**<Logo>**

**Grobkonzept**

Retro-Indie-Spiel Spielobjekte

**Inhaltsverzeichnis**

[1 Allgemeines 2](#_Toc435525031)

[1.1 Ziel und Zweck des Dokuments 2](#_Toc435525032)

[1.2 Besonderheiten an GameMaker Studio 2](#_Toc435525033)

[1.3 Abkürzungen <TODO falls vorhanden> 2](#_Toc435525034)

[2 Bemerkungen zu GameMaker Studio: Pro 2](#_Toc435525035)

[3 Spielobjekte 3](#_Toc435525036)

[3.1 Dekorationen 3](#_Toc435525037)

[3.1.1 Steinwand 3](#_Toc435525038)

[3.1.2 Teppich 3](#_Toc435525039)

[3.1.3 Wandregal 4](#_Toc435525040)

[3.1.4 Tisch 4](#_Toc435525041)

[3.1.5 Stuhl 4](#_Toc435525042)

[3.1.6 Kronleuchter 5](#_Toc435525043)

[3.1.7 Fackel 5](#_Toc435525044)

[3.1.8 Kerze 5](#_Toc435525045)

[3.2 Interaktive Objekte 6](#_Toc435525046)

[3.2.1 Bücherschrank 6](#_Toc435525047)

[3.2.2 Tür 6](#_Toc435525048)

[3.2.3 Schlüssel 6](#_Toc435525049)

[3.2.4 Schalter 7](#_Toc435525050)

[3.2.5 Gittertor 7](#_Toc435525051)

[3.2.6 Fliegendes Buch 7](#_Toc435525052)

[3.2.7 Stacheln 8](#_Toc435525053)

[3.2.8 Leiter 8](#_Toc435525054)

[3.2.9 Spielstart Menü <TODO> 8](#_Toc435525055)

[3.2.10 Laden Menü <TODO> 8](#_Toc435525056)

[3.2.11 Einloggen – Registrieren Menü <TODO> 8](#_Toc435525057)

[3.3 Nicht sichtbare Objekte <TODO> 8](#_Toc435525058)

[3.3.1 Lichtquelle 8](#_Toc435525059)

[3.3.2 Textfeld 8](#_Toc435525060)

[3.4 Spieler <TODO> 8](#_Toc435525061)

[3.5 Gegner <TODO> 8](#_Toc435525062)

1. Allgemeines
   1. Ziel und Zweck des Dokuments

Dieses Dokument enthält ein grobes Konzept aller Spielobjekte. Es werden die Spielobjekte vorgestellt und die Zusammenhänge erklärt. Aus den Erklärungen können dann die Feinplanung mittels UML Diagramme entstehen.

* 1. Besonderheiten an GameMaker Studio
  2. Abkürzungen <TODO falls vorhanden>

1. Bemerkungen zu GameMaker Studio: Pro
2. Spielobjekte

<Beschreibung?>

* 1. Dekorationen

Dekorationen sind Spielobjekte, die meistens nur eine Kollision besitzen. Da die Kollision im Spieler berechnet wird, gibt es kaum Quellcode, der für diese Objekte benötigt werden. Das Verhalten der Objekte ähnelt sich stark. Der Unterschied besteht nur aus dem Bild, der Kollisionsbox, und ob das Objekt eine volle, halbe oder keine Kollision besitzt. Eine volle Kollision bedeutet, dass das Spielerobjekt, egal von welcher Seite, das Objekt nicht überschneiden kann. Bei einer halben Kollision gilt dies nur von der oberen Seite, der Spieler kann also durch das Objekt laufen und auf dem Objekt stehen.

* + 1. Steinwand

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | obj\_stonewall | **Priorität** | hoch |
| **Status** | fertig |
| **Kollision** | voll |
| **Beziehung** | obj\_carpet | | |
| **UML Referenz** |  | | |

Die Steinwand dient zum Abgrenzen des Raumes, indem sich der Spieler befindet. Bis auf die Kollision besitzt die Steinwand keiner weiteren Funktionalitäten. Die Steinwand besitzt eine Beziehung zu **obj\_carpet**. Immer wenn über einer Steinwand keine weitere Steinwand vorkommt, dann wird ein Teppich exakt auf die Oberkante der Steinwand gelegt. Dadurch wird ein Akzent gesetzt, sodass der Fußboden sich etwas von den Wänden und Decken unterscheidet.

