

---

## Gruppe GG: Konzeptabgabe

Mitglieder: Christop Bathon, Johanna Kipping, Fabian Kropfhamer, Ilijana Spasojevic

### 1. Spiel mit geplantem Feature-Umfang

#### 1.1 Überblick über das geplante Spiel

Das Endprodukt soll ein Schießspiel sein, bei dem keine normalen Kugeln geschossen werden, sondern Paintballs. Beide Spieler gehen mit einer vollen Lebensleiste ins Spiel. Bei jedem Schuss den eine Figur abbekommt, verliert sie ein Leben. Sobald die Lebensleiste leer ist, d.h. man zu häufig abgeschossen wurde, ist das Spiel beendet. Das Ziel ist es den gegnerischen Spieler so oft abzuschossen bis dessen Lebensleiste aufgebraucht ist.

Die Arena wird statisch aus der Vogelperspektive dargestellt.

In der Spielarena befinden sich:

- Objekte hinter denen man sich verstecken kann
- Gegenstände, die man als Feature einsammeln kann
- Bereiche, die die Spielfigur verlangsamen/ beschleunigen

#### 1.2 MVP mit geplantem Feature-Umfang

Das Minimal Valuable Product (**MVP**) besteht aus folgenden Komponenten:

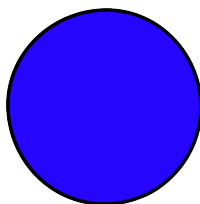
- 2 Spieler mit Waffen (können schießen und sich bewegen)
- Statische Arena aus Vogelperspektive
- Schießen sich gegenseitig ab
- Lebensanzeige: Wird weniger, wenn man abgeschossen wird
- Ablauf:
  - Beginn: Jede Figur auf einer Seite der Arena
  - Sieg: Wenn gegnerische Lebensanzeige leer
  - Ziel: Gegner so oft abzuschießen bis dessen Lebensanzeige aufgebraucht ist

Zu den geplanten Features gehören:

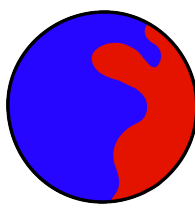
- Spielanleitung (zu Beginn)
- Charakter aussuchen (zu Beginn)
- Lebensleiste
- Schild zur Abwehr
- Power-Ups
  - Leben auffüllen
  - Geschwindigkeit/Schiessgeschwindigkeit erhöhen
- Portale zum Durchlaufen
- Eis (Beschleunigen Spiele) & Sand (Verlangsamen Spieler) auf Spielfläche
- Sound hinzufügen

#### 1.3 Spiel-Design

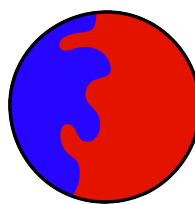
Jeder Spieler hat eine feste Paintball-Farbe, mit der er versucht seiner Gegner komplett einzufärben. Die Darstellung der Figur entspricht somit der "Lebensleiste". Verliert man ein Leben, so wird die Spielfigur ca 33% mit der gegnerischen Farbe eingefärbt. Verliert man zwei Leben sind es ca 66%. Hat man alle drei Leben verloren ist man komplett mit der gegnerischen Farbe eingefärbt und das Spiel ist vorbei. Erste Design-Entwürfe sehen wie folgt aus:



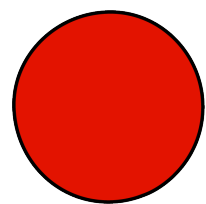
3 Leben



2 Leben



1 Leben



Game Over

## **2. Spiel-Modellierung**

### **2.1 Sequenz-Diagramm**

Für den Ablauf des Spieles ergeben sich 3 unterschiedliche Sequenzdiagramme, je nach Szenario (Siehe Anhang). Das erste Diagramm stellt die Spielroutine mit Verbindungsaufbau dar. Die anderen beiden Diagramme den Spielvorgang bei regulärem Spielende mit Gewinnen/Verlieren (Anhang 5.2) und das außerordentliche Beenden des Spieles dar (Anhang 5.3). Bei außerordentlicher Spielbeendigung wird die Verbindung von einem der beiden Spielern beendet und somit der andere Spieler automatisch benachrichtigt und als Sieger erklärt. Bei regulärer Spielbeendigung, d.h. einer der beiden Spieler hat gewonnen während der Andere verloren hat, gelangt man zu einem „End-Bildschirm“, der über die Option verfügt nochmal ein Spiel-Request zu senden.

