NyanCat TD

Inhaltsverzeichnis

1. Einleitung 1
2. Features 2
2.1 Ideen 2
2.1.1 Türme 2
2.1.2 Level 2
2.1.3 Minions 2
2.1.4 Sonstiges 2
3. Spiel- und Levelkonzept 3
3.1 Spielkonzept 3
3.2 Ziel des Spiels 3
3.3 Spielregeln 3
3.4 Levelkonzept 4
4. Architektur 5
4.1 Model 5
4.1.1 Überblick über ihr Model 5
4.1.2 Aus dem Spielkonzept abgeleitete Entities 6
4.1.3 Erläuterung des konzeptionellen Modells und der Entities 6
4.1.4 Beschreibung des Zustands der konzeptionellen Modelentities 9
4.1.5 Beschreibung des dynamischen Verhaltens des Models 13
4.1.6 Erläuterung wie Spiellevel definiert werden 13
4.2 View 14
4.2.1 Überblick über ihren View 14
4.2.2 Erläuterung des statischen HTML Dokuments zur Darstellung des Spielfelds 15
4.2.3 Erläuterung der Wirkungsweise des View-Updates 15
4.2.4 Gestaltung des Spiels mittels CSS 16
4.3 Controller 16
4.3.1 Überblick über ihren Controller 16
4.3.2 Erläuterung der Interaktionsmöglichkeiten des Nutzers 17
4.3.3 Erläuterung des Timers 18
4.3.4 Abbildung der Nutzerinteraktionen auf Modelinteraktionen 20
4.3.5 Abbildung der Timer-Events auf Modelinteraktionen 20
5. Zusammenfassung und Ausblick 22
5.1 Zusammenfassung 22
5.2 Ausblick 22

1. Einleitung

Dieses Projekt wurde von Florian Winzek und René Kremer erarbeitet.

Die Verantwortlichkeiten lagen dabei im Bereich der Tower Entitäten bei Florian Winzek und der Level / Minion Entitäten bei René Kremer.

Der Controller und View wurde zusammen entwickelt.

Wir stimmen der Veröffentlichung des Spiels zu, jedoch nur unter der Voraussetzung, dass nach der Klausurenphase (ab dem 23.07.2015) noch von uns die benutzen Bilder durch Lizenz konforme (CC Lizenz) Bilder ausgetauscht werden dürfen. Dafür werden die Bilder ausgetauscht, sowie die Minion Namen in XML Level Datei und CSS Datei geändert. Das restliche Programm ändert sich nicht.

Sollte dies nicht gegeben werden können, sehen wir aus Gründen der benutzen Bilder von der Veröffentlichung ab.

So long and thanks for all the fish  !

2. Features

Hier werden ein paar Features dargestellt.

2.1 Ideen

2.1.1 Türme

- verschiedene Türme
- Angriffsgeschwindigkeit je Turm
- Upgradefunktion der Türme
- Berechnung des Schadens
- Schadensklasse
- Fähigkeiten von Türmen

2.1.2 Level

- verschiedene Wellen je Level
- mehrere Level
- Highscore
- Punkte pro Wave + pro Minion + Punkte je Gold
- verschiedene Schwierigkeitsgrade
- Jedes Level hat eine Anzahl von Leben für den Spieler
- Level verloren, wenn Leben auf 0

2.1.3 Minions

- Laufgeschwindigkeit je Minion
- verschiedene Minions
- Bosse
- Rüstungsklassen je Minion

2.1.4 Sonstiges

- Baumenü als Sidebar
- Stop Button, um das Spiel zu pausieren

3. Spiel- und Levelkonzept

3.1 Spielkonzept

Bei einer Tower Defense geht es darum, dass auf einem Pfad Minions laufen und diese das Ende nicht erreichen dürfen.

Auf dem Weg dorthin hat ein Spieler die Möglichkeit, Türme zu bauen, um so die Minions aufzuhalten.

Die Türme schießen dabei auf die Minions bis diese sterben.

Sollte ein Minion das Ende des Pfads erreichen, so verliert der Spieler ein Leben.

Sind die Leben des Spielers auf 0, so verliert er.

Jedes Level hat mehrere Waves, in denen Minions über den Pfad laufen. Jede Wave wird mit einem Countdown angekündigt.

In dieser Zeit kann der Spieler schon Türme bauen.

Der Spieler erhält für jeden getöteten Minion Gold, womit er neue Türme bauen oder upgraden kann, sowie Punkte für seine persönliche Highscore.

3.2 Ziel des Spiels

Das Ziel des Spiels ist es die Minions am Durchlaufen des Pfads zu hindern und möglichst viele Punkte für getötete Monster zu sammeln.

Auf diesem Weg dort hin, muss der Spieler versuchen möglichst viele Minions zu töten, da er ansonsten für jeden nicht getöteten Minion ein Leben verliert.

Sind die Leben auf 0, so verliert er das Spiel. Das Spiel wird gewonnen, wenn das letzte Level geschafft wurde.

3.3 Spielregeln

Das Spiel wird auf dem Spielbrett gespielt und es können keine Türme auf dem Pfad gebaut werden, um die Minions zu blockieren.

Es dürfen Türme nur dann gekauft werden, wenn ein freies Feld ausgewählt wurde und genügend Gold für den Kauf oder das Upgrade vorhanden ist.

3.4 Levelkonzept

Das Spiel hat 3 Schwierigkeitsgrade: easy, medium, hard.

