**Dokumentation**

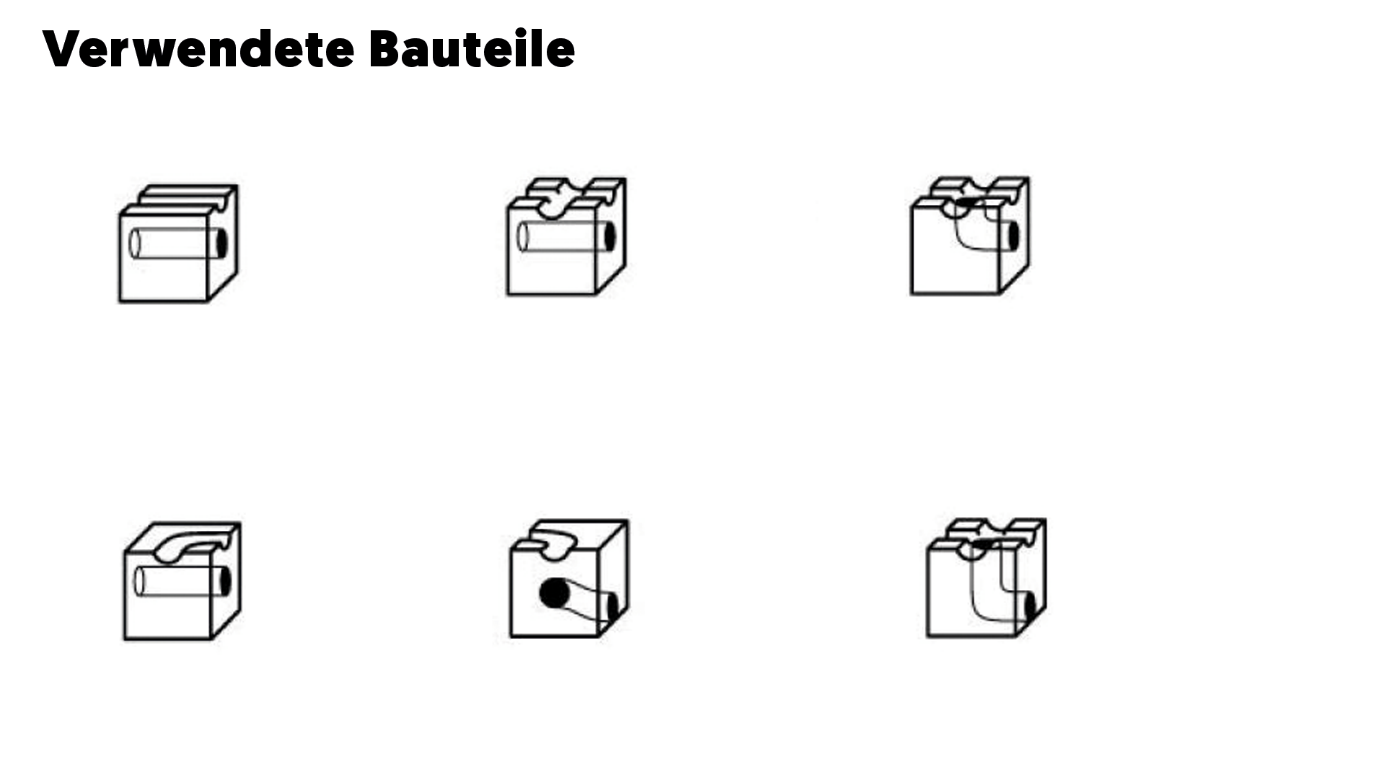