### **2.2 UML-Klassendiagramm**

Das Spiel wird wie folgt gegliedert sein: (Anhang 5.4)

### **2.3 Ordner-Struktur**

Um eine klare Übersicht über die Klassen zu gewinnen, wird die Ordnerstruktur eingeteilt in einen Server- und Client-Ordner. Diese enthalten die im UML-Diagramm gegebenen Klassen (siehe Anhang 5.5)

## **3. Milestone-Plan**

Es ist geplant die Milestones 2-3 Wochen vor dem offiziellen Abgabetermin einzureichen.

Geplante Umsetzung des Spiels	
<b>Milestone I</b>	Server-Client-Struktur mit bereits implementiertem MVP
<b>Milestone II</b>	MVP mit: <ul style="list-style-type: none"><li>- Objekte und ihrer Collision Detection</li><li>- Lebensleiste</li></ul>
<b>Milestone III</b>	Endgültiges Spiel mit Features

In der ersten Phase ist geplant, dass die Gruppe gemeinsam eine Server-Client-Struktur und ein MVP programmiert, bevor es dann zur einzelnen Bearbeitung von Features geht. Somit soll sichergestellt werden, dass jedes Team-Mitglied das Konzept des MVPs und die Grundstruktur der Architektur verinnerlicht. Außerdem gestaltet es sich schwierig Aufgaben aufzuteilen, bevor nicht ein erstes Rohprodukt steht, das man dann gemeinsam überarbeitet. Sobald das MVP fertiggestellt ist, werden die Features an die einzelnen Gruppenmitglieder aufgeteilt und schrittweise in das bestehende Spiel integriert.

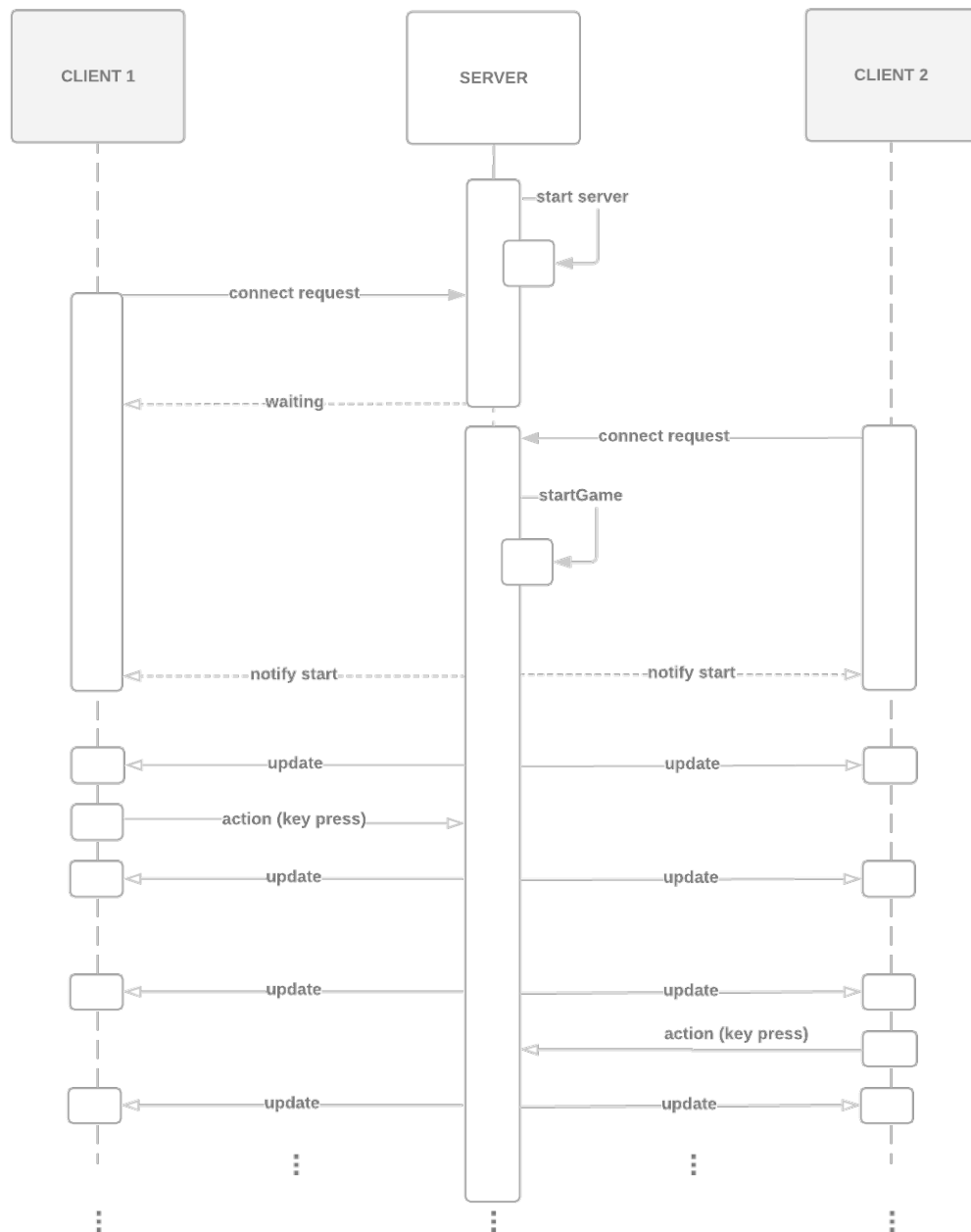
## **4. Herausforderungen und erwartete Probleme**

