



GameCreator wird dir ermöglichen in

simplyn Schritten ein eigenes dreidimensionales „Jump and Run“ Spiel zu erstellen. Dies funktioniert, indem der Spieler und Blöcke in dem Spiel schon gegeben sind und du sie auf einfachen Wegen bearbeiten kannst, sodass du dein eigenes Spiel als Endprodukt hast. Um in GameCreator einzusteigen und es verwenden zu können bearbeite alle folgenden Aufgaben, in denen dir langsam und in kleinen Schritten erklärt wird, wie du GameCreator verwendest. Um Verständnis für die Umgebung zu bekommen schaue auf die Karte (MAP), auf dieser siehst du 4 Plattformen und zwischen diesen zum Teil schon Blöcke, mit denen du zu den verschiedenen Plattformen kommst. Deine Aufgabe wird es nun sein weitere Blöcke zwischen den Plattformen zu erstellen, damit man zu der letzten Goldenen Plattform gelangt. Damit du weißt, wie du dies anstellst befolge einfach den Aufgaben.

Aufgabe 1:

- Kopiere vom Tauschordner den Ordner „01 GameCreator“ in dein Gloop Verzeichnis, welches du im Informatik Ordner finden solltest, falls nicht, erstelle dieses und kopiere es nun hinein.
- Öffne die „package.bluej“ Datei um das BlueJ Fenster zu öffnen.
- Nun wirst du wahrscheinlich 10 verschiedene Klassen sehen, wo drunter sich auch die Klasse „Game“ befindet. Führe nun die Methode „void main(String[] args)“ aus, indem du einen Rechtsklick auf diese Klasse machst. BlueJ sollte jetzt ein Fenster von dem Spiel öffnen.

- Probiere erst einmal die Steuerung aus, indem du dich mit „a, s, w, d“ in alle Richtungen bewegst, mit „space“ springst und mit „q, e“ die Kamera nach Links oder Rechts drehst.
- Nachdem du dich ein wenig mit der Steuerung auskennst versuche nun auf die nächste Plattform und weiter zu gelangen, indem du von einen auf den nächsten Block springst, bis du auf der nächsten Plattform ankommst. Darauf aufpassen, dass du nicht runterfällst brauchst du dabei nicht, da, falls dies passieren solle, du wieder auf der ersten Plattform landest.
- Gehe danach in die Klasse „Game“ und schau sie dir an, damit du ein Verständnis für den Code bekommst schau dabei am besten auch auf die Dokumentation, in der alle Klassen genauestens geschildert sind. Dir sollte auffallen, dass sich bestimmte Zeilen sehr oft in dem Code wiederholen, welche sind:

```
Block block1 = new Block(100, 50, -300, 100, 100, 100, "images/box_white.png");

map.addBlock(block1);
```

Beschreibe nun, mit der Verwendung der Dokumentation, bei einigen dieser Zeilen was in dieser passiert, indem du mit „//“ ein Kommentar hinzufügst. Zum Beispiel: „//Es wird das Objekt X an den Koordinaten x, y, z erstellt ...“

- Ändere nun einige Werte von verschiedenen Blöcken(nach der Aufgabe sie natürlich wieder zurück), sodass als Beispiel einige Blöcke größer sind als davor oder sie nun eine andere Position haben. Du kannst zusätzlich auch versuchen die Textur zu ändern, indem du einen anderen String übergibst.

Wahrscheinlich ist dir auch schon aufgefallen, dass du nicht bis zur dritten Plattform kommst, dies sollte man aber. Daher musst du nun selber an verschiedenen Positionen Blöcke erstellen, um einen Weg zur dritten Plattform zu erstellen.

Aufgabe 2:

- Der erste Block müsste sich an den Koordinaten „x = -1300, y = 900 und z = 0“ befinden und in jede Richtung 100 groß sein. Achte auch darauf, dass du den Block der Map hinzufügst, da er ansonsten nicht geladen wird.

Bei den zweiten Koordinaten „x = -1100, y = 950 und z = -300“ fehlt ein weiterer Block, welcher die gleichen Größen, wie der erste Block hat.