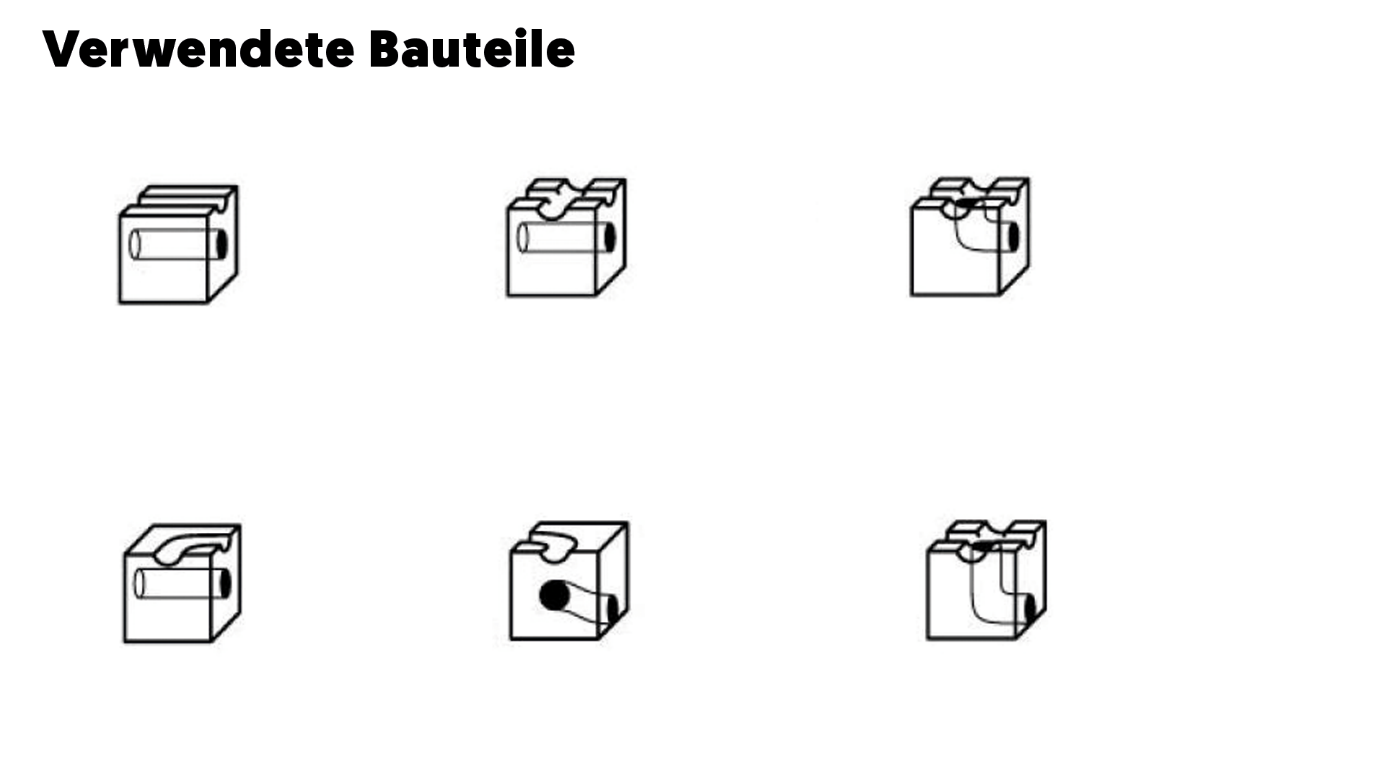
LHM

