

# Blocks

Christoph Winklhofer  
MatNr. 0426461 Kennz. 033532 christoph.winklhofer@gmx.at

Thomas Weber  
MatNr. 0526341 Kennz. 033532 weber.t@gmx.at

2. Mai 2007

## 1 Zum Spiel

Blocks ist ein 3D Plattformer mit Puzzle-Elementen. Der Spieler steuert eine Spielfigur durch eine Welt, die vollständig aus Holzbausteinen besteht. Dabei hat er die Aufgabe Murmeln aufzusammeln und den Ausgang des jeweiligen Levels zu erreichen.

Die Spielfigur kann frei durch die Welt laufen und springen. Außerdem kann sie Blöcke bis zu einer gewissen Größe schieben und ziehen.

Die Blöcke, aus denen die Welt besteht, folgen einer eigenen, vereinfachten, Physik. Im Ruhezustand sind sie immer an einem Raster ausgerichtet. Ausserdem lassen sie sich nur entlang den Achsen verschieben und sind nicht drehbar.

### 1.1 Steuerung

Die Spielfigur wird mit einem Game-Controller gesteuert. Idealerweise wird dabei ein XBox 360 Gamepad verwendet.

- Mit dem linken Control-Stick wird die Spielfigur bewegt. Die Richtung ist abhängig von der Kamera. So läuft die Spielfigur z.B. immer in Blickrichtung der Kamera, wenn der Spieler den Stick vorwärts bewegt.
- Mit dem rechten Control-Stick wird die Kamera gesteuert. Mit vorwärts/rückwärts kann man die Distanz der Kamera zum Spieler verändern. Mit links/rechts rotiert man die Kamera entlang der Y-Achse um den Spieler.
- Beim Betätigen von A springt die Spielfigur.
- Mit B werden Blöcke verschoben.

#### 1.1.1 Springen

Wenn die Spielfigur Bodenkontakt hat und A betätigt wird springt sie nach oben. Dabei behält sie die Geschwindigkeit, die sie bereits hat. Die Spielfigur ist in der Luft weiterhin, jedoch eingeschränkter, mit dem linken Control-Stick steuerbar.

Neben dem normalen Sprung beherrscht die Spielfigur noch einen „Wall-Jump“. Dabei kann sich die Spielfigur, wenn sie mit genug Anlauf gegen eine Wand springt, durch rechtzeitiges Betätigen von A erneut von der Wand abstoßen. Zwischen zwei Wänden lässt sich das unbeschränkt oft wiederholen. So können sonst schwer erreichbare Bereiche betreten werden. Im Moment wird der richtige Zeitpunkt zum erneuten Betätigen von A dadurch angezeigt, dass die „Spielfigur“ rot wird.

### 1.1.2 Schieben

Wenn die Spielfigur direkt vor einem Block steht und B betätigt wird wechselt sie in den Schiebe-Modus. Das ist im Moment daran erkennbar, dass sie gelb wird. Wenn man nun den linken Control-Stick in die entsprechende Richtung drückt, und der Block nicht zu groß ist, wird ihn die Spielfigur in die entsprechende Richtung schieben oder ziehen.

### 1.1.3 Tastatur-Befehle

- Mit *ESC* kann man das Spiel beenden.
- Mit *F1* kann man sich die Szene in Wireframe-Mode ansehen.
- Mit *F5* kann man ein Level neu starten.

### 1.1.4 Kommandozeilen-Parameter

Blocks bietet die Möglichkeit, die Anzeige-Einstellungen und das Start-Level als Kommandozeilen-Parameter mitzugeben.

> `Blocks.exe [levelname [width height [0|1]]]`

- levelname: Der Name des Levels, in dem man starten möchte. Optionen sind „Start“, „Level Two“, „Tutorial“ und „Castle“. Standard ist „Start“.
- width/height: Gibt die Fenstergröße an. Standard ist 1024x768.
- 1|0: Soll Fullscreen verwendet werden? 0 = Nein, 1 = Ja. Standard ist Ja.

## 2 Implementierung

### 2.1 Kollisionserkennung

Die Kollisionsgeometrie der Spielfigur wird durch drei Kugeln beschrieben. Genauer sind das eine Kugel für den Kopf, eine größere Kugel für den Bauch und eine Kugel für die Füße. Außerdem gibt es eine „Sensor-Kugel“, die dazu benutzt wird, Blöcke direkt vor der Spielfigur zu finden, die aber nicht für die Kollisionserkennung benutzt wird.

Abhängig von den Kollisionspartnern verwenden wir unterschiedliche Methoden der Kollisionserkennung.

- Bewegte Blöcke erkennen Kollisionen im Vorhinein, indem sie vor jedem Wechsel von einer Octree-Zelle zur nächsten überprüfen, ob diese nicht schon von einem anderen Block besetzt ist.