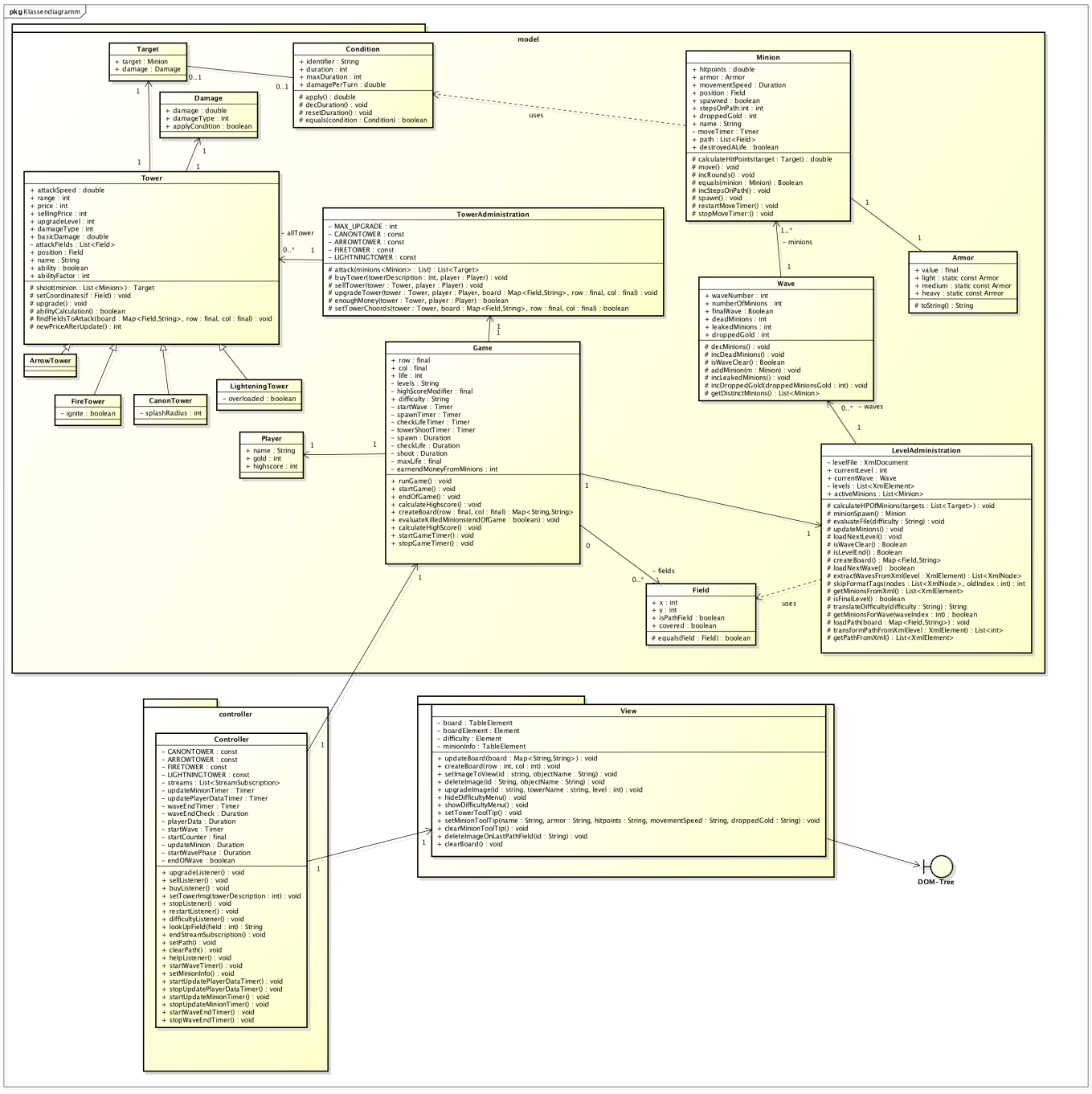
Jeder Schwierigkeitsgrad hat 3 Level mit jeweils 5 Waves. Jede Wave hat mehrere Minions, die von einer oder mehreren Arten sein können.

Am Ende jedes Levels ist ein Boss, ein sehr starker Minion.

Der Pfad des Levels wird zufällig aus der XML Datei geladen.

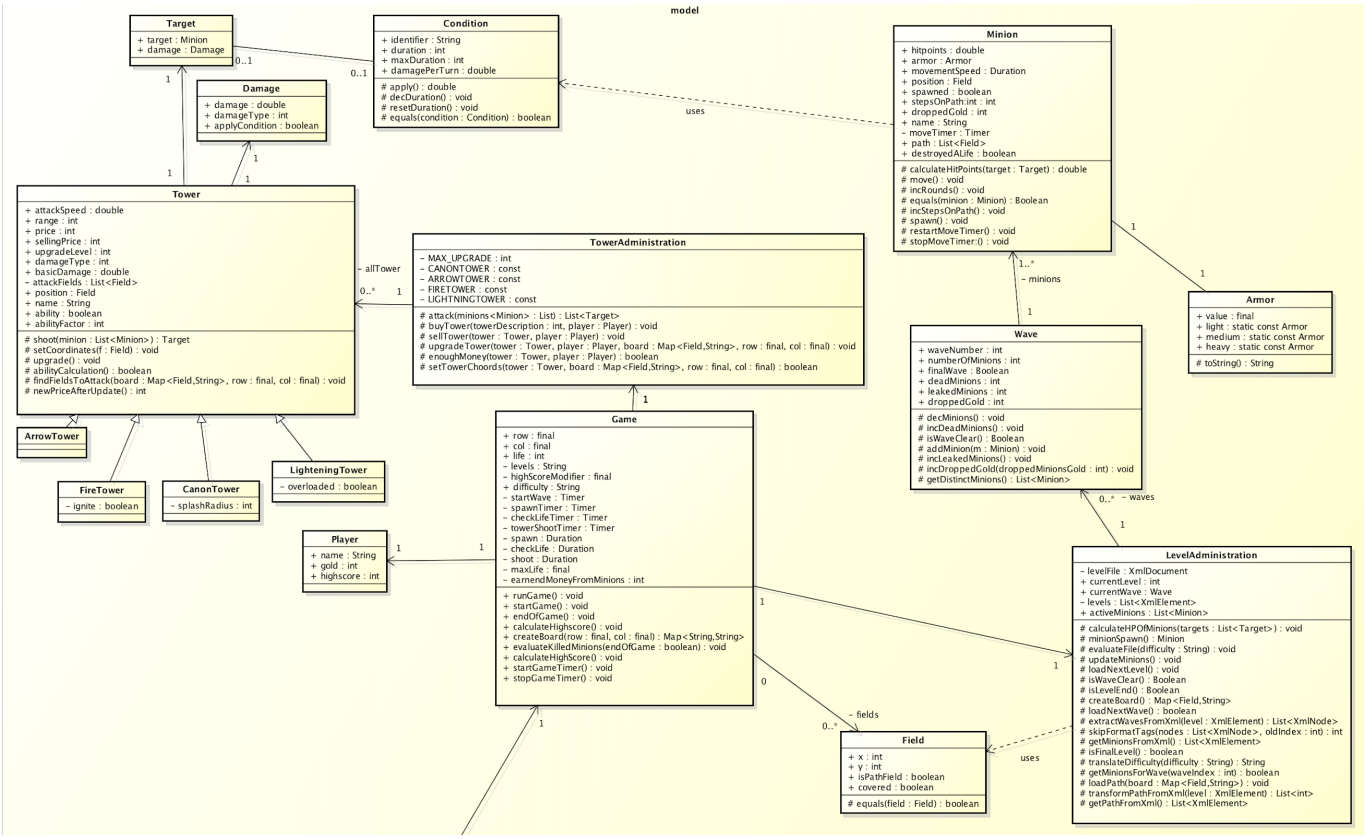
Die Level können durch ein XML Schema selbst definiert werden.

4. Architektur



4.1 Model

4.1.1 Überblick über ihr Model



4.1.2 Aus dem Spielkonzept abgeleitete Entities

Laut dem Spielkonzept gibts es folgende Entitäten im Spiel:

- Tower
- Level
- Wave
- Minion
- Field
- Target

Aus den Features wurden weiterhin abgeleitet:

- Condition
- Armor
- Damage
- Arrow Tower
- Canon Tower
- Fire Tower
- Lightning Tower

4.1.3 Erläuterung des konzeptionellen Modells und der Entities

Game

Das Game hat einen Player, ein Spielbrett in Form von mehreren Fields, eine Tower Administration und eine Level Administration.