# Einleitung

Wir sind die Projektgruppe LHM und bestehen aus den Teammitgliedern Leon Wessel, Michael Lopes Marques und Henri Katzenberg.

Die Aufgabe des Projektes war die Simulation einer Kugelbahn. Dafür sollten Elemente aus dem gegebenen Cuboro Set verwendet werden.  
Außerdem soll eine passende Graphische Oberfläche, die die Simulation anhalten, stoppen und auch beschleunigen oder verlangsamen kann, implementiert werden.

# Kugelbahn und ausgewählte Elemente



(6)

(5)

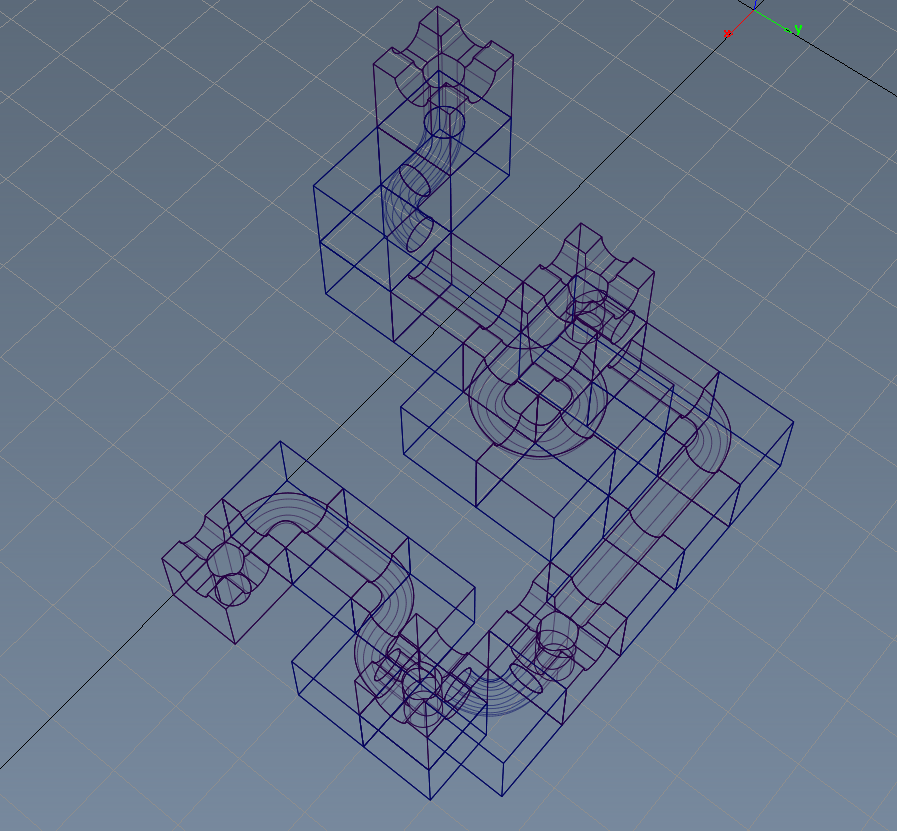
(4)

(3)

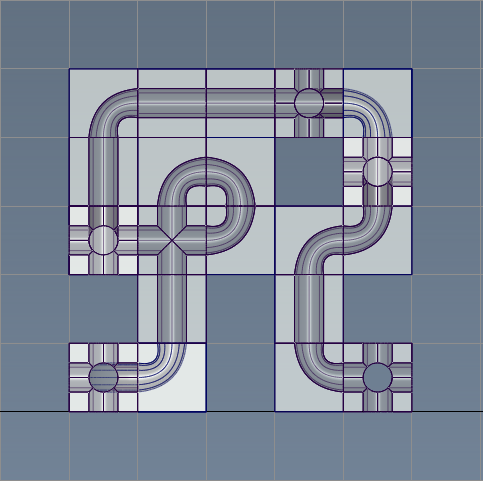
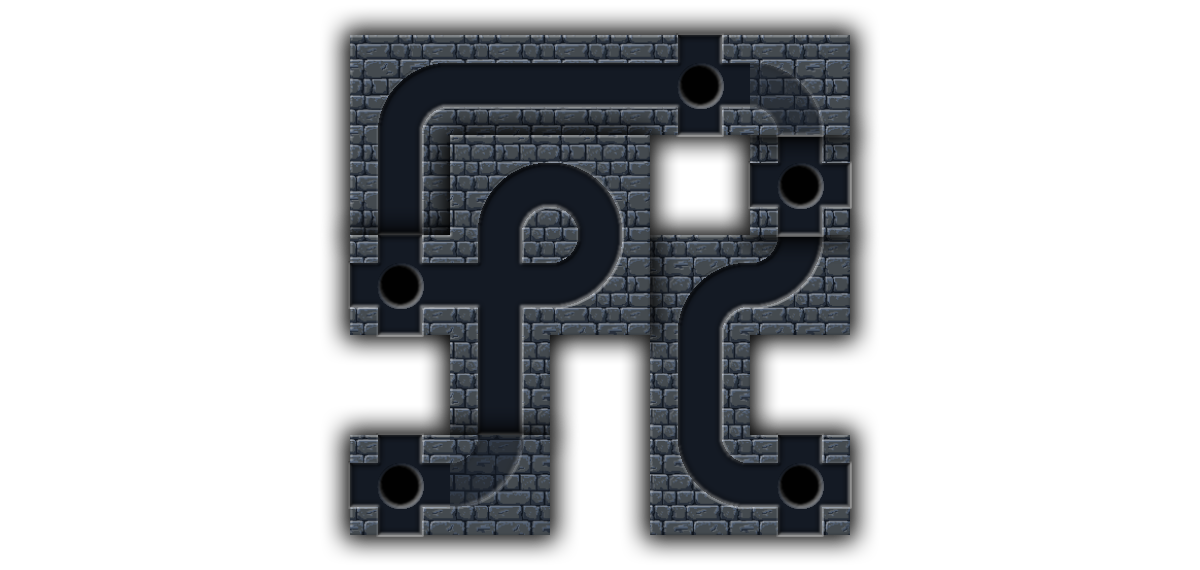
(2)

