-Input	Ausgerüstetes Item, Eingabe von dem Spieler
Verarbeitungsschritte	<ol style="list-style-type: none">1. Prozess starten.2. Spieler klickt A oder B-Button an.3. Die entsprechende Fähigkeit wird ausgeführt.
Ergebnis	Der Charakter setzt seine Fähigkeiten an.
Plausibilitäten / Test Cases	Nach der Eingabe vom Spieler soll die richtige Funktion aufgerufen werden und die richtige Animation abgespielt werden.
Fehlerbehandlung	-
Folgeprozess	-

Tabelle 9: F5 - Fähigkeiten einsetzen

5.7 F6 - Schlagen

Zweck / Ziel	Der Spieler kann mit seinem Charakter die Gegner schlagen.
Akteur / Auslöser	Spieler
Vorbedingung	F1 - Spiel starten
Daten-Input	Eingabe von dem Spieler (Button A)
Verarbeitungsschritte	<ol style="list-style-type: none">1. Prozess starten.2. Spieler klickt A (simpler Angriff).3. F6 - Schlagen aufrufen.
Ergebnis	Nach der Eingabe schlägt der Charakter.
Plausibilitäten / Test Cases	Nach der Eingabe vom Spieler soll die richtige Funktion aufgerufen werden und die richtige Animation abgespielt werden.
Fehlerbehandlung	-
Folgeprozess	-

Tabelle 10: F6 - Schlagen

5.8 F7 - Schießen

Zweck / Ziel	Mit dieser Funktion kann der Spieler die Gegner beschießen.
Akteur / Auslöser	Spieler
Vorbedingung	F3 - Charakter auswählen
Daten-Input	Eingabe von dem Spieler (Button B)
Verarbeitungsschritte	<ol style="list-style-type: none">1. Prozess starten.2. Spieler hat das entsprechende Item aufgehoben.3. Spieler klickt A (simpler Angriff).4. F7 - Schießen aufrufen.
Ergebnis	Nach der Eingabe schießt der Charakter.
Plausibilitäten / Test Cases	Der Charakter schießt nur, wenn das richtige Item aufgehoben wurde.
Fehlerbehandlung	-
Folgeprozess	-

Tabelle 11: F7 - Schießen

5.9 F8 - Charaktere / Items / Gegner / Levels designen

Zweck / Ziel	Diese Funktion hat das Ziel, die Charaktere, Items, Gegner und Levels zu malen und zu animieren.
Akteur / Auslöser	Admin
Vorbedingung	-
Daten-Input	-
Verarbeitungsschritte	<ol style="list-style-type: none"> 1. Verschiedene Charaktere zeichnen (mindestens 2) 2. Verschiedene Gegner zeichnen (mindestens 3 verschiedene Typen), die auf verschiedenen Viren und Bakterien basieren (wie CoVID-19, H2N2, usw.) 3. Verschiedene Items zeichnen (mindestens 2), die auf verschiedenen Hygieneartikeln sich basieren. (wie Händedesinfektionsmittel und Mundschutzmaske) 4. Verschiedene Levels (mindestens 2) 5. Die Bewegung der Charakteren animieren (für F4, F5, F6 und F7). 6. Die Bewegung der Gegnern animieren (und auch die Folge von F6 und F7). 7. Die Benutzung jeder Items animieren
Ergebnis	Das Design von verschiedenen Charakteren, Items und Gegnern, die im Spiel je nach der Eingabe von dem Spieler animiert werden können.
Plausibilitäten / Test Cases	<ul style="list-style-type: none"> - Alle Charaktere, Items, Gegner und Levels werden aus Top-Down-Ansicht gesehen. - Es sollte jeweils mehr als zwei verschiedene Charaktere, Items und Gegnertypen. - Jeder Charakter hat ein Bewegungsanimation. - Die Fähigkeiten der Charakteren haben verschiedene Animationen. - Items können aufgehoben werden. - Verschiedene Items erhöhen verschiedene Eigenschaften des Charakters.
Fehlerbehandlung	-
Folgeprozess	F9 - Charaktere / Items / Gegner / Levels anlegen

Tabelle 12: F8 - Charakter/Items/Gegner/Levels designen

5.10 F9 - Charaktere / Items / Gegner / Levels anlegen

Zweck / Ziel	Das Ziel dieser Funktion ist es, die Charaktere, Items, Gegner und Levels in das Spiel anzulegen und zu programmieren.
Akteur / Auslöser	Admin
Vorbedingung	F8 - Charaktere / Items / Gegner / Levels designen.
Daten-Input	Die Bilder (Animation) von jeden Charakter, Items und Gegner als Properties.
Verarbeitungsschritte	<ol style="list-style-type: none"> 1. In Unity werden alle Animation Properties (Bilder/PNG) und Assets aus jeder Charakter, Items und Gegner hochgeladen und als Tilemaps bezeichnet. 2. Die Levels werden in Szenen-Ansicht bearbeitet. 3. Nach dem Hochladen von Animation Properties werden die zu programmierenden Properties/Assets auf die Levels platziert. 4. Bei jedem Property/Asset wird ein Script erstellt und in Visual Studio 2019 geöffnet (beim Doppelklick von Script).
Ergebnis	Die gezeichneten Assets werden in das Spiel implementiert und können vom Spieler ausgewählt/genutzt werden.
Plausibilitäten / Test Cases	Die angelegten Assets werden vom Admin manuell auf ihre Funktionalität getestet.
Fehlerbehandlung	Assets prüfen und neu anlegen
Folgeprozess	<ul style="list-style-type: none"> - F3 - Charaktere bewegen - F5 - Fähigkeiten einsetzen - F6 - Schlagen - F7 - Schießen - F10 - Item aufsammeln - F11 - Charakter heilen