Der Spieler verwaltet sein Gold und seine Punkte, erhält neue Punkte und Gold jedoch vom Spiel zugewiesen.

Das Spielbrett wird anfangs vom Spiel erzeugt anhand der Länge und Breite des Bretts.

Die Level Administration wird genutzt, um die Level zu laden, deren Waves und Minions sowie die Pfade der Fields, auf denen die Minions sich bewegen.

Tower Administration

Die Tower Administration beinhaltet eine Liste aller Türme, die vom Spieler gekauft wurden. Sie verwaltet die Platzierung bzw. Löschung, sowie die Angriffe der einzelnen Tower.

Tower

Die Klasse Tower stellt das Grundgerüst aller im Spiel vorhanden Türme dar. Sie definiert die Struktur und die Eigenschaften, sowie das Verhalten des Turms beim Angriff.

Ein Turm hat einen (Ver-)Kauf-/Upgradepreis, ein Angriffsziel, einen Schadenswert und eine spezielle Fähigkeit(optional). In dieser Variante von Tower Defense, wird es 4 Arten von Türmen geben, die sich in den genannten Werten unterscheiden werden.

Damage

Die Klasse Damage ist eine Hilfsklasse, um einen Schadenswert eines Towers zu berechnen. Ein Damageobjekt setzt sich zusammen aus:

- Grundschadenswert
- Schadenstyp(Belagerung, Stichschaden,Feuer,Blitz)
- Zustand(spezielle Fähigkeit, welche unter einer zufälligen Warscheinlichkeit eintritt)

Condition

Eine Condition ist, wie oben beschrieben, eine extra Fähigkeit, die ein Tower haben kann, um noch mehr Schaden an den Minions anrichten zu können. Diese ist wie folgt strukturiert:

- Identifier bzw. einen Namen der Fähigkeit
- eine Dauer oder auch Abklingzeit
- den extra Schadenswert

Target

Die Klasse Target ist eine weitere Hilfsklasse, um das Ziel(Minion) mit dem vom Turm ausgehenden Schaden, sowie der Condition zu bündeln. Dieses erstellte Objekt wird dann von der LevelAdministration benutzt, um den genauen Schadenswert zu diesem Minionobjekt berechnen zu

können.

Level Administration

Die Level Administration lädt anhand einer XML Datei die Level ein, die Waves, die Minions und einen Pfad für die Minions.

Außerdem kann die Level Administration prüfen, ob die Wave vorbei ist, das Level oder ob es sich um das letzte Level handelte.

Die Level Administration übergibt auch die Treffer von den Türmen an die Minions.

Die Level Administration besteht aus dem XML Dokument des Levels, einer Nummer des aktuellen Levels, dem aktuellen Wave Objekt, einer Liste von Levels und den aktiven Minions

Wave

Die Wave ist ein Teil eines Levels. Sie enthält die gesamte Anzahl an Minions und zählt das fallen gelassene Gold der getöteten Minions.

Außerdem kann sie die verschiedenen Miniontypen zurück geben und getrennt getötete sowie durchgelassene Minions voneinander und zählt diese.

Minion

Die Minions bewegen sich auf dem Spielbrett und aktualisieren ihre Hitpoints.

Ein Minion hat

- eine Anzahl von Hitpoints
- eine Rüstungsklasse
- eine Bewegungsgeschwindigkeit
- eine Position auf dem Brett
- einen Status, ob er schon aufgerufen wurde und aktiv auf dem Brett ist
- einen Zähler seiner Schritte auf dem Pfad
- einen Wert an Gold, den er fallen lässt
- einen Namen
- einen Timer für seine Bewegung
- den Pfad
- einen Status, ob er ein Leben zerstört hat oder nicht, also ob er das Ende des Pfads erreicht hat.

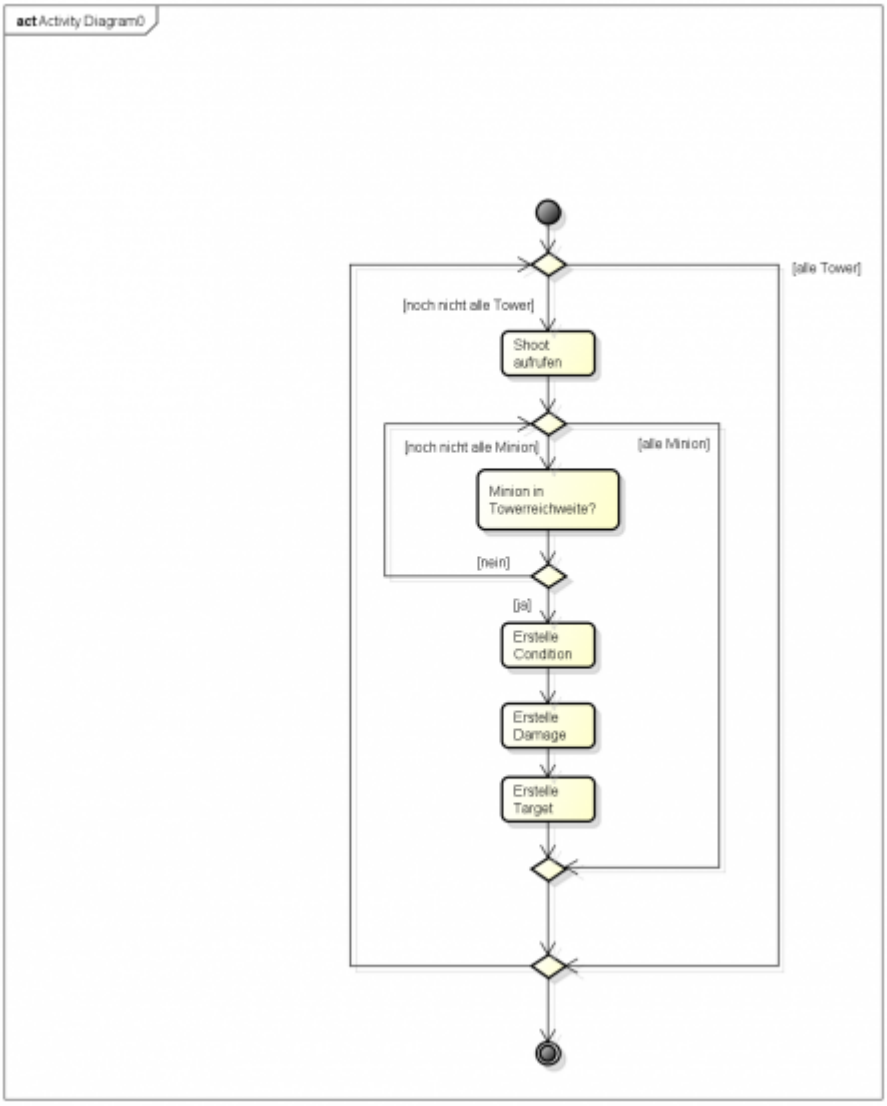
Armor

Armor ist eine Hilfsklasse, die den Wert Light, Medium oder Heavy annehmen kann und damit die Rüstungsklasse eines Minions definiert.