- Kollisionen des Spielers mit einfachen Objekten, wie dem Ziel oder Murmeln, werden durch einfache Kugel/Kugel Kollisionsüberprüfung bestimmt.
- Kollisionen zwischen der Spielfigur und Blöcken werden mit Kugel/Polyeder Kollisionsüberprüfung erkannt. Dabei wird jeder Block durch einen konvexen Polyeder repräsentiert, der aus einer Menge von Ebenen besteht, die sein Äußeres beschreiben. Vereinfacht beschrieben wird bei der Kollisionserkennung überprüft, ob der Abstand des Kugel-Mittelpunktes von jeder Teil-Ebene des Polyeders kleiner als ihr Radius ist. Da es ineffizient wäre, diese Überprüfung für jeden Block zu machen, werden nur die Blöcke, die innerhalb der Bounding-Box der Spielfigur sind, aus dem Octree abgefragt und getestet. Jeder dieser Blöcke wird nun mit jeder Teilkugel der Spielfigur auf eine Kollision überprüft.

## 2.2 Kamera und Steuerung

Der Spieler wird von einer 3<sup>rd</sup> Person Kamera verfolgt, die entlang der Y-Achse frei um den Spieler drehbar ist. Mit dem rechten Control-Stick lässt sich die Kamera um die Spielfigur rotieren und hinein-, und hinaus-zoomen. Die Richtung, in die die Spielfigur läuft, wenn der linke Control-Stick betätigt wird, ist abhängig von der Kamera. Dazu bietet die PlayerCamera-Klasse, die für die Spielfigur verwendet wird, eine `projectInput(V2f vector)` Methode, die den Input des Spielers in einen Raum-Vektor umwandelt.

Wir haben das Verhalten der Spielfigur als State-Machine modelliert. Das heißt, dass verschiedene Klassen für verschiedene Zustände der Spielfigur zuständig sind. Wenn die Spielfigur läuft, befindet sie sich so zum Beispiel im Zustand `PlayerRunning`. Sollte der Spieler jetzt den A Knopf zum Springen drücken oder von einem Block laufen, wird `PlayerRunning` eine Instanz von `PlayerFalling` erzeugen, die sie als aktiven Zustand ablöst.

Mit diesem Modell lassen sich sehr schnell zusätzliche Bewegungsfolgen anfügen. So beherrscht die Spielfigur zum Beispiel einen Wall-Jump, der ausgelöst wird, wenn die Spielfigur mit ausreichend Anlauf gegen eine Wand springt und erneut A drückt.

## 2.3 Bewegte Objekte

Neben der Spielfigur ist jeder Block beweglich, wenn er mit ausreichend Kraft angeschoben wird. Die Spielfigur hat standardmäßig eine Kraft von 64. Das bedeutet, dass sie einen jeden Block, der ein geringeres Volumen als 64 hat, bewegen kann. Außerdem gibt es skriptbare Action-Blöcke, die sich von Alleine bewegen, und auch selbst andere Blöcke verschieben können. Diese Blöcke können auch auf den Kontakt mit der Spielfigur warten und Signale an andere Blöcke schicken. So lassen sich auch komplexere Schalter-Systeme realisieren. Für die Action-Blöcke haben wir eine einfache Befehls-Sprache entwickelt, die die Blöcke nach und nach aus einer Liste abarbeiten.

## 2.4 Texture Mapping von Blöcken

Die einzigen Objekte, die bis jetzt mit einer Textur überzogen wurden, sind die Blöcke. Da ein Block eine beliebige Form und Größe haben kann, werden alle

Vertex-, und Textur-Koordinaten der Blöcke prozedural erzeugt. Ein Block ist ein normaler Quader mit abgerundeten Kanten. Zur Texturierung wird die Oberfläche in 3 Teil-Flächen unterteilt, die einzeln texturiert werden. Das sind die vordere und hintere Seite und eine Schleife um den Block herum. Für diese Teile lassen sich dann einfach korrekte Textur-Koordinaten berechnen, indem man sie plättet, und ihre 2D-Koordinaten bestimmt. Zur Texturierung verwenden wir eine Ahorn-Textur, die mit `GL_MIRRORED_REPEAT` wiederholt wird.

So lässt sich ein beliebig größer und geformter Block unverzerrt und überzeugend mit einer Holz-Textur überziehen.

## 2.5 Beleuchtung

Zur Zeit verwenden wir lediglich eine einzige parallele Lichtquelle. Die Blöcke und Murmeln haben ein einfaches Material mit Specular Highlight.