(1)

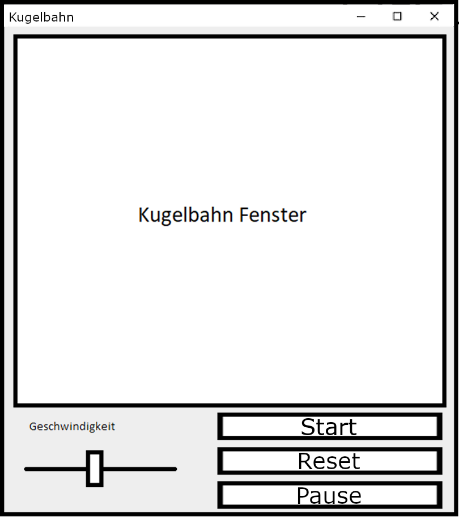
Für die Kugelbahn haben wir essentielle Elemente, wie das Geradenstück (1) und die normale Kurve (4) gewählt. Um die Bahn etwas interessanter zu gestalten benutzen wir auch eine Kreuzung (2), sowie zwei Elemente, die die Kugel unterschiedlich weit herunterfallen lassen (3&6). Da sich die Kugel nach dem Fall im inneren eines Elementes befindet brauchen wir ebenfalls eine Kurve im Inneren (5), die die Beiden unterschiedlichen Höhen verbinden kann.

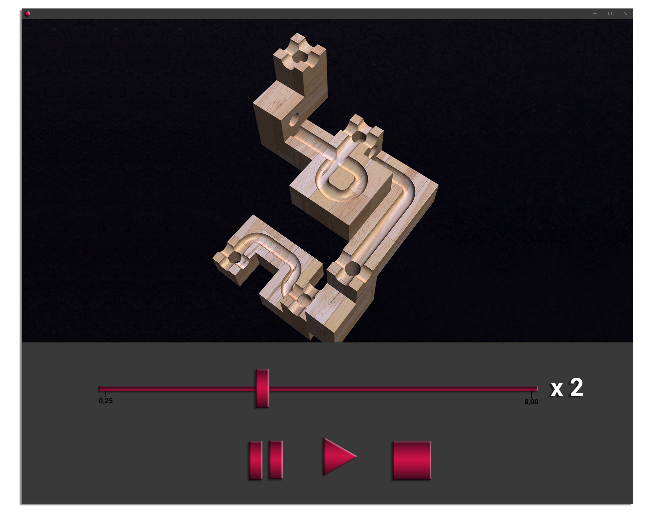
 

Die Kugelbahn befinden sich in einem 5x5x5 Raster und wurde zunächst in Autodesk Alias modelliert um uns einen Überblick zu verschaffen. Wir haben uns letzten Endes für eine 2D-Ansicht entschieden.

