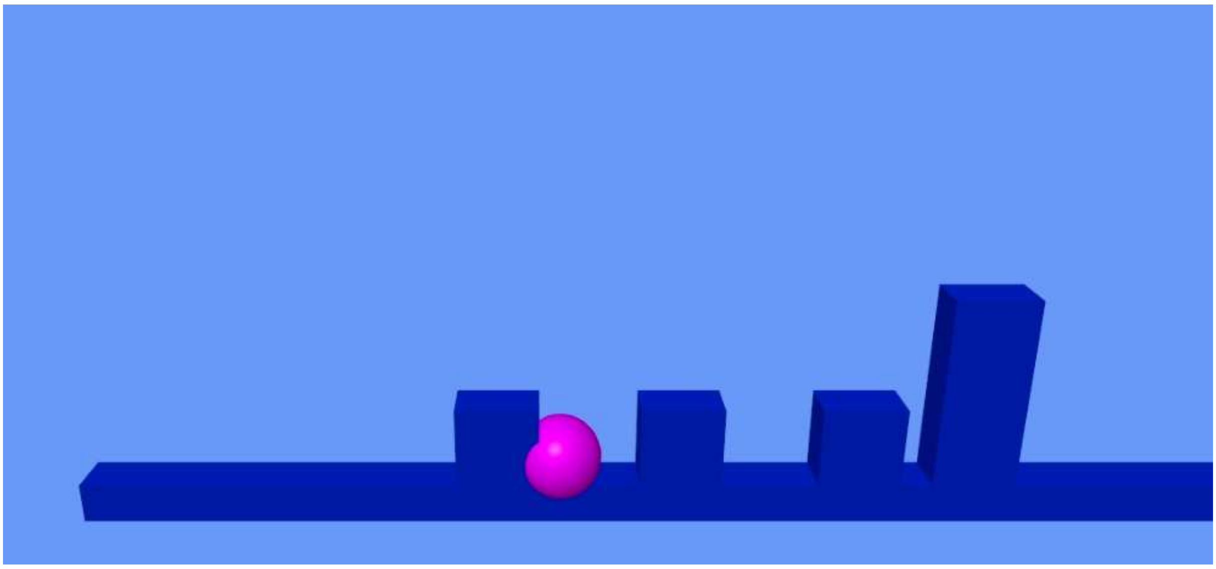
Interim Report

We have made a more detailed plan with each feature, how long we will spend for that feature and who is going to implement it. Each week we sat together and discussed problems and assigned new tasks for the following week. During the week we communicated through E-mails or through our Facebook group, which turned out to be very handy.
Right now we have nearly completed our low target. What is still missing are 2 more level concepts and the corresponding models and object textures for the level model.

Process and Problems

We started by first creating an UML diagram and discussing the basic implementation of our game. We also tried to get some design patterns into our design, for example we are having a State pattern to set the active screens (menus or game play).
We created the classes and soon had a first version of our game:



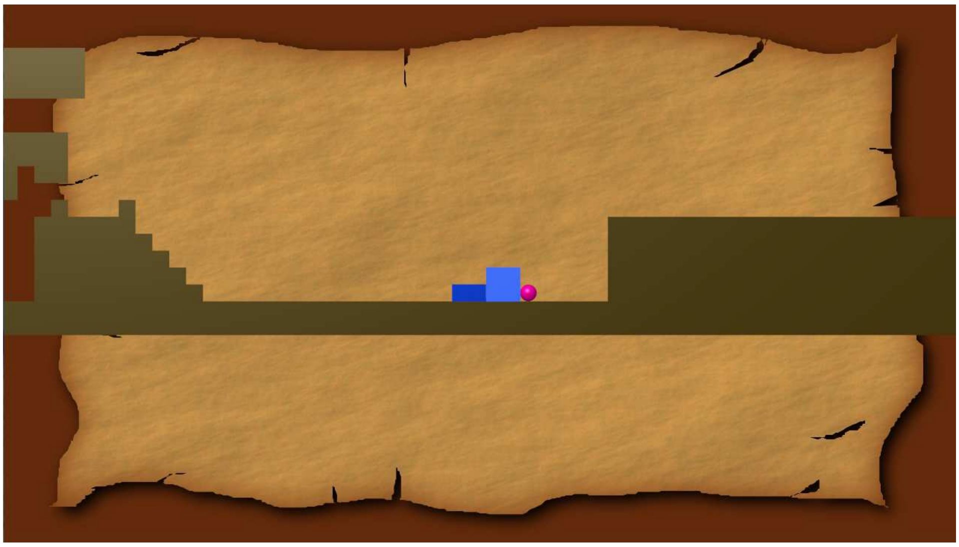
Very first version

This version only had boxes and a sphere as the Little Sister. Little sister could already jump and turn the boxes around. There is a distinction whether a box is dynamic or static. With static boxes, as the ground in the picture above, there is only collision. The parameters for jumping and running of Little Sister will still need fine tuning.
A problem we already had here, was the mapping between the 3D models and the 2D shapes that Farseer needs to construct the fixtures for the collision. For Farseer, these shapes need to be convex.
We constructed a first solution that worked for simple boxes, where we had two parts, a 3D model from Maya and a 2D model, from Maya as well. We then extracted the vertices from the vertex buffer of the Maya model.