## 2.6 Externe Bibliotheken

- glfw: OpenGL Framework für die Erzeugung eines Anzeigefensters und Abfragen der Eingaben.  
<http://glfw.sourceforge.net/>
- GLee: Automatisches Laden von OpenGL-Extensions.  
<http://elf-stone.com/glee.php>
- IlmIf: Lesen von OpenEXR (HDR Dateiformat) Dateien  
<http://www.openexr.com>
- Imath: Mathematik Bibliothek  
<http://www.openexr.com>
- tinyxml: Parsen von XML Dateien  
<http://www.grinninglizard.com/tinyxml>

## 3 Das Tutorial-Level

Das Tutorial Level ist eine Einführung in die Steuerung von Blocks. Die einzelnen Plattformen müssen nicht absolviert werden. Weiters muss der Ausgang auch nicht erst durch einsammeln von Murmeln freigeschalten werden sondern ist von Beginn an aktiviert.

- Auf der ersten Plattform links kann der Spieler einen Geschicklichkeitsparcour durchlaufen. Die Spielfigur wird mit dem linken Analogstick bewegt und springt mit dem A Button. Als Belohnung erhält der Spieler eine Murmel.
- Auf der zweiten Plattform rechts sind einfache Schieberätsel zu lösen. Zum schieben eines Blocks muss die blaue Koordinatenachse der Spielfigur den Block berühren. Durch anschließendes halten des B Button (Spielfigur wird gelb) kann der Block mit dem linken Analogstick bewegt werden. Die Murmel befindet sich im Inneren des oberen Gebäudes deren zwei Eingänge mit Blöcken verschlossen sind. Der linke Block ist zu groß und

kann nicht bewegt werden. Durch herausziehen des rechten Blocks gelangt der Spieler ins Gebäude.

- Die dritte Murmel erreicht der Spieler indem er einen Walljump anwendet. Dazu stellt man sich in den Spalt der beiden Blöcke und springt mit Anlauf auf eine Wand. Zum richtigen Zeitpunkt (Spielfigur wird rot) muss der Spieler nun auf die gegenüberliegende Wand springen. Dies wiederholt sich bis der Spieler oben auf dem Block angekommen ist.
- Den Ausgang erreicht man, in dem die Spielfigur auf den roten Schalter am Ende des Weges springt. Dadurch werden die Transportblöcke eingeschalten über die man den Ausgang erreicht und ins Burg Level gelangt.

## 4 SPOILER: Walkthrough durch Burg-Level

Das hier ist eine Schritt für Schritt Anweisung zum Lösen des großen Burg-Levels.

Erklärung: Richtungsangaben wie Links, Rechts, Vorne, hinten etc. sind alle von der Blickrichtung der Spielfigur zu Beginn des Levels aus angegeben.

- Laufe durch das Burgtor.
- Hier sollte Dir der eine Block in der Wand des großen Turmes auffallen, der etwas kleiner ist als die anderen und einen anderen Farbton hat. Ziehe ihn heraus, und hole Dir die blaue Murmel, die im Raum dahinter versteckt ist.
- Schiebe den Block jetzt nach rechts in die Lücke bei der Treppe.
- Statt die Stiege hinauf zu laufen, laufe nach vorne in den Garten hinter dem großen Turm.
- Entspann' Dich ein wenig beim Springbrunnen. ;)
- Bei der Wand hinter der linken Treppe im Garten ist ein Spalt, der zu einem Gang unter der Burgmauer führt. Folge dem bis zum Ende.
- Hier sind vier kleine Blöcke und ein großer Träger, der den kleinen Turm in der hinteren Burgmauer stützt.
- Schiebe den Träger von der Plattform. Der Turm wird nun herunterstürzen und auf den kleinen Blöcken aufliegen, ohne Dich zu erschlagen.
- Jetzt ist es Zeit auf die Burgmauer zu gehen. Lauf dazu die Treppe hinauf, die Du im dritten Schritt vervollständigt hast.
- Laufe jetzt durch den Großen Turm durch und auf den Eckturm hinten links.
- Spring jetzt vom Turm auf die hintere Burgmauer und laufe über den, nun um 4 Blöcke verkleinerten Turm, auf den Eckturm hinten rechts. Hier findest Du die grüne Murmel.

- Lauf jetzt wieder auf den Eckturm hinten links und spring auf die bewegliche Plattform, die sich zwischen den beiden Ecktürmen links hin und her bewegt.
- Von der Plattform kommst Du zum vorderen linken Eckturm, auf dem Du einen roten Block findest.
- Dieser Block ist ein Schalter. Wenn Du auf ihn springst, werden die Zinnen der vorderen Burgmauer nach oben schweben und einen Pfad zum Eckturm rechts vorne formen.
- Tu' das und hol Dir die rote Murmel.
- Jetzt hast Du alle drei Murmeln, die Du benötigst, um das Level abzuschließen.
- Laufen nun in den großen Turm zum Thron und springe auf ihn.
- Der Thron wird nun mit Dir nach oben fahren.
- Hier musst Du jetzt von Plattform zu Plattform springen, bis Du ganz oben am Turm angelangt bist.
- Hier ist das Ziel, dass Dich wieder an den Start dieser Demo zurück bringt.