

# SUPER BLOBIO 64

---

Maximilian Flack

Das Spiel soll ähnlich wie Mario 64 ein 3d Jump and Run Platformer werden.  
Der Spieler spielt Blobio. Gesteuert wird mit WASD zum Laufen und Leertaste zum Springen.  
Wenn man mehrmals hintereinander springt, kann Blobio bis zu einem dreifach Sprung ausführen.

Man läuft in einer 50 Meter großen Welt herum. Die Weltabgrenzung ist eine Hecke.

Das Ziel des Spiels ist es, einen Stern einzusammeln.

Dieser erscheint, wenn man eine bestimmte Anzahl Münzen eingesammelt hat.

Blobio hat eine festgelegte Anzahl Leben und verliert eines davon, wenn er einen Gegner berührt oder herunterfällt.

Diese Gegner sind Blockies. Es gibt in der Welt eine festgelegte Anzahl Blockies, die zufällig in der Welt verteilt sind.

Die Blockies können sterben, wenn Blobio auf sie springt.

Nr	Criterion	Explanation
1	Units and Positions	<p>0 ist die Mitte der Map und der Punkt, wo der Spieler Charakter spawnnt, jedoch ein bisschen nach oben verschoben, damit er nicht in den Boden clippt. So wird die gesamte Map und alle Entities und Collectibles um den Ursprung und somit den Spieler herum erstellt.</p> <p>1 ist das Equivalent zu einem Meter. Der Spieler Charakter ist 2 Meter hoch, die Gegner einen Meter und die Münzen einen halben Meter. Die Map ist 60 Meter lang und breit und die Hitboxen der Mauern 10 Meter hoch. Das ist so gewählt, weil die Größe der Map so ausreichend viel Platz bietet und die Höhe der Mauern, damit der Spieler sie nicht überspringen kann.</p>
2	Hierarchy	<ul style="list-style-type: none"> <li>-Game</li> <li>--Environment</li> <li>---Ground</li> <li>---Skybox</li> <li>---Borders</li> <li>----BorderNorth</li> <li>----... (Erstellt je eine Border für jede Himmelsrichtung, und zwei im Norden, weil die Tür hier zwischen den Bordern ist)</li> <li>---StarPlatform</li> <li>----Door</li> <li>----Star</li> <li>----Platforms</li> <li>-----Platform</li> <li>-----...(Mehrere Plattformen zum drauf springen)</li> <li>--Avatar</li> <li>---Body</li> <li>----...(Mehrere Bodyparts mit Meshes, um das Aussehen des Charakters anzupassen)</li> <li>---Camera</li> <li>---Sounds</li> <li>--Entities</li> <li>---Enemies</li> <li>--Collectibles</li> <li>---Coins</li> <li>--Sound</li> </ul> <p>Die Nodes „Enemies“ und „Coins“ sind leer, weil sie später mit einer vorher festgelegten Anzahl an Nodes dynamisch im Code generiert werden. So lässt sich der Überblick</p>