* + 1. Teppich

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | obj\_carpet | **Priorität** | niedrig |
| **Status** | offen |
| **Kollision** | keine |
| **Beziehung** | obj\_stonewall | | |
| **UML Referenz** |  | | |

Der Teppich ist nur ein Bildelement ohne jegliche Funktionalität. Hier muss eine eigene Klasse erstellt werden, damit die Steinwand das Richtige Objekt findet.

* + 1. Wandregal

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | obj\_shelf | **Priorität** | hoch |
| **Status** | offen |
| **Kollision** | halb |
| **Beziehung** | obj\_player | | |
| **UML Referenz** |  | | |

Das Wandregal dient als Sprungplattform für die Spielfigur. Dieses Objekt besitzt im Kern eine volle Kollision. Dank dem GameMaker können wir aber eine halbe Kollision simulieren. Solange die unterste Koordinate des Spielerobjektes unterhalb der obersten Koordinate des Wandregals Kollision Box ist, wird das Wandregal zwar gezeichnet, aber die Kollisionsmaske ist aus. Erst wenn der Spieler über dem Wandregal ist, wird diese eingeschaltet.

* + 1. Tisch

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | obj\_table | **Priorität** | mittel |
| **Status** | offen |
| **Kollision** | halb |
| **Beziehung** | obj\_player | | |
| **UML Referenz** |  | | |

Der Tisch besitzt im Kern keine spielerische Relevanz. Trotzdem gehört dieses Spielobjekt zu den Objekten mit halber Kollision. Die Kollision wird wie bei obj\_shelf berechnet.

* + 1. Stuhl

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | obj\_shelf | **Priorität** | niedrig |
| **Status** | offen |
| **Kollision** | keine |
| **Beziehung** | -- | | |
| **UML Referenz** |  | | |

Der Stuhl ist ein reines Dekorationselement, welches im Raum platziert werden kann. Der Spieler kollidiert weder mit diesem Objekt und es gibt keine Interaktion mit dem Objekt.

* + 1. Kronleuchter

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | obj\_chandalier | **Priorität** | hoch |
| **Status** | offen |
| **Kollision** | keine |
| **Beziehung** | obj\_lightSource | | |
| **UML Referenz** |  | | |

Der Kronleuchter ist ein Lichtträger im Spiel. An allen Stellen, wo ein Kronleuchter erzeugt wird, erzeugt der Kronleuchter eine Lichtquelle (**obj\_lightSource**). Dabei wird angegeben, wie weit der Lichtkegel der Lichtquelle reicht.

* + 1. Fackel

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | obj\_torch | **Priorität** | hoch |
| **Status** | offen |
| **Kollision** | keine |
| **Beziehung** | obj\_lightSource | | |
| **UML Referenz** |  | | |

Die Fackel besitzt das gleiche verhalten wie der Kronleuchter (**obj\_chandalier**).

* + 1. Kerze

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | obj\_candle | **Priorität** | hoch |
| **Status** | offen |
| **Kollision** | keine |
| **Beziehung** | obj\_lightSource | | |
| **UML Referenz** |  | | |

Die Kerze besitzt das gleiche verhalten wie der Kronleuchter (**obj\_chandalier**).

* 1. Interaktive Objekte
     1. Bücherschrank

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | obj\_bookcase | **Priorität** | hoch |
| **Status** | offen |
| **Kollision** | voll |
| **Beziehung** | obj\_player, obj\_switch, obj\_stonewall | | |
| **UML Referenz** |  | | |

Der Bücherschrank ist ein bewegliches Objekt mit voller Kollision. Dies bedeutet, dass der Spieler den Bücherschrank verschieben kann und so eine Sprungplattform besitzt, damit er z.B. höhere Bereiche erreichen kann. Wenn der Bücherschrank gegen eine Wand geschoben wird, dann muss der Schrank vor der Wand stoppen.

