

Prototyping interaktiver Medien-Apps und Games

Game: Jump and fight

Student: David Niemann

Professor: Prof. Jirka Dell'Oro-Friedl

Studiengang: Medieninformatik

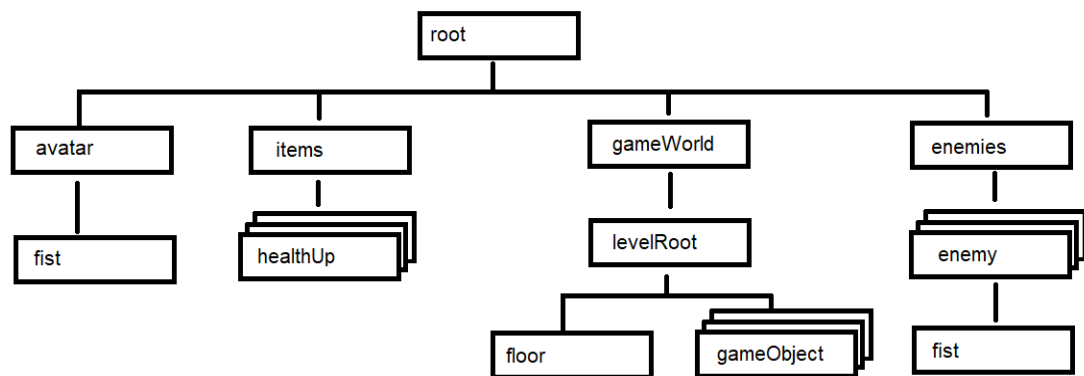
Matrikelnummer: 263468



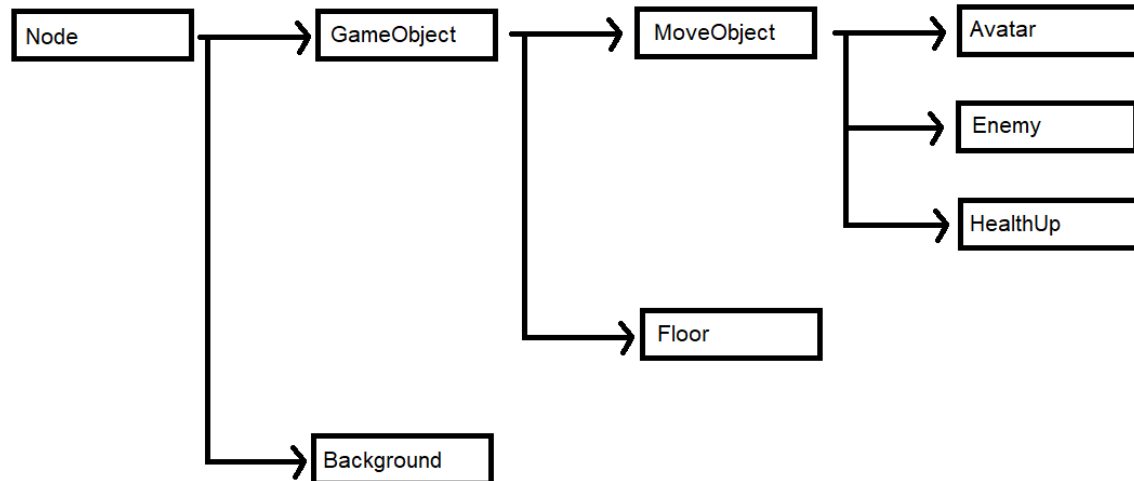
Nr.	Bezeichnung	Inhalt
	Titel	Jump and fight
	Name	Niemann David
	Matrikelnummer	263468
1	Nutzerinteraktion	A & D: Laufen Space: Sprung Linke Maustaste: Angriff Buttons: Pause, Starten & Restart
2	Objektinteraktion	Avatar, Enemies und Items(HealthUp) kollidieren mit Boden und Wänden, um auf ihnen zu Laufen oder den Spielbereich einzuschränken Avatar und Enemies haben Kinderobjekte, die kollidieren erkennen, um dich gegenseitig Schaden zuzufügen Items (HealthUp) können durch eine Kollision von dem Avatar aufgenommen werden
3	Objektanzahl variabel	Die Spielwelt mit Gegnern und Items wird zufällig und mit variabler Anzahl an Böden, Gegnern und Items während der Laufzeit generiert.
4	Szenenhierarchie	Die Wertobjekte, Gegner und Items haben einen eigenen Knoten, um die Kollision leicht abfragen zu können Eine Note des Sprites zeigt an den jeweiligen Enemies oder Avatar, um des Sprites in Laufrichtung zu flippen, zu ermöglichen Avatar und Enemies haben eine „Faust“ untergeordnet,
5	Sound	Hintergrund Musik bei Angriffen und Treffern Sounds ausgelöst Laufen springen und landen des Avatars hat jeweils einen Sound
6	GUI	Spiel kann pausiert, gestartet und Neugestartet werden
7	Externe Daten	Weltgröße, Gegner und Avatar Stärke, Leben
8	Verhaltensklassen	GameObjekt allgemeine Eigenschaften MoveObjecte Haben eine Bewegungsrichtung Enemy & Avatar Haben erweiterte Eigenschaften zu MoveObject
9	Subklassen	Es gibt viele Klassen, die alle auf der Node aufbauen
10	Maße & Positionen	Die Spielewelt ist in Units aufgabelt was 2 Welteinheiten widerspiegelt: Avatar 2*2 Units

		Enemy 1*2 Units HealthUp 1*1 Units Platformer x*1 Units
11	Event-System	Die Steuerung des Avatars wird durch ein Event ausgelöst Zeitevent bestimmen wie lange die Animation und die HitBoxen der Angriffe aktiv sind

Szenenhierarchie



Subklassen



Spiele Konzept:

Ein Jump and Run mit Fighting Elementen im Acade-Style.