Zu den möglichen Herausforderungen während der Spieleentwicklung gehören:

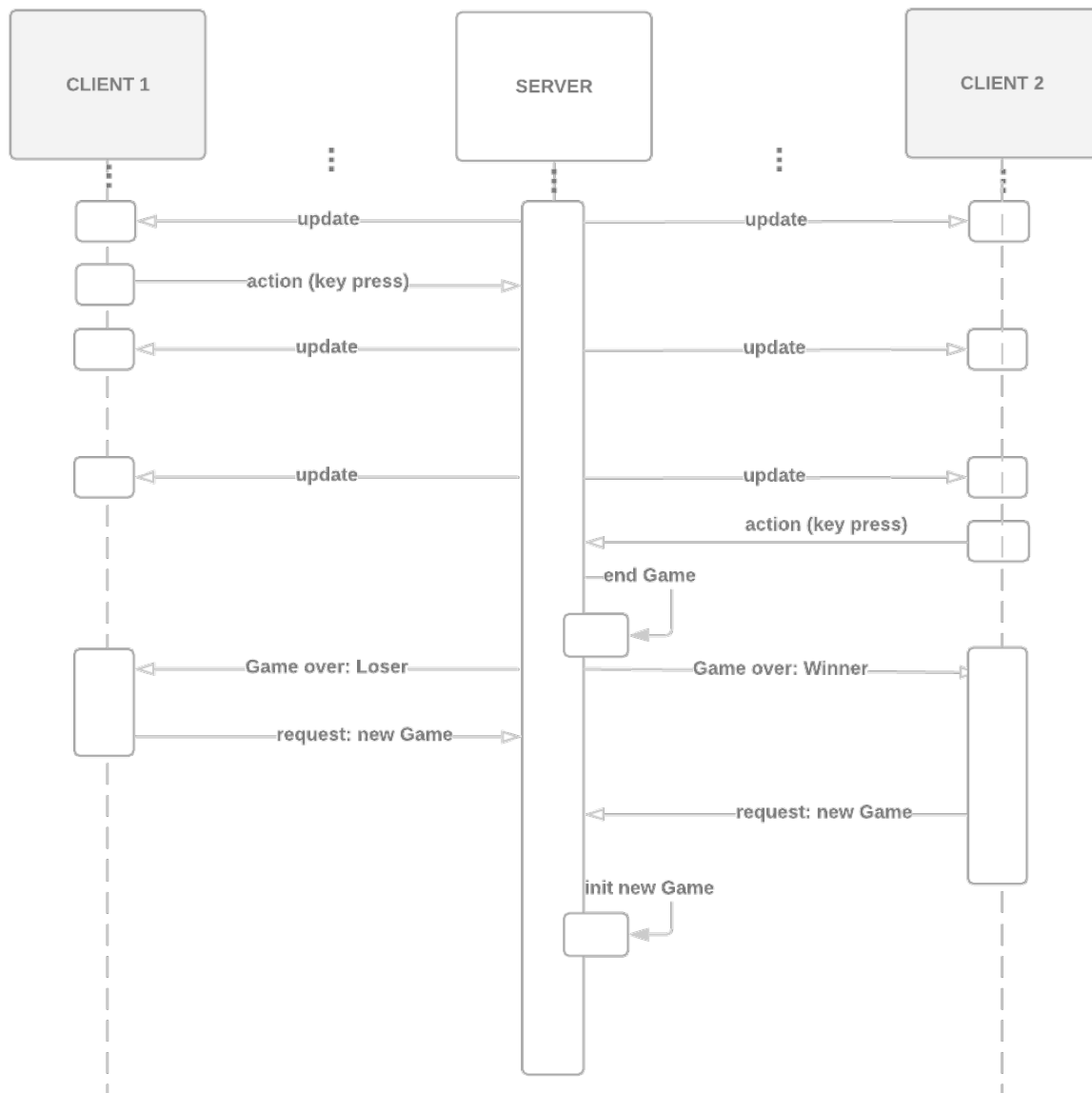
- Probleme geplante Aufgaben einzuteilen
- Programmierkenntnisse, die sich erst im Laufe des Semesters stetig verbessern
- Mangelnde Zeit (Besuch anderer Vorlesungen, Feiertage wie Weihnachten etc.)
- Mögliche Probleme bei Mergen von getrennt programmierten Features
- Umfang und Schwierigkeit der kommenden Aufgaben im Voraus noch unklar
- Einzelnen Team-Mitglieder kennen ihre Stärken hinsichtlich Programmierung noch nicht so gut