Tabelle 13: F9 - Charakter/Items/Gegner/Levels anlegen

5.11 F10 - Item aufsammeln

Zweck / Ziel	Mit dieser Funktion kann der Charakter Items von Gegnern oder Kisten einsammeln und benutzen.
Akteur / Auslöser	Spieler
Vorbedingung	F8 - Charaktere / Items / Gegner / Tilemaps designen und F9 - anlegen
Daten-Input	Items
Verarbeitungsschritte	<ol style="list-style-type: none">1. Prozess starten.2. Der Spieler klickt A neben den Items.3. Items wird aufsammelt. (Siehe F8 für die Items)
Ergebnis	Der Charakter wird ein erhöhtes Eigenschaft haben, und je nach Items kann F6 - Schlagen oder F7 - Schießen aufrufen.
Plausibilitäten / Test Cases	<ul style="list-style-type: none">- Die Items können aufsammelt werden.- Verschiedene Items können verschiedene Eigenschaften des Charakters erhöhen.- Wenn die Items kleinen Angriffsbereich hat (wie Händedesinfektionsmittel), wird nur F6 - Schlagen aufgerufen.- Wenn die Items großen Angriffsbereich hat (wie Hygiene Spray), wird nur F7 - Schießen aufgerufen.
Fehlerbehandlung	-
Folgeprozess	-

Tabelle 14: F10 - Items aufsammeln

5.12 F11 - Charakter heilen

Zweck / Ziel	Mit dieser Funktion kann der Charakter mit Items sich heilen.
Akteur / Auslöser	Spieler
Vorbedingung	F10 - Items aufsammeln
Daten-Input	Items, Eingabe von dem Spieler
Verarbeitungsschritte	<ol style="list-style-type: none">1. Prozess starten.2. Der Spieler klickt den Heilen-Button.3. Fall 1: Der Spieler hat keine Items eingesammelt, die die Eigenschaften zum Heilen haben. Es passiert nichts. Fall 2: Der Spieler hat gültige Items eingesammelt. Je nach Item werden die Lebenspunkte des Charakter erhöht. (Siehe F8 für die Items)
Ergebnis	Je nach Fälle wird entweder nichts passieren oder der Lebenspunkt des Charakters wird sich erhöht.
Plausibilitäten / Test Cases	Der Lebensbalken des Charakters erhöht sich.
Fehlerbehandlung	-
Folgeprozess	-

Tabelle 15: F11 - Charakter heilen

5.13 F12 - Gegner erzeugen

Zweck / Ziel	Mit dieser Funktion werden Gegner auf der Karte generiert.
Akteur / Auslöser	System
Vorbedingung	F1 - Neues Spiel starten
Daten-Input	kein
Verarbeitungsschritte	<ol style="list-style-type: none">1. Prozess starten.2. Level wird geladen und dabei werden die Gegner erzeugt.
Ergebnis	Es befinden sich Gegner auf der Karte.
Plausibilitäten / Test Cases	Die Karte darf nicht leer sein.
Fehlerbehandlung	Der Spieler soll benachrichtigt werden und wird dazu aufgefordert ein neues Spiel zu starten.
Folgeprozess	-

Tabelle 16: F12 - Gegner generieren

5.14 F13 - Highscore berechnen

Zweck / Ziel	Mit dieser Funktion wird der Highscore des Benutzers berechnet..
Akteur / Auslöser	System
Vorbedingung	F1 - Neues Spiel starten
Daten-Input	Anzahl der besiegtten Gegner, Überlebensdauer
Verarbeitungsschritte	<ol style="list-style-type: none">1. Prozess starten.2. Gegner werden besiegt.3. Pro besiegtten Gegner gibt es 10 Punkte sowie für jede zehn Sekunden, die der Spieler überlebt.4. Score wird angezeigt.
Ergebnis	Der Highscore wird dem Benutzer angezeigt.
Plausibilitäten / Test Cases	Nach dem Besiegen von Gegnern darf der Score nicht 0 sein.
Fehlerbehandlung	Der Spieler soll benachrichtigt werden und wird dazu aufgefordert ein neues Spiel zu starten.
Folgeprozess	-

Tabelle 17: F13 - Highscore berechnen

6 Backlog

Funktionalitäten	Sprint 1	Sprint 2	Sprint 3
F1 - Neues Spiel starten	x		
F2 - Pause-Menü aufrufen			x
F3 - Charakter auswählen		x	
F4 - Charakter bewegen	x		
F5 - Fähigkeit einsetzen		x	
F6 - Schlagen		x	
F7 - Schießen		x	
F8 - Charaktere / Items / Gegner / Levels designen	x	x	x
F9 - Charaktere / Items / Gegner / Levels anlegen		x	
F10 - Item aufsammeln		x	
F11 - Charakter heilen			x
F12 - Gegner generieren			x
F13 - Highscore berechnen			x

Tabelle 18: Backlog

7 Aufteilung der Kapitel

Autor	Aufgabe/Kapitel
Adib Ghassani Waluya	3, 5.1, 5.3 5.4, 5.5, 5.7, 5.8, 5.10, 5.12, 5.13, 6
Michael Reno	4, 5.1, 5.2, 5.3, 5.4, 5.5, 5.6, 5.9, 5.11, 5.14, 7
Minh Hoang Do	1, 2, 3, 5.1, 5.2, 5.6, 5.8, 5.10, 5.11, 5.13, 6