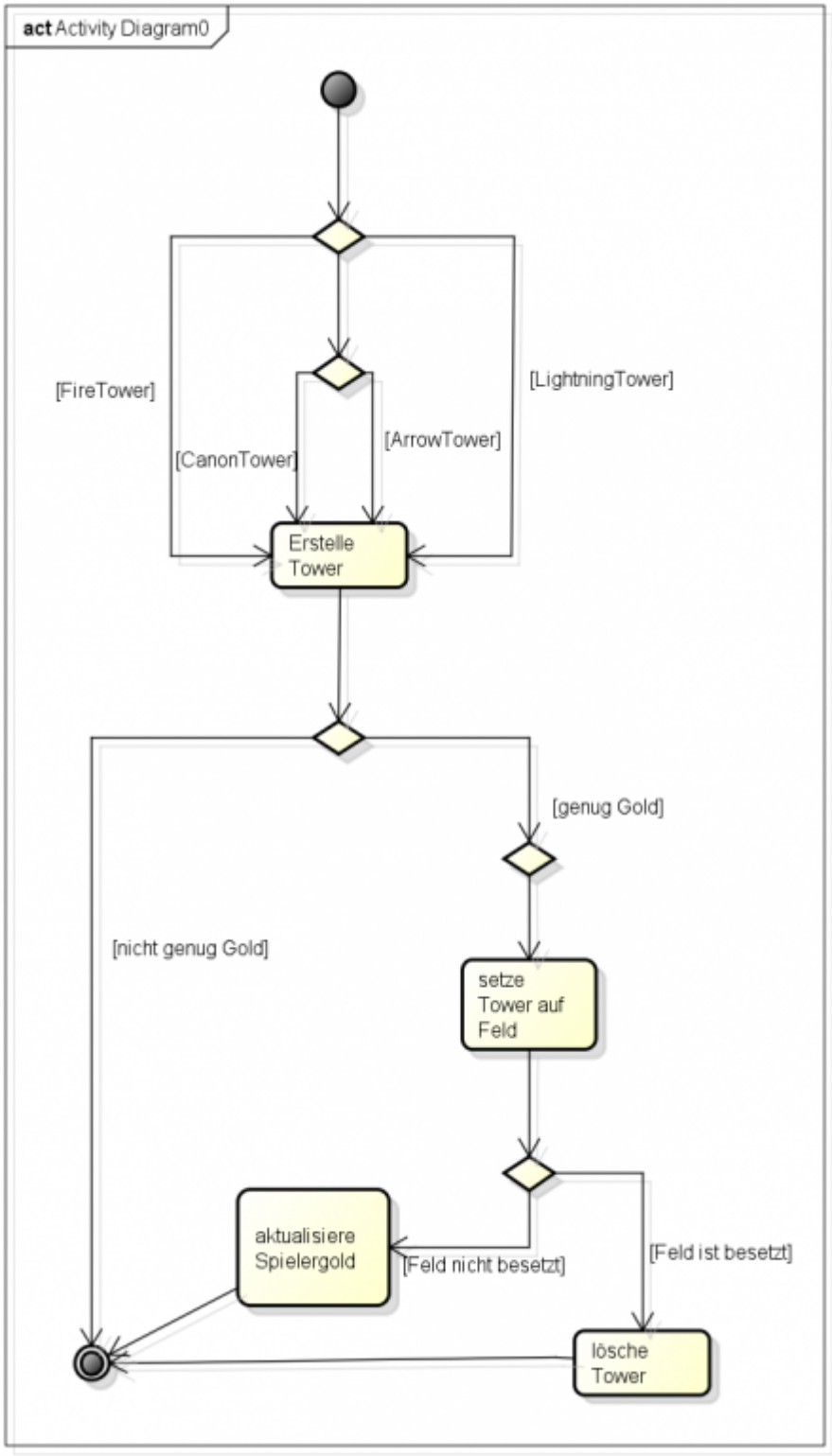
4.1.4 Beschreibung des Zustands der konzeptionellen Modelentities

Tower Administration

Attack



Buy



Sell

Löscht den Turm aus der Liste und aktualisiert das Gold des Spielers.

Upgrade

Erhöht die Fähigkeiten des Towers um einen Faktor und aktualisiert das Gold des Spielers, sofern

