



Nr	Bezeichnung	Inhalt
	Titel	Das schräge Murmel-Labyrinth
	Name	Viktor Lukanow
	Matrikelnummer	263613
1	Nutzerinteraktion	Neigung der Plattform: A/D -> leichte Rotation um Z-Achse, W/S -> leichte Rotation um X-Achse; Level-Auswahl über User Interface

2	Objektinteraktion	<p>Bei Kollision der Kugel mit den Barrieren ertönt ein Sound. Kollidiert eine Barriere ein erstes mal mit der Kugel färbt sie sich gelb und erhöht die Restitution, federt den Ball nun also mehr ab. Bei der zweiten Kollision färbt sich die Barriere rot und erhöht die Restitution noch einmal um das doppelte.</p>
3	Objektanzahl variabel	<p>Level 1-3 erzeugen und entfernen unterschiedliche Barrieren und deren Rigidbodies. Die Erzeugung geschieht unmittelbar bei Änderung des Werts im User Interface. Des Weiteren wird nach erfolgreichem Abschluss jedes Levels eine neue Kugel erzeugt um das nächste Spiel spielen zu können. Dies geschieht nach Auswahl des nächsten Levels in der Success-Message.</p>

4	Szenenhierarchie	<p>Der Graph <i>root</i> hat zwei Kinder: <i>environment</i> (alle statischen Elemente) und <i>moveables</i> (alle dynamischen Elemente, in diesem Fall die Kugel). In <i>environment</i> befindet sich die Node <i>basicFloor</i> (für den Boden der Plattform). Deren Kinderknoten heißen <i>barriers</i> (die dort befindlichen Kinderknoten stellen die äußere Barriere der Plattform dar), <i>level1</i>, <i>level2</i> und <i>level3</i> (Die Kindernoten dieser „Level“-Nodes stellen die inneren Barrieren, also das Labyrinth auf der Plattform dar. Sie sind im Spiel niemals zusammen zu sehen und werden einzeln aktiviert/deaktiviert, je nach Level-Auswahl)</p>
5	Sound	<p>Bei Kollision der Kugel mit den Barrieren ertönt ein Sound. Bei erfolgreichem Abschluss eines Levels und der Entscheidung weiterzuspielen ebenfalls.</p>
6	GUI	<p>Wahl zwischen Level 1-3 und dadurch unterschiedlichen Labyrinth-Szenarien.</p>

7	Externe Daten	In der Config.json hat der Nutzer die Möglichkeit bestimmte Parameter, die das Verhalten der Kugel beeinflussen einzustellen. In diesem Fall sind das: <i>Mass</i> (das Gewicht der Kugel), <i>Restitution</i> (die Stärke der Federung bzw. wie stark die Kugel von Gegenständen abprallt) und <i>Friction</i> (die Reibung der Kugel auf Oberflächen). Die gewählten Einstellungen können im Spiel rechts oben im Viewport eingesehen werden.
8	Verhaltensklassen	Ball-Klasse: Dynamische SphereNode in der Größe 1x1x1 mit rotem Material. Wird bei Erstellung am Punkt (2, 5, 2) spawned. Mass, restitution und friction sind variable Parameter, die in der Config.json abgelegt werden. Gui-Klasse: Graphisches User-Interface mit Input-Feld, bei dem die Werte 1-3 vom User ausgewählt werden können.
9	Subklassen	Ball extends f.Node; GameState extends f.Mutable

10	Maße & Positionen	Kugel: 1x1x1, davon ausgehend das Labyrinth ca. 20x20 groß. Barrieren sind dreimal so hoch wie der Ball, um ein verlassen der Plattform/des Labyrinths zu vermeiden.
11	Event-System	Die init Funktion wird durch „load“-Event aufgerufen; interactive Viewport wird gestartet, sobald auf das Dialog-Fenster geklickt wurde. Die update-Methode wird mit einem Loop-EventListener aufgerufen; Die Kollisionsprüfung zwischen Kugel und Barriere geschieht mit einem EventListener; In der Success-Message hat der Nutzer die Möglichkeit ein Level auszuwählen, was ebenfalls durch einen EventListener geschieht, der bei Klick auf das entsprechende HTML-Element feuert.