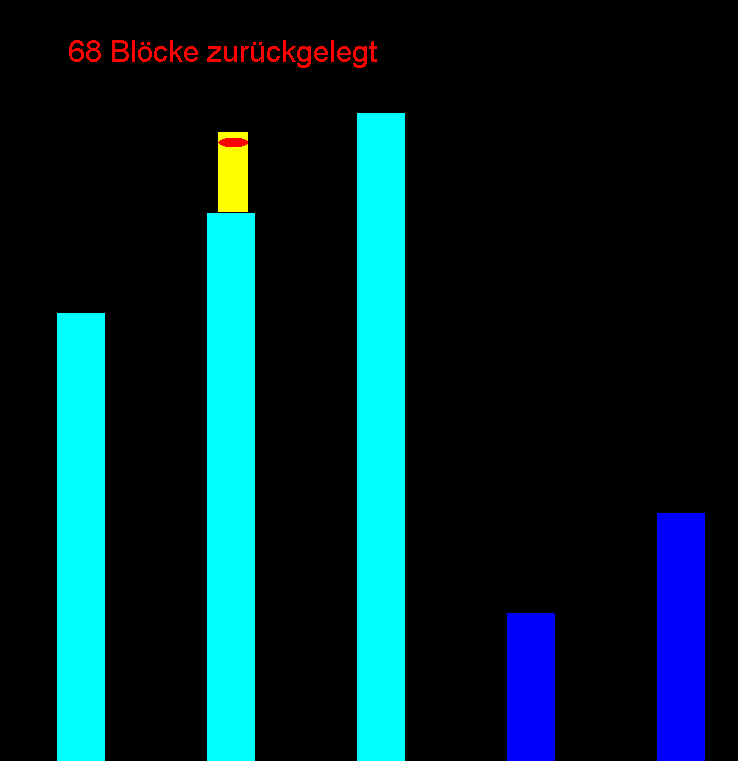
Projektdokumentation Modul 226b, Jump & Run



Daniel Wilmes

Inhalt

[Spielidee 3](#_Toc94545153)

[Mindmap 3](#_Toc94545154)

[Klassendiagramm 4](#_Toc94545155)

[Vererbung 4](#_Toc94545156)

[Funktionen erklärt 5](#_Toc94545157)

[Sprung Funktion 5](#_Toc94545158)

[Kollision Funktion 5](#_Toc94545159)

[Automatisches Löschen von Wandstücken 5](#_Toc94545160)

[Testing 6](#_Toc94545161)

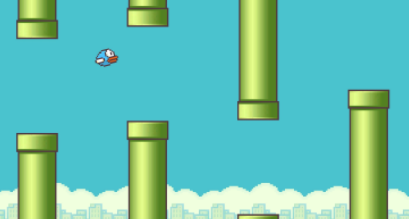
[Teste y-Koordinate von Player-Klasse 6](#_Toc94545162)

[Teste x-Koordinate von Wall-Klasse 6](#_Toc94545163)

[Teste die Breite der Wall-Klasse 6](#_Toc94545164)

[Fazit 7](#_Toc94545165)

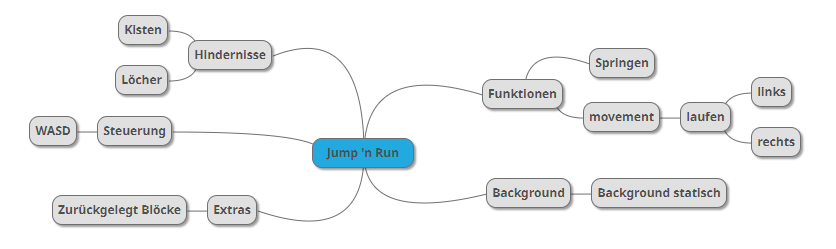
## Spielidee

Ich spiele sehr gerne Jump and Run Spiele und habe mich dazu entschlossen auch ein Spiel in diesem Genre zu machen. Die Idee war ein Spiel, das in die Richtung von Flappy Bird geht, aber statt durch die Hindernisse zu fliegen, springt man von Plattform zu Plattform. Die Schwierigkeit war hierbei nicht runterzufallen.