# Konzeptentwürfe

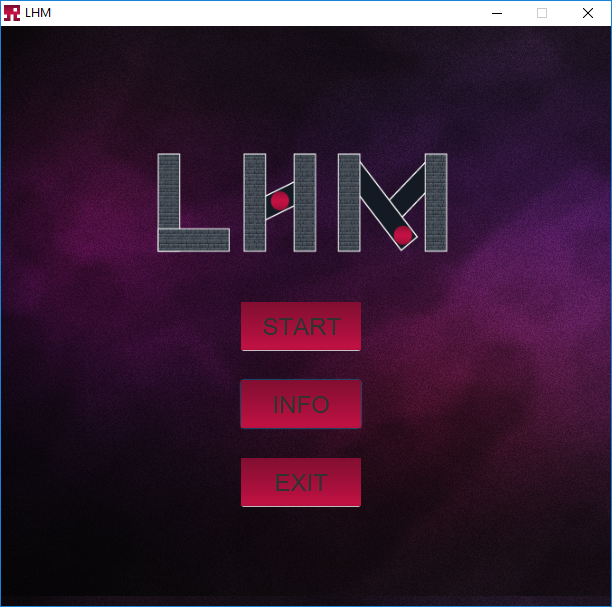
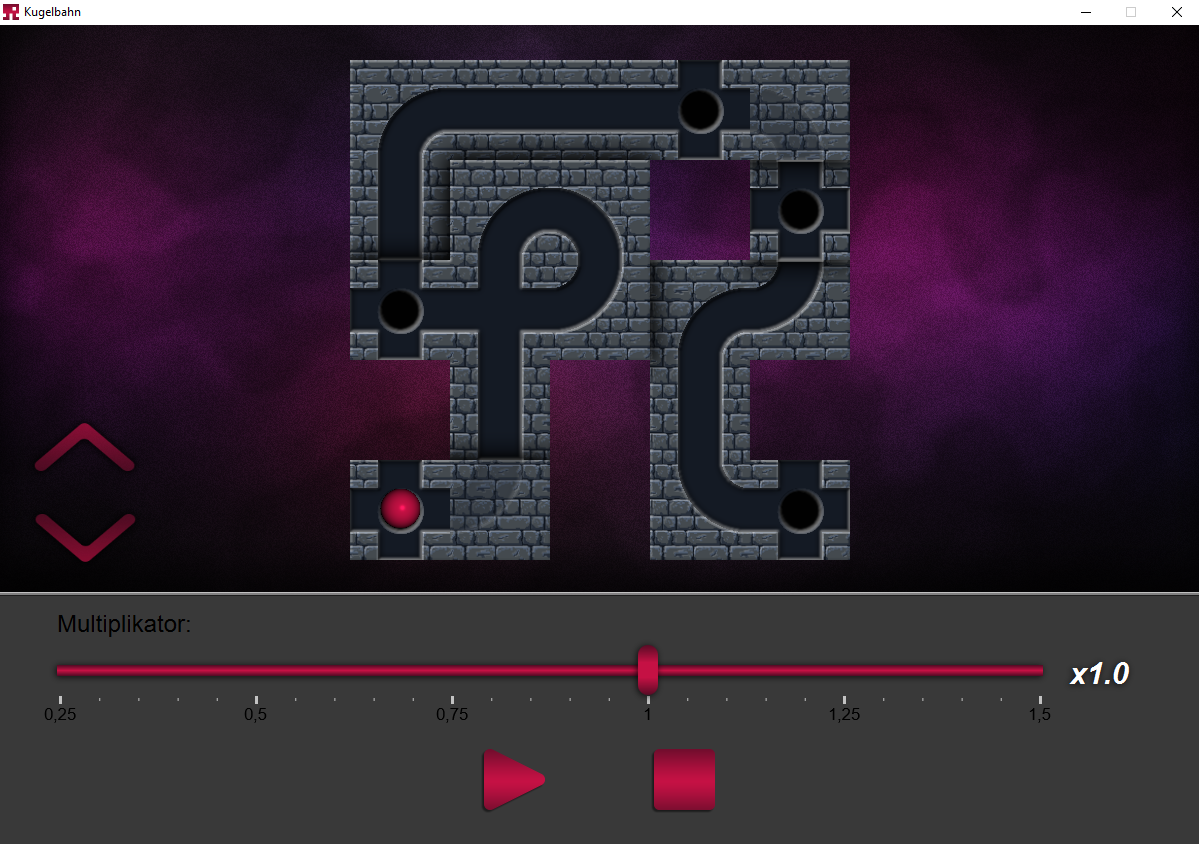
Die Graphische Benutzeroberfläche unseres Projektes sollte grundlegende Funktionen, wie Start, Stopp und Reset der Simulation beinhalten. Weitere Anforderungen waren ein Geschwindigkeitsregler und eine Geschwindigkeitsanzeige. Zunächst hielten wir diese in einem sehr simplen Konzept fest.