		<p>behalten, was wo im Graphen vorhanden ist, auch wenn es über Code erstellt wird.</p> <p>Dadurch, dass die Kamera und die Sounds am Avatar-Knoten hängen, bewegt sich die Kamera mit dem Avatar mit und Sounds werden direkt beim Avatar abgespielt. Alles im Knoten „Sound“, welcher ein direkter Nachfahre des „Game“ Knotens ist, wird später dynamisch über Code an andere Knoten verteilt.</p>
<b>3</b>	Editor	<p>Alles, was nicht dynamisch über Code generiert wird, wird im Editor erstellt. Das bringt den Vorteil, dass man direkt sieht, was an den Meshes verändert wird und wo ihre Position in der Welt ist. Der Avatar wird im Editor erstellt, ebenso wie alle Environment-Objekte. Auch die Soundeffekte und die Musik wird über den Editor importiert. Außerdem werden bestimmte Komponenten, wie Licht, Kamera, Audio und Rigidbodies im Editor erstellt. Wenn ein Knoten eine dynamische Anzahl Kindknoten haben soll, ist es sinnvoller, diese über Code zu erstellen und dann in den Knoten einzufügen, das wird bei „Enemies“ und „Coins“ gemacht. Hier werden alle Komponenten per Code erstellt und eingefügt.</p>
<b>4</b>	Scriptcomponents	<p>Die ScriptComponent „EnemyMovement“ wird dynamisch zu jedem erstellten Enemy hinzugefügt. Sie bewirkt, dass auf den RigidBody des Enemy eine Kraft gewirkt wird, die ihn in einer Kreisähnlichen Bewegung laufen lässt. Das als Script-Komponente zu implementieren war in diesem Fall sehr nützlich, da man dadurch ganz einfach die Komponente hinzufügen und auch wieder entfernen kann, je nachdem ob der Enemy gerade laufen oder stehen soll.</p>
<b>5</b>	Extend	<p>Die Klassen Avatar, Coin und Enemy extenden <i>f.Node</i>. Das ist vor Allem bei Coin und Enemy nützlich, da dadurch je nach Anzahl der Coins und Enemies ganz einfach neue Klassen, also neue Nodes, des jeweiligen Typen entstehen können und diese dann anderen Nodes zugeordnet werden können. Außerdem extendet EnemyMovement <i>f.ComponentScript</i>, EnemyStateMachine <i>f.Aid.ComponentStateMachine</i> und GameState <i>f.Mutable</i>.</p>
<b>6</b>	Sound	<p>Der Spieler macht ein Geräusch beim Springen. Es ertönt ein Geräusch, wenn man eine Münze einsammelt. Die Gegner machen durchgehends Geräusche. Ebenso kommen durchgehend Umgebungsgeräusche. Wenn man alle Münzen eingesammelt hat, ertönt eine Musik und die Umgebungsgeräusche verstummen.</p>

		<p>Wenn man den Stern einsammelt, verstummt alles andere und es ertönt eine Melodie.</p> <p>Die Komponente „AudioListener“ sitzt auf dem Knoten „Avatar“. Das ist sinnvoll, weil somit der Spieler alles in Bezug darauf hört, wo sich der Avatar gerade befindet. Alle Sounds, außer die Geräusche der Gegner befinden sich ebenfalls auf dem Avatar-Knoten, weil bei allen Geräuschen, also bei Springen, Münzen einsammeln, Umgebungsgeräusche, Alle-Münzen-Lied und der Sternmelodie hört man die Geräusche direkt. Nur bei den Gegnern werden die Sounds dynamisch jedem Gegner zugeteilt, damit sie immer lauter werden, je näher man dem Gegner ist.</p>
7	VUI	<p>Das User-Interface besteht aus zwei Elementen. Eines besteht aus dem Symbol von einem Herz, zusammen mit der Anzahl der verbleibenden Leben des Spielers. Das andere ist eine Münze, zusammen mit den bereits eingesammelten Münzen und der gesamten Anzahl an Münzen im Spiel. Wenn der Spieler getroffen wird oder von der Plattform fällt, wird die Anzahl Zahl neben dem Herz um eins heruntergesetzt. Ebenso wenn man eine Münze einsammelt, wird die Zahl der Münzen um eins hochgesetzt.</p>
8	Event-System	<p>Die verschiedenen Events sind playerHit, allCoinsCollected und endGame. playerHit bewirkt, dass die Anzahl der Leben um eins heruntergesetzt wird. Außerdem wird der Spieler wieder an den Anfangsspawnpoint zurückgesetzt und bekommt 2 Sekunden Immunität. Dieses Event wird sowohl von einer Kollision mit einem Gegner, als auch durch aus der Welt fallen ausgelöst. allCoinsCollected wird ausgelöst, wenn man alle Münzen eingesammelt hat. Dadurch wird das Tor geöffnet und der Weg zum Stern wird frei und es wird eine Melodie abgespielt. EndGame wird aufgerufen, wenn der Spieler keine Leben mehr hat, und wenn er den Stern berührt. Es werden bei beiden Möglichkeiten verschiedene Parameter mitgegeben, womit unterschieden werden kann, ob der Todesscreen oder der Gewinnscreen angezeigt werden sollen.</p> <p>Das Event-System ist nützlich, weil dadurch Methoden an unterschiedlicher Stelle und zu unterschiedlichen Zeitpunkten im Code aufgerufen werden können.</p>