Erfolgsspiel: Flappy Bird

## Mindmap

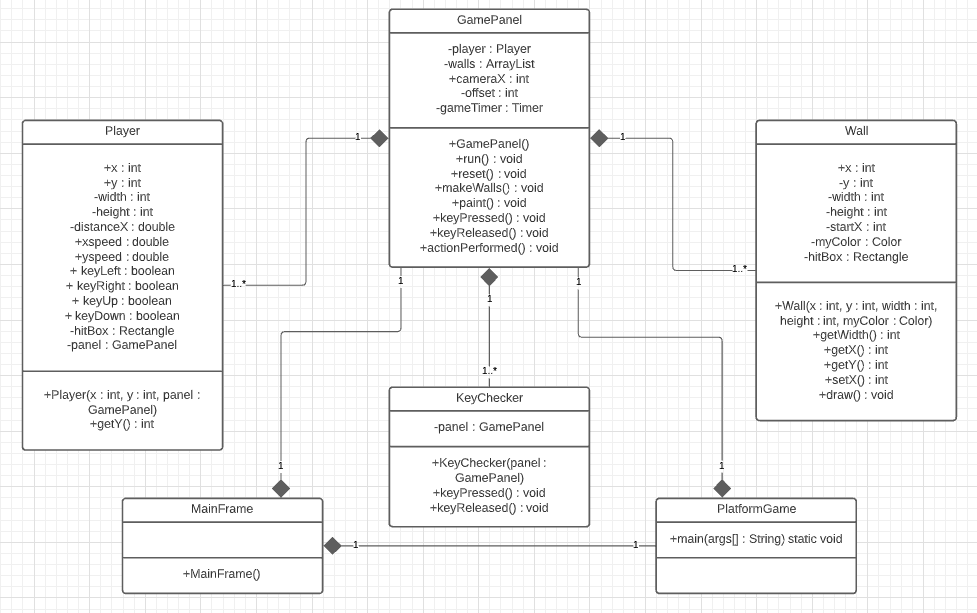
Eine Mindmap ist für mich ein guter Weg alle meine Gedanken zum Projekt aufzuschreiben.



## Klassendiagramm

In meinem habe ich 6 verschiedene Klassen genutzt.

Die Player und Wall Klasse definieren die Werte für die Spielerfigur und die Hindernisse. Diese zwei Klassen sind mit einer Komposition mit der Klasse GamePanel verbunden, den ohne die Klasse GamePanel würde keine von beiden Klassen einen weiteren Sinn haben.   
Die Klasse KeyChecker, die zuständig für das Abhören der Bedienung durch die Tastatur ist, ist auch mit einer Komposition mit der Klasse GamePanel verbunden. Die Klasse GamePanel ist für die Spielmechanik zuständig aber könnte nicht ohne die Main Klasse PlatformGame auskommen.   
Die MainFrame Klasse stellt das ganze Spiel dar.



## Vererbung

In diesem Snippet der Klasse GamePanel sieht man sehr gut, die Vererbung der Klassen Player und Wall. Ohne diese Klassen würde alle Werte und Funktionen von der Player und Wall-Klasse fehlen.

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

## Funktionen erklärt

### Sprung Funktion

Ich habe mich sehr lange mit dieser Funktion beschäftigt, da ich nie wusste wie ich diese schreiben soll. Zusätzlich musste noch eine Kollision Abfrage in dieser Funktion sein, dass wen ein Block im Weg ist, der Spieler bei Berührung wieder runterfällt.   
Intercects ist ein Boolean der prüft ob sich die Hitboxen des Spielers und der Wandstücke überschneiden.



### Kollision Funktion

Das ist der Code für die Vertikale Kollision, da die Horizontale eigentlich gleist ist, bis auf das Vertauschen des y zum x- Wert ist habe ich nur diesen drin.

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

### Automatisches Löschen von Wandstücken

Dieser Snippet ist dafür da, dass wen eine Wand hinter der x- Koordinate -900 gelöscht werden soll. Somit bleibt das Spiel performant auch wenn man sehr weit ist.

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

## Testing

### Teste y-Koordinate von Player-Klasse

Prüfe, ob die y-Koordinate auch richtig übernommen wird im Konstruktor

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

### Teste x-Koordinate von Wall-Klasse

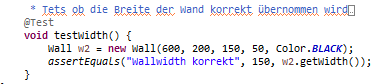
Prüfe, ob die x-Koordinate auch richtig übernommen wird im Konstruktor

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

### Teste die Breite der Wall-Klasse

Prüfe, ob die Width korrekt übernommen wird, für die Kollision



## Fazit

Mit dem Endprodukt bin ich sehr zufrieden, es ist noch ausbaufähig jedoch funktioniert es so wie ich es mir vorgestellt habe was mich sehr glücklich macht.   
Für die Zukunft habe ich aber sehr viel dazu gelernt durch dieses Projekt. Die Zeit, die man hat, sollte man auch nutzen und nicht versuchen alles am Ende zu machen, den man weiss nie, ob etwas dazwischenkommt, was den kompletten Zeitplan auf den Kopf schmeisst.