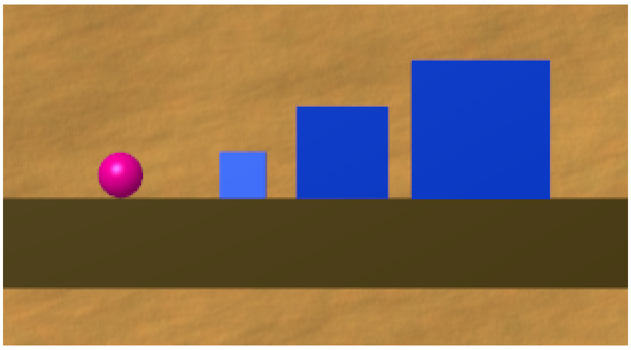
Based on this version, Fabian and Christian implemented box selection, box enlarging/shrinking, shaders and camera movement, whereas Marina modeled the first level in Maya. This turned out to be much more work than we had estimated. We thought that, as our solution for getting the Farseer shapes was working for boxes, this would also work for a whole level. This assumption was of course wrong. We had a lot of trouble to get a level that worked. The vertices that we extracted from the shape model were completely out of order and the indices from the index buffer were not meaningful. This of course did not generate the desired shapes in Farseer.

We also tried a different approach, where we created the shapes using the TextureToPolygon function from Farseer. We used a rendered image from Maya where the shape was stored in the alpha channel. The construction of that polygon actually worked quite well, but here we had the problem of setting the correct scale and transformation vector so that the shape matched the 3D model. This was tedious, so that solution was discarded as well.

After that, Christian started to work on that topic. He implemented an approach where he extracts first the vertices from the 3D model and then constructs neighbour information. The final vertices are those that lie on the boundary. Up until now, this approach is the most promising but it still fails to generate shapes from concave polygons. This is also why we did not yet think about two more levels, as our first level is not yet working. So this is the version we're having so far:



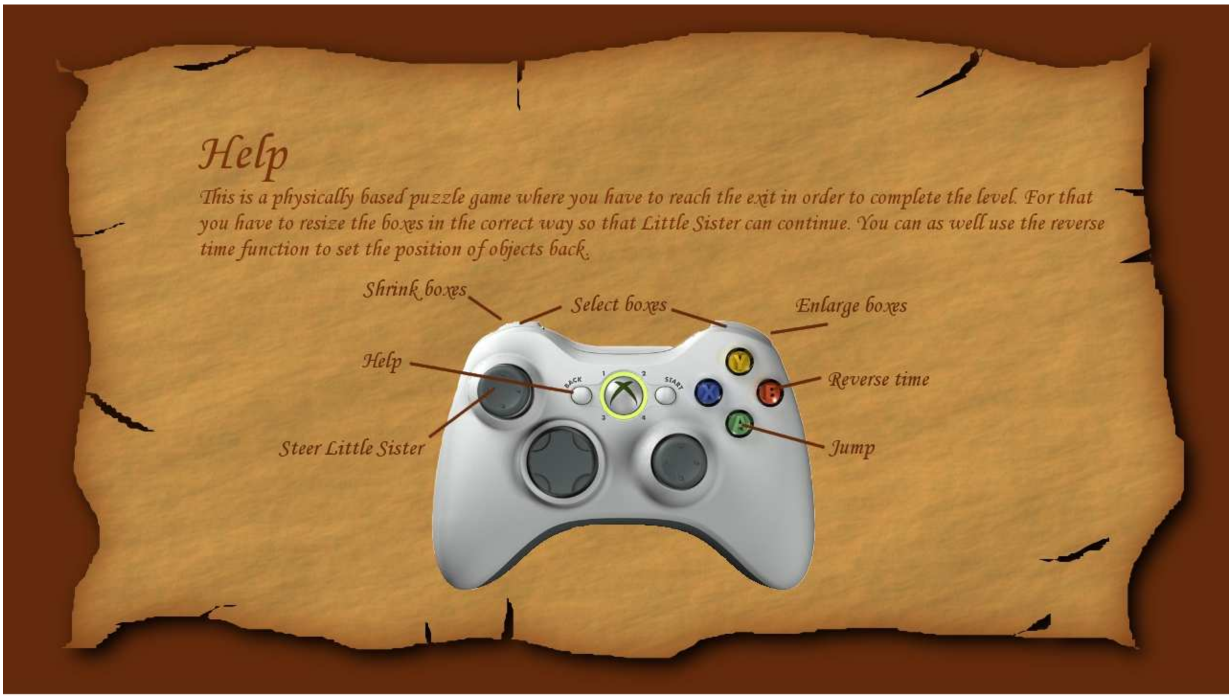
Little Sister can select boxes, then they get colored in light blue, and when selected, she can resize them. As explained in our prototype we only support 3 different sizes of the boxes. The boxes are also movable.