Wir waren uns beim ersten Entwurf bereits einig, dass sich alle Funktionen unter dem Fenster, in dem die Bahn visualisiert wird, befinden sollten. Für den nächsten Entwurf haben wir uns überlegt die Gestaltung der GUI an der eines Musik-Players anzulehnen.

Nachdem wir die 2D-Ansicht aller Ebenen der Kugelbahn in einem „Pixel-Art“ Stil gestaltet und die GUI implementiert hatten entschieden wir uns nicht mehr getrennte Play und Pause Buttons zu benutzen. Daher benutzen wir nun einen Button, der je nach Zustand Play oder Pause betätigt. Außerdem fügten wir einen Schalter zum durchschalten der Ebenen hinzu.

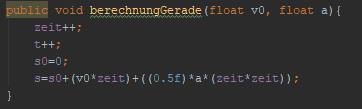
# Die fertige Anwendung

# Für die Simulation verwendete Formeln

Damit wir eine halbwegs brauchbare Physik haben, haben wir uns an der Vorlesung „Simulation und Modellieren“ orientiert. Für unsere Simulation haben wir die Formel der gleichmäßig beschleunigte Bewegung verwendet:

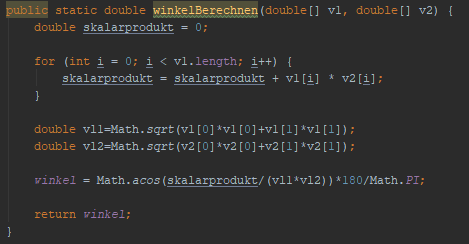
In unserem Programm haben wir diese Formel auch mit denselben Bezeichnungen so eingesetzt. Außer den Wert „*t*“, diesen haben wir durch die Variable „*zeit*“ ausgetauscht.



Um eine Kurve zu simulieren haben wir bei der draufschauenden Ebene ebenfalls die Formel der gleichmäßig beschleunigten Bewegung genommen und deren zurückgelegte Strecke anteilsweise an die x- und y-Achse übertragen.

Diese Anteile haben wir mit einem Winkel bestimmt, um eine möglich sauber Kurvenlaufbahn zu bestimmen. Die Bestimmung des Winkels wurde immer mit der aktuellen Position der Kugel und dem zu erreichenden Kurvenendpunkt ermittelt. Da wir uns in diesem Fall in einem zweidimensionalen Raum aufgehalten haben, mussten wir auf folgende Formel zugreifen:

Im Programm haben wir dies in der Methode „*winkelBerechnen*“, wie folgt umgesetzt:



Auch bei den freifallenden Kurven in Z-Richtung haben wir dieses Prinzip so angewendet.

# Übersicht

Leon Wessel: 35%  
Michael Lopes Marques: 32%  
Henri Katzenberg: 33%

Sowohl an der GUI als auch an der Umsetzung der Simulation war jedes Teammitglied zu ungefähr gleichen Teilen beteiligt.

# Quellen

Brick-Textur auf den Elementen der Kugelbahn: <https://media.indiumgames.com/medialibrary/2014/07/Pixel_BrickWall2x4.png> (abgerufen am 19.05.2019)