* + 1. Tür

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | obj\_door | **Priorität** | hoch |
| **Status** | offen |
| **Kollision** | keine |
| **Beziehung** | obj\_player, obj\_key | | |
| **UML Referenz** |  | | |

Die Tür dient für den Übergang in die verschiedenen Level. Eine Tür öffnet sich erst, wenn der Spieler alle Schlüssel im Level aufgenommen hat. Danach öffnet sich die Tür und falls der Spieler die Aktionstaste betätigt, teleportiert sie ihn in das nächste Level. Außerdem wird das Spiel automatisch gespeichert.

* + 1. Schlüssel

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | obj\_key | **Priorität** | hoch |
| **Status** | offen |
| **Kollision** | keine |
| **Beziehung** | obj\_player, obj\_door | | |
| **UML Referenz** |  | | |

Der Schlüssel liegt verteilt im Raum herum. Wenn der Spieler den Schlüssel berührt, dann verschwindet dieser Schlüssel und erzeugt für eine kurze Zeit ein kleines Lichtaufflackern.

* + 1. Schloss <TODO>
    2. Schalter

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | obj\_switch | **Priorität** | mittel |
| **Status** | offen |
| **Kollision** | keine |
| **Beziehung** | obj\_player, obj\_bookcase, obj\_gridDoor, obj\_flyingBook | | |
| **UML Referenz** |  | | |

Der Schalter dient zum Öffnen von Gittertüren. Sobald sich der Spieler draufstellt, können verschiedene Gittertüren geöffnet werden. Dabei kann es sein, dass der Spieler nur den Schalter berühren oder ein Bücherregal drauf stellen muss.

Ein Schalter kann auch ein Fliegendes Buch in Bewegung setzen. Er aktiviert also bestimmte Objekte

* + 1. Gittertor

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | obj\_gridDoor | **Priorität** | mittel |
| **Status** | offen |
| **Kollision** | voll |
| **Beziehung** | Obj\_switch | | |
| **UML Referenz** |  | | |

Das Gittertor versperrt dem Spieler bestimmte Bereiche. Um das Tor zu öffnen muss der Spieler einen oder mehrere Schalter aktivieren.

* + 1. Fliegendes Buch

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | obj\_flyingBook | **Priorität** | niedrig |
| **Status** | offen |
| **Kollision** | halb |
| **Beziehung** | obj\_player, obj\_switch | | |
| **UML Referenz** |  | | |

Das Fliegende Buch bewegt sich von links nach rechts oder von oben nach unten. Dabei kann es sein, dass der Spieler vorher einen Schalter aktivieren muss, damit sich das Buch bewegt. Das Buch besitzt weiterhin eine halbe Kollision

* + 1. Stacheln

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | obj\_spikes | **Priorität** | hoch |
| **Status** | offen |
| **Kollision** | voll |
| **Beziehung** | obj\_player | | |
| **UML Referenz** |  | | |

Die Stacheln sind im Boden eingelassen. Sobald der Spieler die Stacheln berührt, stirbt dieser.

* + 1. Leiter

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | obj\_ladder | **Priorität** | niedrig |
| **Status** | offen |
| **Kollision** | keine |
| **Beziehung** | obj\_player | | |
| **UML Referenz** |  | | |

Die Leiter ist eine weitere Möglichkeit, damit der Spieler hohe Stellen erreichen kann. Sobald der Spieler die Leiter mit mehr als der Hälfte der Fläche berührt, kann der Spieler hochklettern. Um die Leiter zu verlassen, muss der Spieler springen.

* + 1. Spielstart Menü <TODO>
    2. Laden Menü <TODO>
    3. Einloggen – Registrieren Menü <TODO>
  1. Nicht sichtbare Objekte <TODO>
     1. Lichtquelle
     2. Textfeld
  2. Spieler <TODO>
  3. Gegner <TODO>