## 5. Anhang

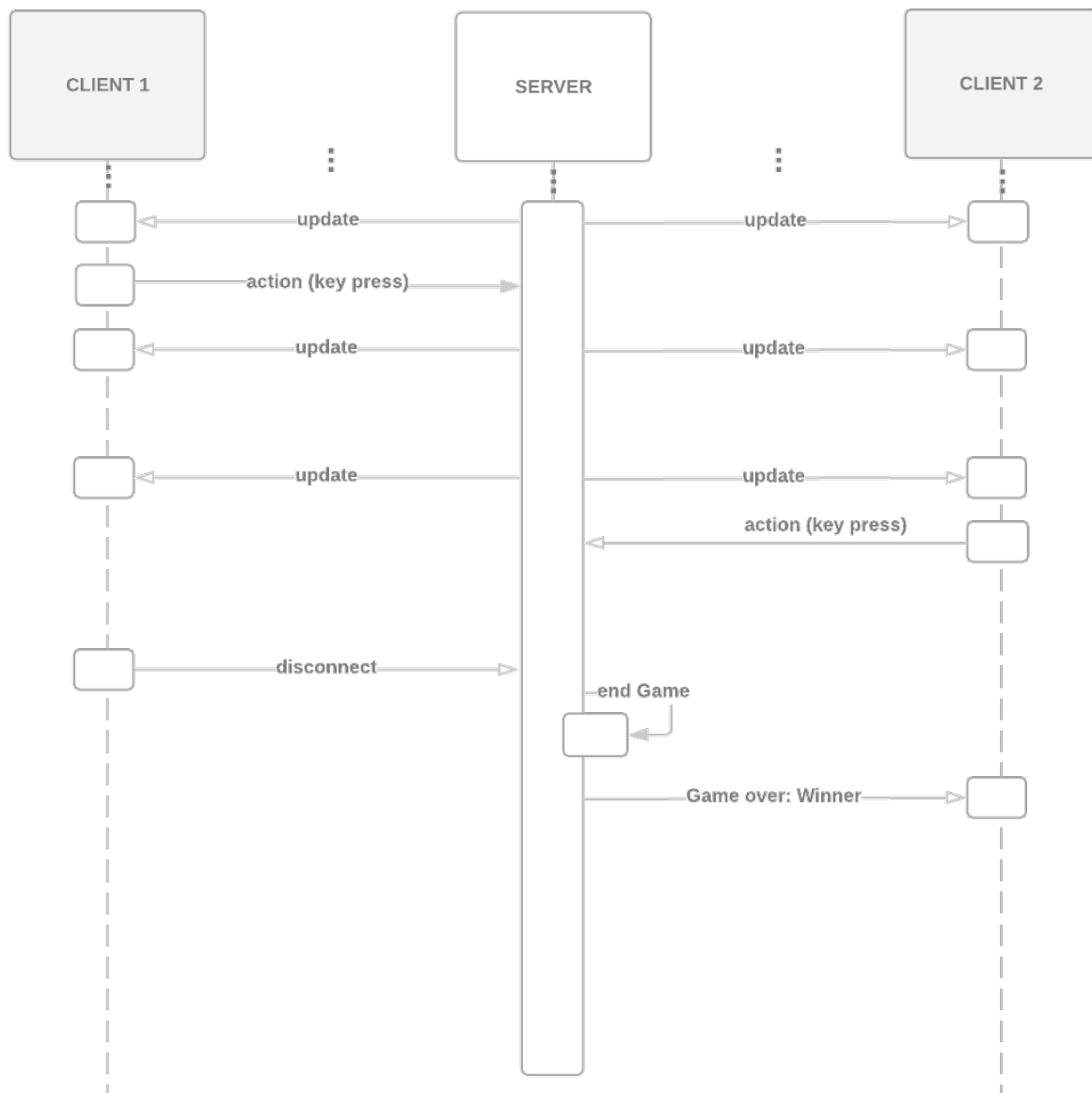
### 5.1. Sequenzdiagramm



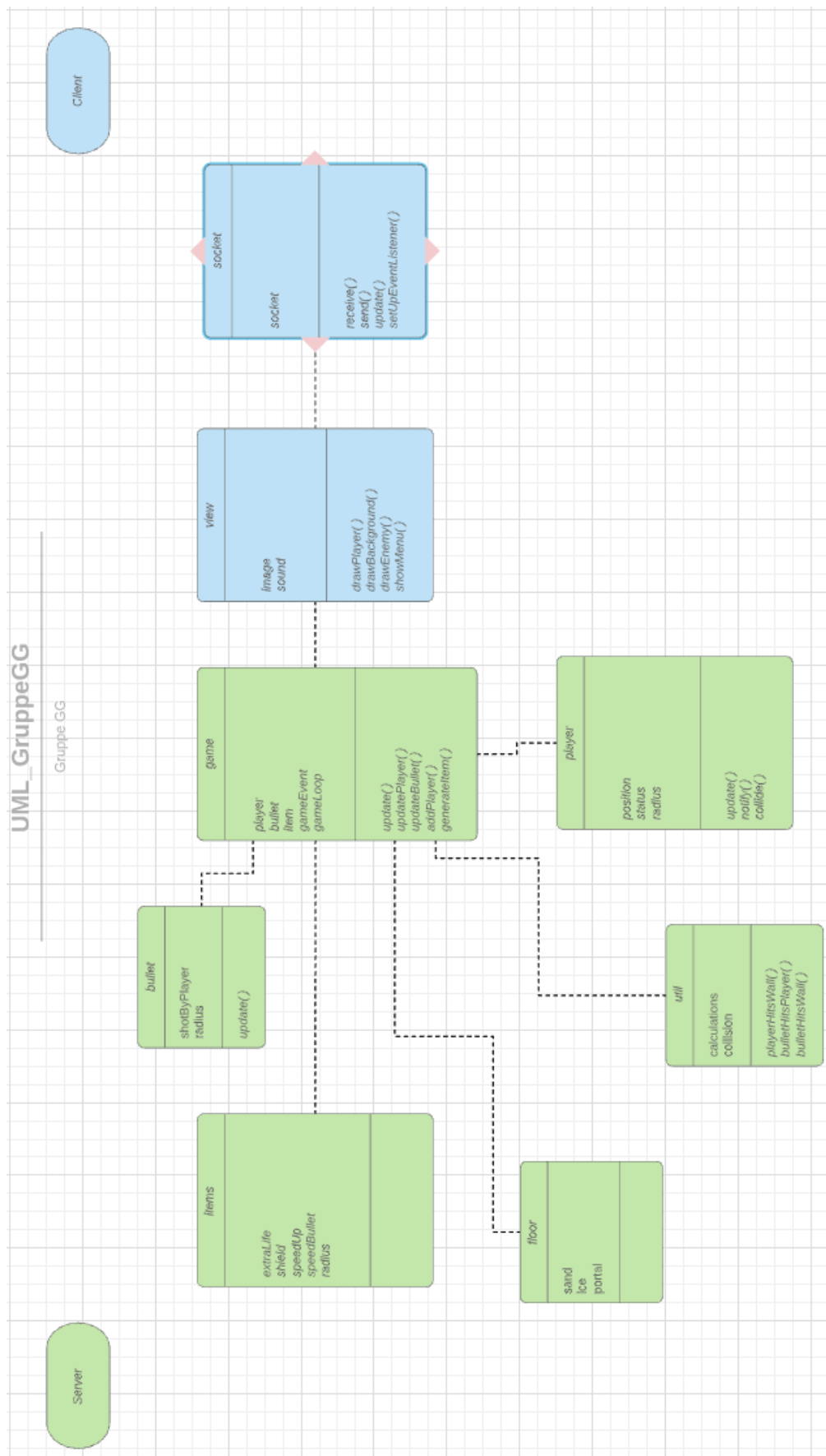
## 5.2 Sequenzdiagramm



### 5.3 Sequenzdiagramm



## 5.4 UML-Diagramm



## 5.5 Ordner-Struktur

gg

```
|__ dist
|__ node_modules
|__ public
|__ src
    |__ client
        |__ css
        |__ img
        |__ js
        |__ Model
        |__ view
        |__ Controller
    |__ server
        |__ Model
        |__ Controller
```