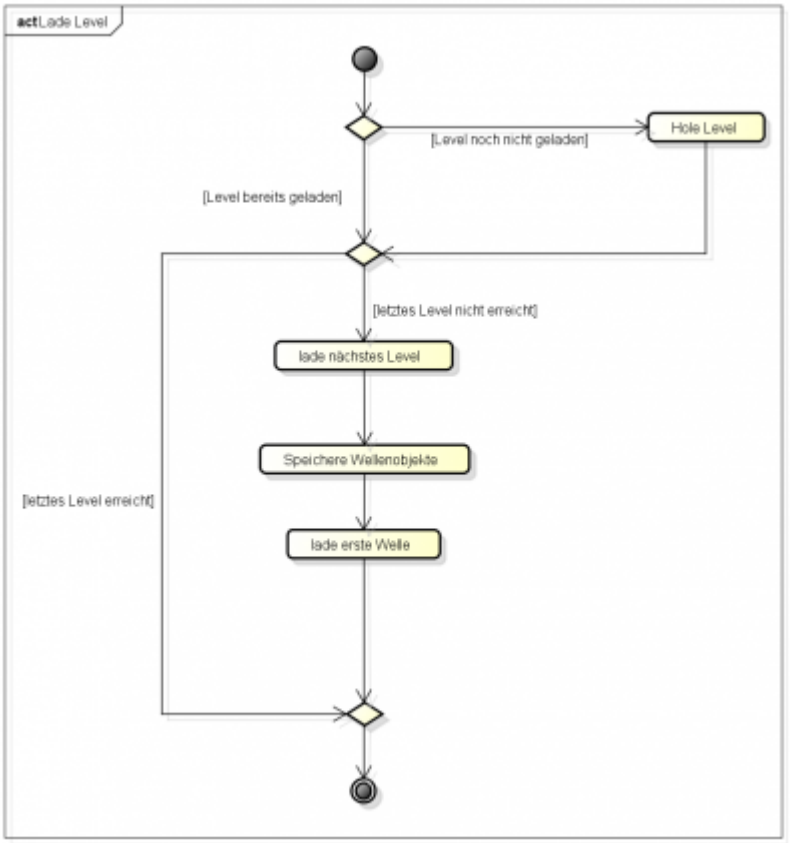
genügend Gold vorhanden ist

SetTowerChoords

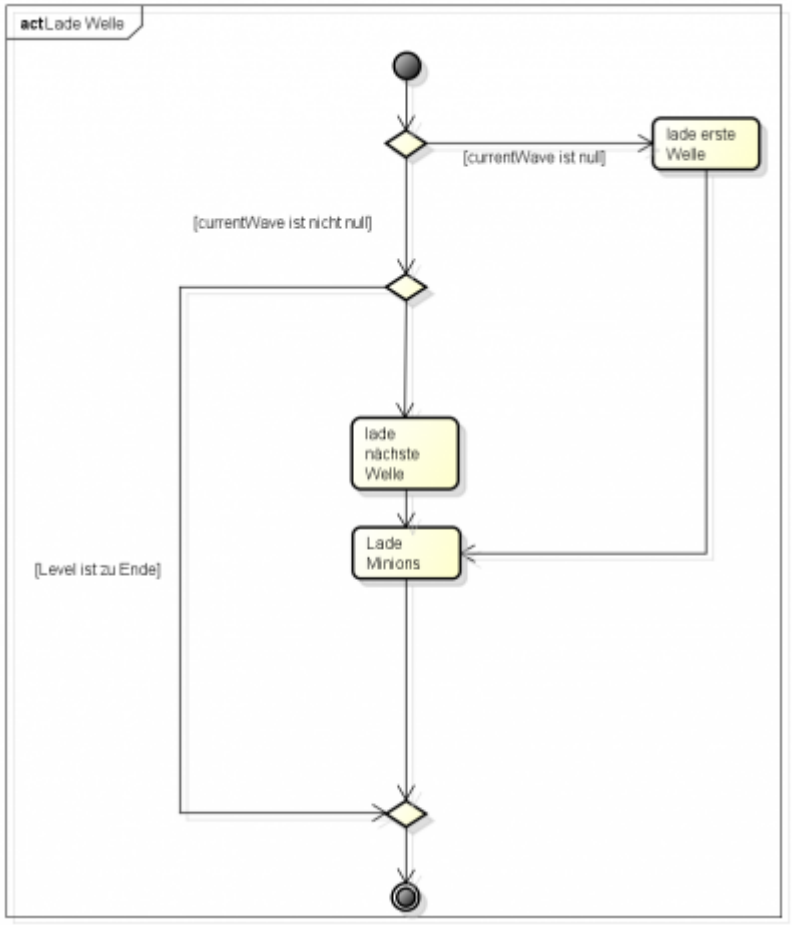
Setzt den Tower auf die vom Spieler vorgegebenen Choordinaten, sofern das Feld kein Pfad oder schon von einem anderen Turm belegt ist

Level Administration

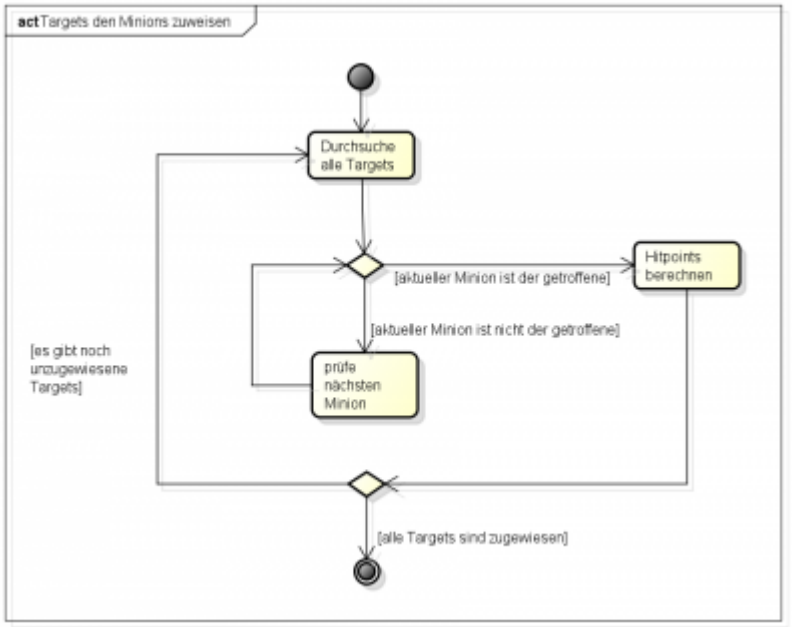
loadNextLevel



loadNextWave



calculateHitPoints



Die Minions selber berechnen nun ihre aktuellen Hitpoints und ziehen sich dabei anhand des übertragenen Basis Schadens und des Damage Typs gegen ihre Rüstungsklasse die Hitpoints ab, die sie verloren haben.

minionSpawn

Die Level Administration geht durch die Liste der Minions der Wave und lässt den nächsten das Feld betreten, der es noch nicht betreten hat.

Dabei setzt die Administration ihm einen Pfad, die Startposition und fügt ihn seiner activeMinion Liste hinzu.

Wenn der Minion startet, aktiviert er seinen Timer, der ihn automatisch nach Ablauf des Intervalls ein Feld weiter bewegt, bis er das Ende des Bretts erreicht hat.

4.1.5 Beschreibung des dynamischen Verhaltens des Models

SpawnTimer

Der SpawnTimer wird beim Start des Spiels gestartet und ruft periodisch die Level Administration auf um neue Minions zu spawnen

TowerShootTimer

Der TowerShootTimer wird beim Start des Spiels gestartet und ruft periodisch die Tower Administration auf.

Diese gibt eine Liste von Targets zurück, die dann an die Level Administration übergeben wird, damit die Targest zu gewiesen werden können und die Minions ihre Hitpoints aktualisieren.

CheckLifeTimer

Der CheckLifeTimer wird beim Start des Spiels gestartet und prüft periodisch, ob ein Minion das Ende des Pfads erreicht hat, setzt den Wert beim Minion, dass er ein Leben zerstört hat und zieht dem Spieler ein Leben ab.

MoveTimer

Die Minions haben eigene MoveTimer, die beim Aufruf von spawn aktiviert werden und periodisch die Minions auf dem Pfad weiter bewegen.

Dieser Timer endet, wenn der Minion den Pfad komplett abgelaufen hat.

4.1.6 Erläuterung wie Spiellevel definiert werden

Spiel Level

Level werden durch Wellen definiert und Wellen haben eine bestimmte Anzahl von Minions.

In unserem Level XML Dokument gibt es für jeden Schwierigkeitsgrad jeweils 3 Level mit jeweils 5 Wellen.

Die Wellen haben unterschiedliche Anzahlen von Minions und in der 5. Welle jedes Levels wartet ein Boss Minion.

