

Binding of glass Mini golf 2

Can you hit it?

Florian Winston, 11727495, winston@student.tugraz.at
Leon Tiefenboeck, 11919874, tiefenboeck@student.tugraz.at

April 30, 2025

1 Game Documentation

So far we have implemented two levels, where each showcases a different technique, however for the levels to be really complete and make sense to play the second technique is needed (per level).

The game is started by opening `main.html` which opens a start screen where one can choose between the two levels. The main concept is the same in both levels: In the bottom of the game area there is a blue ball that can be moved by clicking it, dragging somewhere and then releasing the mouse button. The ball will move in the opposite direction of where was dragged and the momentum of the ball depends on how far back was dragged, much like a spring. Somewhere else in the game area there is a green circle. This is the hole where the ball should go into. If the player manages to get the ball into the hole the level is completed and they are put back into the main menu. On the left side of the screen there is the control panel where level specific techs and the overall animation update rate can be adjusted. Here one can also toggle the needed visualizations for different techs. Now below we will outline the different levels and their techs.

1.1 Level 1

This level so far features the Path Interpolation tech. However this will really only make sense once Rigid Bodies are also implemented, since the moving objects should act as obstacles where the ball bounces off of. For this reason we only added two splines to showcase, one that forms a wave and the other just a straight line. As for now the movement only loops but we plan to add the option to make them move back and forth.

1.2 Level 2

2 Technical Specifications

We implemented our game very modular way. We have one main game class that controls the game logic and everything that is the same in each level. This class also includes one main loop that updates the position of objects and renders them. We can very easily add other objects from other parts in the code to this loop. We can adjust the animation update rate here by simply adding a delay to when the next iteration of this loop is called. We achieve the framerate independence also here, by calculating how long a frame took to produce and then adjust all updates by this delta.

Next we have an individual class for each of the techs. Here everything should also be confined and other parts of the game need only interact with the constructor and the **update** and **render** methods.

Finally we have a file for each level that interacts with the main game class and the different techs. Here we set up the objects and their techs in the respective level and the controls that are level specific (visualization, speed adjustment, etc.). If everything is set up we add the **update** and **render** methods of the objects (that come from the techs) to the main game loop and the start the game by calling the loop.