9	External Data	<p>Die Parameter, die man in der config.json Datei anpassen kann, sind liveCount, enemyCount, coinCount und playerSpeed.</p> <p>LiveCount verändert die Anzahl der Leben, die der Spieler zu Beginn hat.</p> <p>EnemyCount verändert die Anzahl der Gegner, die zu Beginn erstellt und in die Welt gesetzt werden.</p> <p>CoinCount verändert die Anzahl der Münzen, die gesammelt werden müssen, damit sich das Tor zum Stern öffnet.</p> <p>PlayerSpeed verändert die Geschwindigkeit des Spieler-Charakters.</p> <p>All diese Parameter wurden gewählt, um die Schwierigkeit des Spiels anzupassen.</p>
A	Light	<p>Dem Node Environment wurde ein ComponentLight hinzugefügt, dass direktionales Licht erstellt. Dieses Licht kommt von schräg oben und soll dadurch eine Sonne imitieren. Das wurde so gemacht, damit die Objekte von einer Seite angeleuchtet werden und an der anderen Seite einen Schatten haben, um so Tiefe bei den einzelnen Objekten zu erzeugen. Außerdem hat der Stern ebenfalls ein ComponentLight, welches ein Pointlight erstellt, damit um den Stern herum ein Licht scheint. Der Stern hat ebenfalls einen beleuchteten Shader als Material, um sich von der Welt abzuheben und zu leuchten.</p> <p>Die Skybox hat ebenfalls einen beleuchteten Shader, damit diese überall gleichmäßig beleuchtet ist.</p> <p>Alles andere hat einen unbeleuchteten Shader welcher sich vor Allem von der Sonne beleuchten lässt.</p>
B	Physics	<p>Der Spielercharakter hat einen RigidBody, in Quaderform, um seinen Torso herum. Arme, Beine, Kopf und Hut sind nicht oder nur teilweise vom RigidBody umschlossen.</p> <p>Außerdem haben alle Environment-Objekte einen RigidBody, damit der Spieler nicht durch sie durchlaufen oder fallen kann. Bei dem Boden und den Plattformen werden zusätzlich noch die Kollisionen mit dem Spieler registriert, um ihm das Springen wieder zu erlauben, da das weitere Springen, wenn man nach einem Sprung in der Luft ist, verboten wird. Außerdem haben alle Enemies und alle Coins Rigidbodies. Bei den Coins wird die Kollision mit dem Spieler registriert und wenn es soweit ist, wird die Münze eingesammelt, also sie verschwindet und der Münzzähler geht eins hoch. Bei den Enemies wird zwischen zwei Kollisionen unterschieden mithilfe des</p>

		<p>CollisionPoints. Wenn dieser unterhalb eines bestimmten Punkts beim Player liegt, also der Gegner den Spieler am Körper trifft, wird das Event „playerHit“ ausgelöst und wenn der Punkt darunter liegt, also bei den Füßen des Spielers, stirbt der Gegner, also verschwindet aus der Welt. Der Stern hat ebenfalls einen Rigidbody, wenn man mit diesem kollidiert, dann wird das Event „endGame“ abgeschickt mit dem Gewinn als Parameter.</p>
<b>C</b>	Net	-
<b>D</b>	State Machines	<p>Die Gegner haben alle eine StateMachineComponent. Diese bewirkt, dass die Gegner drei verschiedene States haben. Einmal gibt es den State „PATROL“, welcher der normale State ist, wenn der Spieler mehr als 10 Meter vom Gegner entfernt ist. Dieser bewirkt, dass den Gegnern die Komponente „EnemyMovement“ zugewiesen wird, also dass diese sich in einem Kreis bewegen.</p> <p>Dann gibt es den State „LOOK“, wenn der Spieler zwischen 5 und 10 Meter entfernt ist. Dabei wird die Komponente „EnemyMovement“ entfernt, also bleiben die Gegner stehen und werden zu Beginn des States in Richtung Spieler gedreht. Der letzte State ist „ATTACK“, welcher bewirkt, dass die Gegner auf den Spieler zulaufen. Dieser wird ausgelöst, wenn der Spieler weniger als 5 Meter entfernt ist.</p>
<b>E</b>	Animation	<p>Die Münzen bekommen, wenn sie generiert werden, eine Animationskomponente zugewiesen. Diese bewirkt, dass sich die Münzen in einer Loop ständig um 360 Grad um die Y-Achse drehen. Dies soll bewirken, dass die Münzen nicht einfach wie starre Objekte in der Welt aussehen, sondern dynamisch und so, als könnte man mit ihnen interagieren. Außerdem hat die Tür eine Animation, die abgespielt wird, wenn man alle Münzen eingesammelt hat. Sie wird dabei um 5 Meter auf der Z-Achse verschoben, um den Durchgang zum Stern freizugeben.</p>