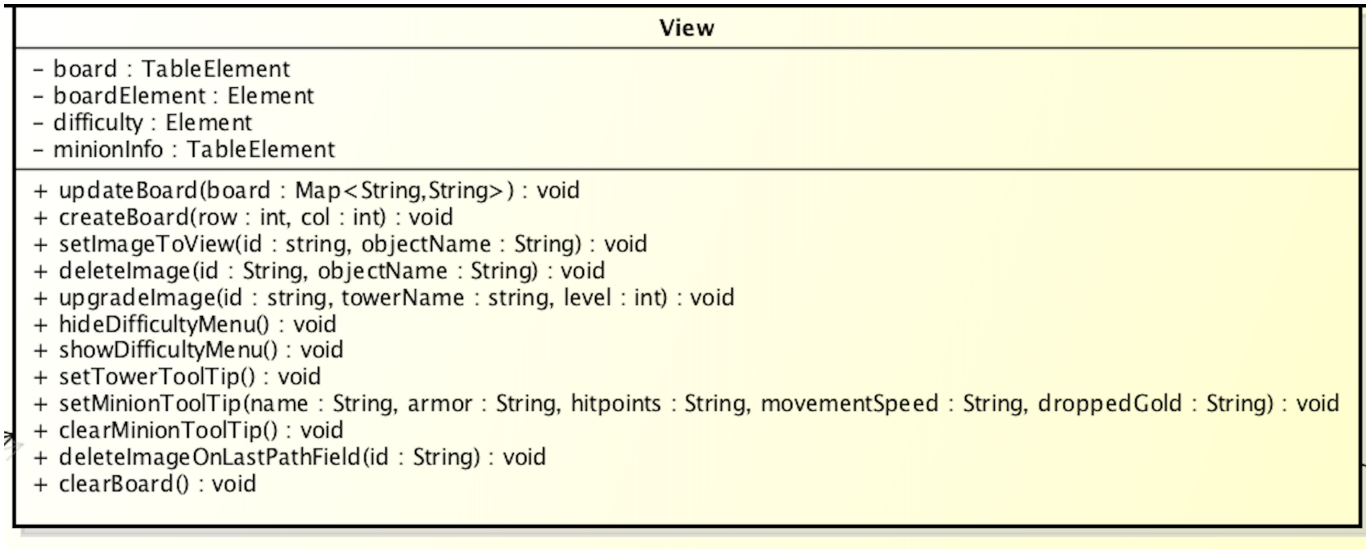
Nach der 5. Welle im 3. Level endet das Spiel und der Spieler hat (hoffentlich) gewonnen.

Das XML Dokument hat folgende Struktur (verkürzte Form zur Darstellung):

```
<allLevels>
   <easyLevels>
    <level no="1" finalLevel="false">
            <wave no="1" finalWave="false">
                <minion ref="weakKratzke" numberOfMinions="10"></minion>
            </wave>
    </level>
   </easyLevels>
   <mediumLevels/>
   <hardLevels/>
   <minions>
    <minion name="weakKratzke">
        <hitpoints>25</hitpoints>
        <armor>light</armor>
        <movementSpeed>1000</movementSpeed>
        <droppedGold>25</droppedGold>
    </minion>
   </minions>
   <paths>
    <path no="1">
        <pathfield>
            <y-coord>0</y-coord>
            <x-coord>5</x-coord>
        </pathfield>
    </path>
    <path no="2">
    </path>
   </paths>
 </allLevels>
```

4.2 View

4.2.1 Überblick über ihren View



4.2.2 Erläuterung des statischen HTML Dokuments zur Darstellung des Spielfelds

Das HTML Dokument besteht in der Grundstruktur aus 3 wichtigen Elementen. In der Mitte gibt es zum einen das Spielbrett, links und rechts davon ist jeweils eine Navigationsleiste angebracht. In der linken Menuleiste sind die Spielerdaten aufgelistet(Name, Punkte, Gold,Leben). Den Namen kann der Spieler am Anfang über einen Input Button eingeben.

Darunter befindet sich das Interaktionsmenü. Zuerst erscheinen hier drei Buttons, die die Schwierigkeit des Spiels bestimmen(Easy, Medium, Hard). Sobald das Spiel mit der gewünschten Schwierigkeit gestartet wurde, läuft an der oberen Kante der Leiste ein Timer herunter, der die Minions spawnen lässt. Während des Spiel gibt es einen Kauf-/Sell-/Upgrade Button, mit denen der Spieler das Spiel spielen kann.

Auf der rechten Seite befindet sich ein Informationmenü. Über 3 Buttons(Game Help, Tower Help, Armor Help) kann der Spieler sich die ihm fehlenden Informationen beschaffen. Sobald das Spiel gestartet wurde, kann der Spieler rechts über einen Stop-Button das Spiel pausieren und über einen weiteren Button wieder starten. Unten in der Leiste befinden sich noch Informationen zu den aktuellen Minions.

4.2.3 Erläuterung der Wirkungsweise des View-Updates

In diesem Spiel gibt es zwei große Wirkungsbereiche des View-Updates.

Einmal die Buttons, mit denen der Spieler interagiert und einmal der sogenannte „UpdateMinionTimer“, welcher dafür sorgt, dass die Hintergrundbilder der Minions auf dem Spielfeld bewegt bzw. gelöscht werden.

Die Buttons, die der Spieler aufruft, verändern die Oberfläche bzw. das Menü selber. Z.B.: Bei dem „Buy“-Aufruf werden Sell-/Upgradebutton eingeklappt(techn.: <button>.hidden = true), dies geschieht über den Controller, welcher die entsprechende Methode im View aufruft. Der Spieler hat dann nur noch die Chance das Spielfeld anzuklicken, um das jeweilige Towerimage zu setzen. Danach klappt das Menü wieder vollständig auf. Ähnlich funktioniert dieses bei den beiden anderen Buttons.

Das Hilfemenü arbeitet äquivalent. Durch Drücken auf die Buttons klappt ein Text auf, welche durch

ein erneutes Klicken wieder verschwindet.

Das Bewegen der Minionimages wird durch einen Timer aus dem Controller gesteuert. Hierzu wird beim Verlassen des Feldes, das Image gelöscht(Controller ruft dazu die „deleteImage“ - Methode auf) und der Standardhintergrund geladen. Beim Betreten des neuen Feldes wird über die schon angesprochenen Minionklassen, das jeweilige Image auf das Tableelement gesetzt.

4.2.4 Gestaltung des Spiels mittels CSS

Das HTML Dokument wurde in verschiedene Div-Container unterteilt, um diese individuell mit CSS stylen zu können.