Das Spiel dauert so lange bis der Spieler es beendet oder die Leben des Avatars auf/unter 0 fallen.

Das Ziel des Spiels ist so viele Punkte wie möglich zu sammeln.

Der Spieler bekommt für jeden besiegten Gegner einen Punkt.

Eine zufällige Weltgenerierung.

Spiel:

Das Spiel kann mit einem Klick auf den Start-Button gestartet werden.

Der Avatar ist ein Ritter und der erste Gegner mit den Sprites eines Skeletes mit einer Axt, starten mit 3 Leben.

Mit dem Drücken der linken Maustaste löst der Spieler eine Angriff-Animation aus, mit der der Gegner verletzt werden kann, diese Aktion kann nicht in der Luft ausgeführt werden. Mithilfe der Tasten A und D sich zum Gegner hinbewegen und seinen Schlägen ausweichen. Der Avatar des Spielers ist es nicht erlaubt das Spielfeld zu verlassen bis der Gegner besiegt ist, solange ist die Kamera auf das Spielfeld gerichtet. Sobald der Gegner besiegt wird, kann der Spieler nach rechts laufen folgt und die Kamera folgt dem Avatar. Sobald der Spieler das neue Level betritt, ist er gezwungen den neuen Gegner zu besiegen.

Beim Betreten eines Levels werden zufällig 0 bis 2 Leben für den Avatar gespart die eingesammelt werden können.

SpielWelt:

Die Spielwelt besteht aus zufalls-generierten Levels. Das Level besteht aus einem Boden, unsichtbaren Wänden und unsichtbarem Dach. Beim Erstellen der Welt werden 1 bis 4 schwebende Plattformen generiert. Diese Plattformen haben eine zufällige Länge von einer Unit bis zur Hälfte der Level Länge. Diese Plattformen werden mit einem maximalen Abstand von einer Unit, sodass der Avatar jede Plattform erreichen kann. Wenn kein Enemy mehr vorhanden ist, wird das nächste Level generiert ohne linke Wand und die rechte Wand des alten Levels wird entfernt, was dem Spieler ermöglicht ins neue Level zu gelangen. Betritt der Avatar nun die Mitte des neuen Levels wird die linke Wand hinzugefügt, sodass der Avatar das Gebiet nicht mehr verlassen kann. Bei diesem Vorgang werden an der Note Items alle Items entfernt und 0 bis 2 neue Items im aktuellen Level erschaffen.

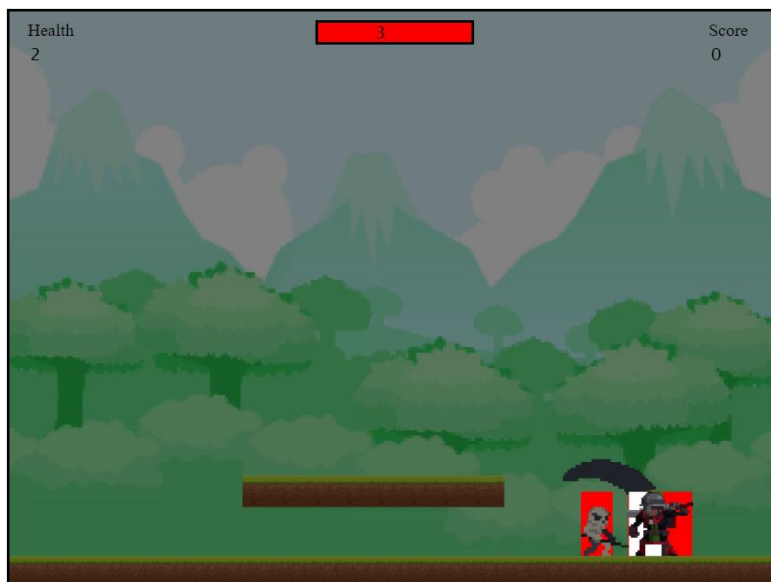
Hitboxen:



Rot: Hitbox des Gegners und Avatars wo die getroffen werden können.

Weiß: HitBox des Schwertes und Axt mit dem der Treffer festgestellt wird.

Es wird eine Kollision des Schwertes und Axt Hitbox mit den jeweiligen Geigerischen Hitbox abgefragt, solange das dafür zuständige MoveObject den Status grounded auf false gesetzt hat.



Enemies:

Die Gegner handeln nach einem einfachen Prinzip. Er wartet bis den Avatar in die Mitte des aktuellen Levels angekommen. Wenn der Avatar sich auf der gleichen Höhe befindet, läuft er auf diesen zu, in anderen Fällen bleibt er auf seiner aktuellen Position stehen. Sobald der Avatar in die Reichweite von einer Unit sich dem Gegner nähert, greift er diesen an.

Der erste Gegner startet, mit 3 Leben und einem Schaden. Das Leben wird pro Level um eins erhöht und der Schaden jede fünf Level um eins erhöht.

Quellen:

Sprites:

Hintergrund und Boden: <https://www.gameart2d.com/free-platformer-game-tileset.html>

Avatar: https://www.sprisers-resource.com/pc_computer/fullmetalfuries/sheet/144070/

Enemy: <https://jesse-m.itch.io/skeleton-pack>

Sounds:

audioEnemyHit: <https://www.fesliyanstudios.com/play-mp3/1135>

audioAvatarHit & audioSword: <https://mixkit.co/free-sound-effects/sword/>

audioJump & audioLand: <https://mixkit.co/free-sound-effects/metal/>

audioStep: <https://mixkit.co/free-sound-effects/footsteps/>

audioCollect & audioBackground: <https://www.beebox.co/>