We also added a background texture, altough the texture above is still a placeholder. We implemented the Reverse time function. Then we already have background music; help screens, story screens and a start menu screen. All these screens are managed by the class ScreenManager, here we implemented a State pattern. Each level is in fact a screen as well.

Help screen:

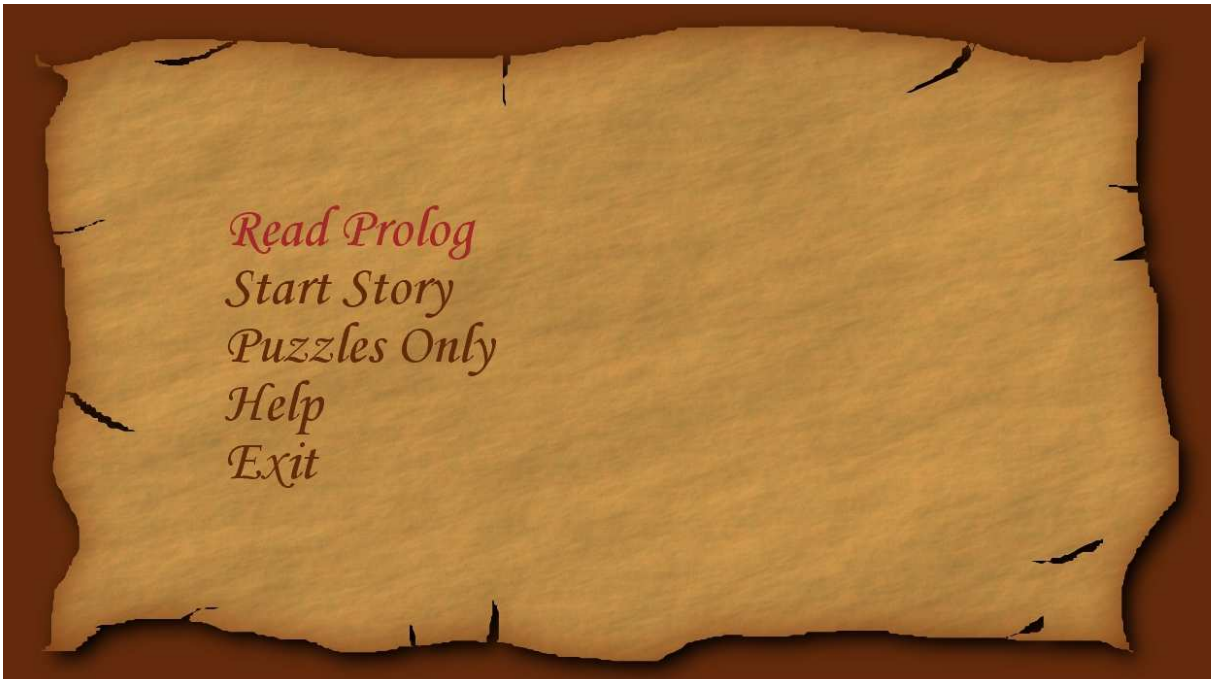
Here the user can get support for playing the game. This help screen is shown by pressing the back button and exited again by pressing the B-button.



Help

This is a physically based puzzle game where you have to reach the exit in order to complete the level. For that you have to resize the boxes in the correct way so that Little Sister can continue. You can as well use the reverse time function to set the position of objects back.

Start up of game:



Apart from our big problem with the shape creation, we did not have many other problems. One problem concerned Little Sisters jumping behaviour, we want her to only be able to jump once and pressing A again should not catapult her higher up. A first solution did not work on boxes that were rotated. On these boxes she then could not jump at all. A solution to that problem was the use of a SensorShape.
All in all we can say that our game already has the basic functionalities that we have desired if only we had a fully working solution for the 2D shape construction.