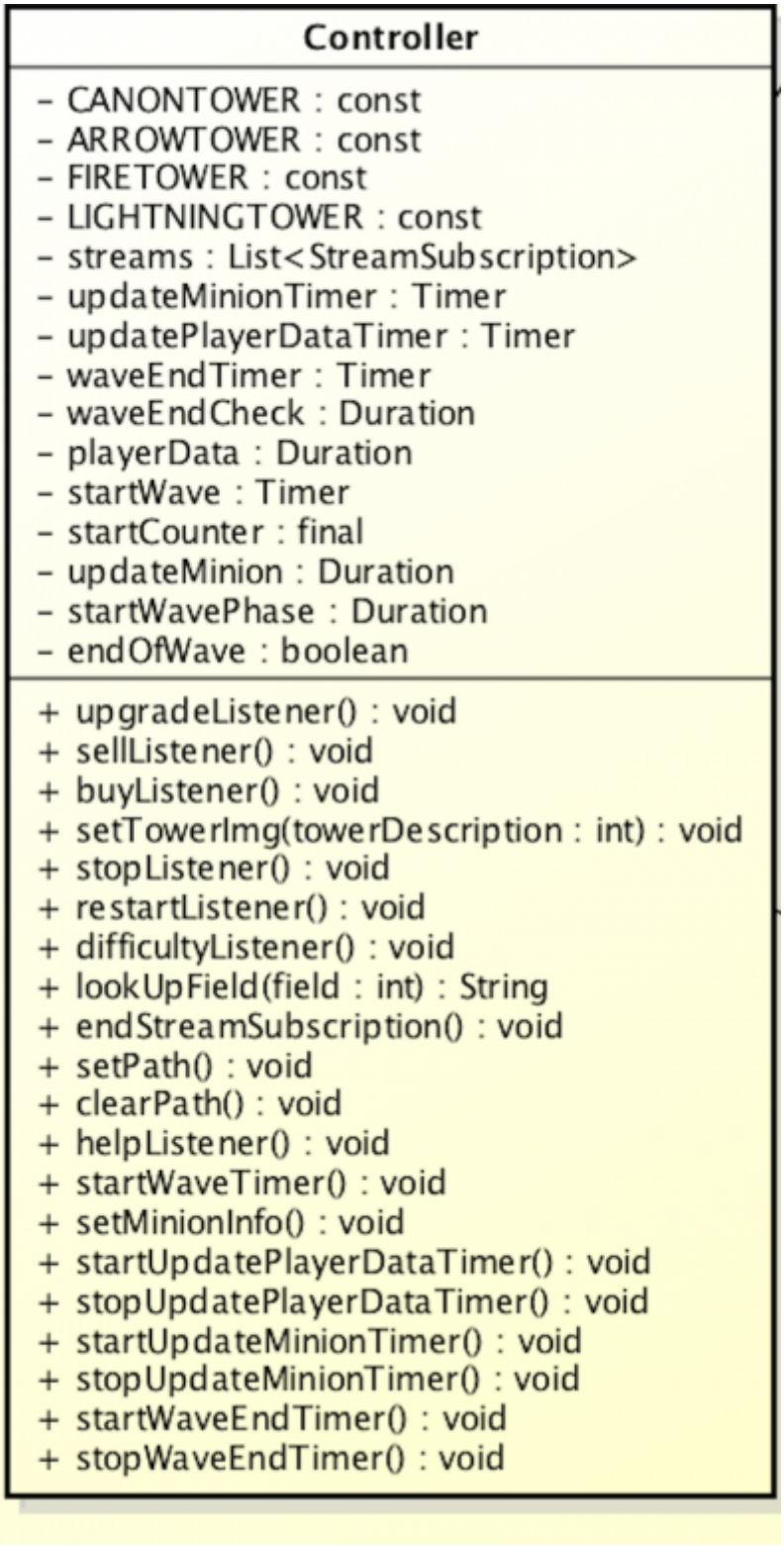
Als Hintergrund wurde eine GIF verwendet. Die einzelnen Tabellenelemente erhalten im Spiel Klassen, abhängig von Objekten(Tower/Minion), welche das jeweilige Hintergrundbild bestimmen sollen.

Die Tabellenelemente sind auf 38×38 Pixel skaliert.

Die Interaktionsbuttons wurden so gestyled, dass sie in einem guten Kontrast zum Hintergrund stehen und sichtbar sind. Die Schriftgrößen wurden möglichst groß und leserlich gehalten.

4.3 Controller

4.3.1 Überblick über ihren Controller



4.3.2 Erläuterung der Interaktionsmöglichkeiten des Nutzers

Türme kaufen

Der Spieler kann durch den Buy Button neue Türme kaufen. Dafür klappt ein Menü auf, dass die 4 Turm-Arten zeigt.

Wählt der Spieler einen Turm aus, kann er mit einem Klick auf ein Grassfeld den Turm dorthin setzen, sollte er genug Gold haben.

Türme verkaufen

Der Spieler kann durch den Sell Button Türme verkaufen, dafür klickt er im Anschluss auf einen Turm, den er verkaufen möchte. Das Gold wird ihm gutgeschrieben. Der Turm verlässt das Brett und Spiel.

Türme upgraden

Der Spieler kann durch den Upgrade Button einen Turm upgraden. Dafür klickt er im Anschluss auf den gewünschten Turm und sollte der Spieler genug Gold haben, wird das Upgrade durchgeführt. Dies wird durch ein neues Tower Image dargestellt.

Spiel stoppen

Der Spieler hat während einer Welle durch den Stop Button die Möglichkeit das Spiel zu stoppen, um in Ruhe Türme zu setzen, upgzugraden oder neue Strategien zu entwickeln.

Spiel fortsetzen

Der Spieler kann mit dem Start Button das Spiel wieder aufnehmen, nachdem er es gestoppt hat.

Spielhilfe anzeigen

Der Spieler kann die Erklärung des Spielprinzips der Tower Defense über den Game Info Button sich anzeigen lassen. Mit einem weiteren Klick auf den Button, wird die Info wieder eingeklappt.

Turm Hilfe anzeigen

Der Spieler kann mit einem Klick auf den Tower Info Button sich weitere Infos über die Türme anzeigen lassen, was z.B. die Arten der Türme und Schadensklassen an geht. Mit einem weiteren Klick auf den Button, wird die Info wieder eingeklappt.

Rüstungsklassen Info anzeigen

Der Spieler kann mit einem Klick auf den Armor Info Button sich weitere Infos zu den Rüstungsklassen und der Wirksamkeit der Schadensklassen gegen diese anzeigen lassen. Mit einem weiteren Klick auf den Button, wird die Info wieder eingeklappt.

4.3.3 Erläuterung des Timers

Start Wave Timer

Der Start Wave Timer zeigt das Label an, welches den Countdown bis zum Start der Wave angibt und aktualisiert diesen sekündlich. Ist der Countdown auf 0, wird das Spiel gestartet, die Minion Info und Level Infos gesetzt, sowie der Update Minion Timer und Wave End Timer aufgerufen. Außerdem wird der Stop Button dann angezeigt.