- Der dritte fehlende Block, welcher von der Unterklasse „JumpPad“ sein muss, befindet sich an den Koordinaten „x = -700, y = 1100 und z = -100“ und auch er hat die Größe 100 in alle Richtungen.

Da es aber ein Spezieller Block ist fehlt ihm noch ein Wert (15), welcher die Sprungstärke beschreibt, mit der du Springst, wenn du auf ihm stehst. Wenn du

nicht weißt in welcher Position dieser Wert eingesetzt werden muss schaue einfach wieder in die Dokumentation, wo du auch die Klasse „JumpPad“ finden wirst.

- Da jetzt alle Blöcke existieren schaffst du es jetzt auch zur dritten Plattform, tue dieses anschließend auch. Falls du aus irgendwelchen Gründen es trotzdem nicht schaffen kannst, schaue noch einmal ganz genau auf die Positionen und Größen der Blöcke.

Aufgabe 3:

Auch von der 3. Plattform zur 4. Plattform fehlen Blöcke, dein Auftrag wird es sein, diese ohne Vorgaben zu tun und somit einen Weg zum Ziel zu erstellen.

- Damit du gleich beim Testen deiner neuen Strecke zur 4. Plattform, die da vorigen nicht immer wiederholen musst solltest du deinen Spawn-Punkt (ein Punkt, auf welchem du erscheinst, nachdem du von einer Plattform gefallen bist, oder einen Teleporter berührt hast) zu den Koordinaten „x = -1000, y = 2000 und z = 1500“ umsetzen indem du beim Erstellen des Spielers, am Ende der Klasse „Game“, die ersten drei Koordinaten änderst. Außerdem solltest du noch am Anfang des Codes die Teleportations-Koordinaten des ersten Teleporters zu diesen ändern, damit du ohne Sorge auch einmal von der Plattform springen kannst.
- Nun ist es dir überlassen einen weiteren Weg zur nächsten Plattform zu „bauen“. Damit es aber ein wenig einfacher geht um herauszufinden, wo die Blöcke hinkönnen oder auf welchen Positionen du sie haben willst gibt es zwei Wege um die Positionen herauszufinden. Die erste wäre einfach die beiden Positionen der Plattformen zu verwenden, damit du weißt, zwischen welchen Zahlen sich die Blöcke befinden müssen. Der Mittelpunkt der ersten Plattform ist „x = -1000, y = 1800 und z = 1500“ und der, der Zielplattform ist „x = -2000, y = 2000 und z = 3500“, wobei beide Plattformen 1000 Einheiten in X als auch Z Richtung breit sind. Außerdem ist im Code ein Bereich markiert, indem du

die Blöcke erstellen kannst.

Die zweite Möglichkeit wäre im Spiel, wenn du die Eingabe mit „Enter“ öffnest, dort „/pos“ eingibst und mit einem zweiten Mal „Enter“ drücken die Eingabe bestätigst, wodurch eine Textausgabe in der Konsole erstellt wird, in welcher deine momentane Position steht. Da während der Texteingabe das Spiel pausiert wird kannst du z.B. von einer Plattform springen, „Enter“ drücken und somit dann die Position abfragen.

- Beim Erstellen der Blöcke versuche auch verschiedene Typen mit einzubringen, wie z.B. das „JumpPad“ oder den „Teleporter“ und auch, dass man nicht nur in einer geraden Linie da hinlaufen kann, sondern dass es vielleicht auch schon eine Herausforderung ist, den Parcours zu bewältigen.

- Verändere zum Schluss, wenn du mit Aufgabe b) und c) fertig bist, die Spawn werte nun wieder so zurück, dass man bei der ersten Plattform spant und den Parcours von Anfang an spielen muss.

Zusatzaufgabe:

Um ein wenig Farbe ins Spiel zu bekommen versuche die Klasse Block so zu gestalten, sodass jeder Block, den der Spieler berührt, seine Textur ändert. Damit dies nicht allzu schwer wird, gibt es die Methode „public boolean onPlayerHitBlock(Player player, GLVektor vektor)“, welche jedes Mal vom Spieler aufgerufen wird, wenn er denjenigen Block berührt, außerdem übergibt der Spieler sich selber, wodurch du sehr einfach auf diesen zugreifen kannst.