Der Start Wave Timer wird dann beendet und gelöscht.

Update Player Data Timer

Der Update Player Data Timer setzt den Spieler Namen und aktualisiert die Punkte-, Gold- und Lebensanzeige. Für die Goldanzeige ruft er im Game die Methode auf, um die getöteten Minions auszuwerten, damit das Game den Goldbetrag des Spielers auf einen aktuellen Stand bringt. Der Timer ruft dann über das Game den Goldbestand des Spielers ab und gibt diesen an den View weiter, damit die Darstellung angepasst wird.

Update Minion Timer

Der Update Minion Timer setzt die Bilder der Minions entsprechend der Positionen der Minions. Dafür ruft er über das Spiel die aktiven Minions auf und löscht übergibt den View deren alte Position, damit die Bilder gelöscht werden können und danach die neue Position, wo das Bild nun gesetzt werden soll.

Außerdem wertet der Timer die getöteten Minions und die, die das Ende des Pfads erreicht haben, aus, damit die Bilder vom View gelöscht werden können. Dann gibt der Controller dem Model das ok, damit die Minions gelöscht werden können.

Über das Game lässt er in dem Schritt auch die Anzahl der durchgelassenen und getöteten Minions erhöhen.

Wave End Timer

Der Wave End Timer aktualisiert die Anzeige der verbleibenden Minion dieser Wave und holt sich dafür über das Game die Anzahl der lebenden Minions, die noch auf dem Brett sind oder noch das Spielbrett betreten.

Ansonsten prüft der Timer, ob das Spiel oder Level zu Ende ist und ruft vom View dementsprechend die Methoden auf, um z.B. den Pfad von den Minion Images zu säubern.

Sollte die Wave zu Ende sein, wird der Pfad über die View gesäubaert und der Wave Timer wieder gestartet, damit die nächste Wave beginnen kann.

Wenn das Spiel zu Ende ist, wird vom Model die Methode endOfGame aufgerufen, die das Spiel beendet, und die Timer des Controllers werden beendet und die Auswahl des Schwierigkeitsgrads wird wieder angezeigt. Der Spieler kann nun ein neues Spiel starten.

4.3.4 Abbildung der Nutzerinteraktionen auf Modelinteraktionen

Buy

Wurde vom Spieler der Button „Buy“ angeklickt, kann er 4 verschiedene Tower auf das Spielfeld setzen.

Sobald ein Feld angeklickt wurde, wird das entsprechende Towerobjekt erzeugt und in der TowerAdminstration in die Liste geschrieben. Das Gold vom Spieler wurde intern angepasst.

Sell

Klickt der Spieler den Button „Sell“, so wird über ein FieldMatching(hierzu hat jedes Tableelement eine ID, welche mit der Position des Towers abgeglichen wird) das richtige Towerobjekt ausgewählt, aus der Liste der TowerAdmin gelöscht und zerstört.

Upgrade

Bei einem Klick auf „Upgrade“ wird ebenfalls über ein FieldMatching das richtige Objekt rausgesucht und eine entsprechende Methode in der TowerAdmin aufgerufen. Diese berechnet dann die neuen Werte des Towers und überschreibt sie mit den alten.

Stop

Sobald der Spieler den Stop Button drückt, werden alle Spieltimer, sowie der Minionbewegungstimer gestoppt.

Start

Sobald der Spieler den Start Button drückt, werden die vom Stop Button deaktivierten Timer wieder neu gestartet.

4.3.5 Abbildung der Timer-Events auf Modelinteraktionen

Start Wave Timer

Der Start Wave Timer ruft beim Ablauf des Countdowns die Start Game Methode des Games auf und startet so das Spiel.

Update Player Data Timer

Der Update Player Data Timer stößt im Game das Zusammenrechnen des von den Minions fallen gelassenen Goldes an. Das Spiel weißt dann dem Spieler dieses Gold zu.

Update Minion Timer

Der Update Minion Timer wertet für den View die Minions aus, die vom Spielbrett als Bild gelöscht werden müssen und speichert diese intern in zwei verschiedene Listen. Diese Listen werden dann durchlaufen und nach dem Löschen im View an das Game und die Level Administration gegeben, damit die Minions aus der aktiven Minions Liste gelöscht werden können.

Die aktiven Minions, die er überprüft, holt er sich vom Spiel.

Wave End Timer

Der Wave End Timer lässt durch das Spiel prüfen, ob die Wave, das Level oder sogar das Spiel zu Ende ist.

Sollte das Spiel zu Ende sein, wird der Timer über das Spiel die letzte Auswertung des erhaltenen Golds anstoßen und die Punkteberechnung.

Das Spiel wird dann aufgefordert sich zu beenden.

5. Zusammenfassung und Ausblick

5.1 Zusammenfassung

Zusammenfassend kann man sagen, dass das Model unabhängig vom View ist und der Controller sich in Hinsicht auf den View damit auseinandersetzt, die Positionen der Minions an den View (um damit die Minion Images zu setzen) zu übergeben und die Position sowie die Tower Beschreibung zum Setzen der Tower. Dies wird über Timer realisiert und die Nutzerinteraktionen über die Listener, die dann häufig View Methoden zum Einklappen von View Elementen zur Folge haben.

Wenige Timer, z.B. der Start Wave Timer oder Wave End Check Timer, Triggern aus dem Controller heraus Methoden im Model.

Ansonsten lässt das Model, sobald das Spiel läuft, die Minions durch eigene Trigger eigenständig starten, sich bewegen und die Türme darauf schießen. Das Prüfen, ob Leben verloren wurden, gehört auch dazu.

Wenn wir dieses Projekt ein zweites Mal schreiben würden, mit den jetzigen Erfahrungen, hätten wir von Anfang an an mehrere Elemente im Vorwege gedacht z.B. die View Elemente, die für den Benutzer wichtig sind (Infos der Leben, Level, Minions), und die Controller Interaktionen mit View und Model.

Auch in Zusammenhang mit den Timern.

5.2 Ausblick

In weiteren Versionen könnte man eine Animation zum Schießen einbauen, die anzeigt welcher Minion getroffen wurde.

Außerdem wären ordentliche Fehlermeldungen für das Platzieren von Türmen auf einem falschen Feld und wegen fehlendem Geld notwendig.

Das Spielkonzept dieser Tower Defense beinhaltet nicht, dass Minions zufällig sich auf dem Pfad bewegen oder über das Brett fliegen, jedoch könnte man überlegen das Spiel daraufhin auszulegen oder besser noch auf einem Canvas gezeichnet mit einer eigenen Wegfindung für die Minions zu versehen, die auch darauf reagiert, dass der Spieler mit seinen Türmen die Minions zu einem bestimmten Weg, aufgrund deren Wegfindung, zwingt.

So kann der Spieler eine sogenannte Maze bauen, wie es auch in einigen Tower Defense Spielen im Grundkonzept verankert ist.