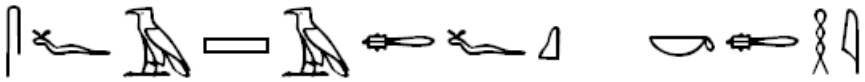
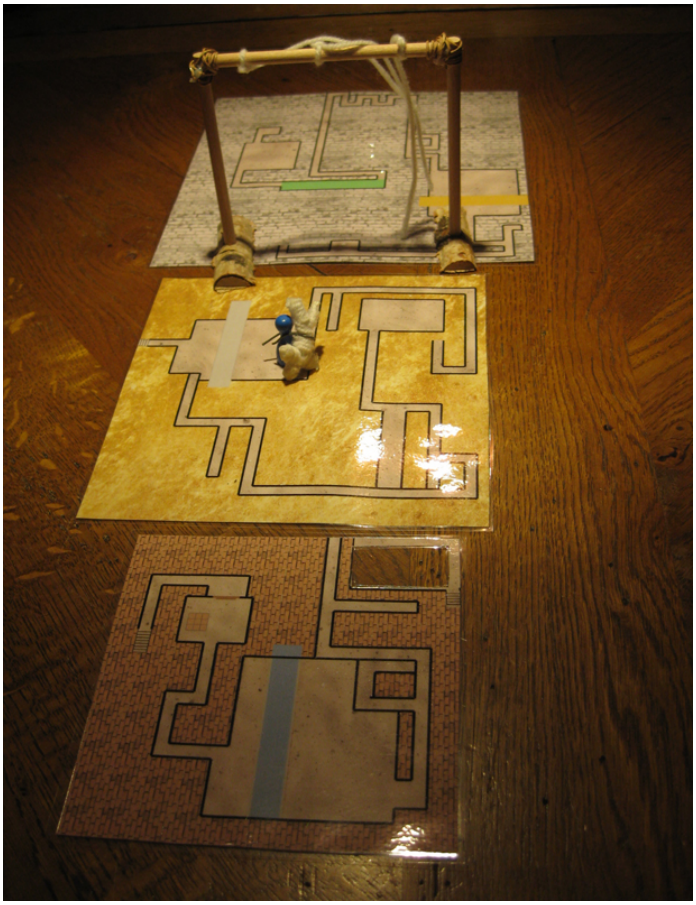


Game Programmming Lab

Game Prototype

Pharaoh's Tomb

Melanie Imhof
Stefan Geiger
Thomas Brunner



Contents

1 Prototype

1.1 Playground

The playground is divided into several different levels of a pyramid. The picture 1 shows three sample levels. The top of the pyramid is in Level 1 which thus is the smallest. The player starts in the first level and has to make it's way through all the levels. There are a lot of corridors where the player can walk along as well as different places where the player has to solve puzzles.

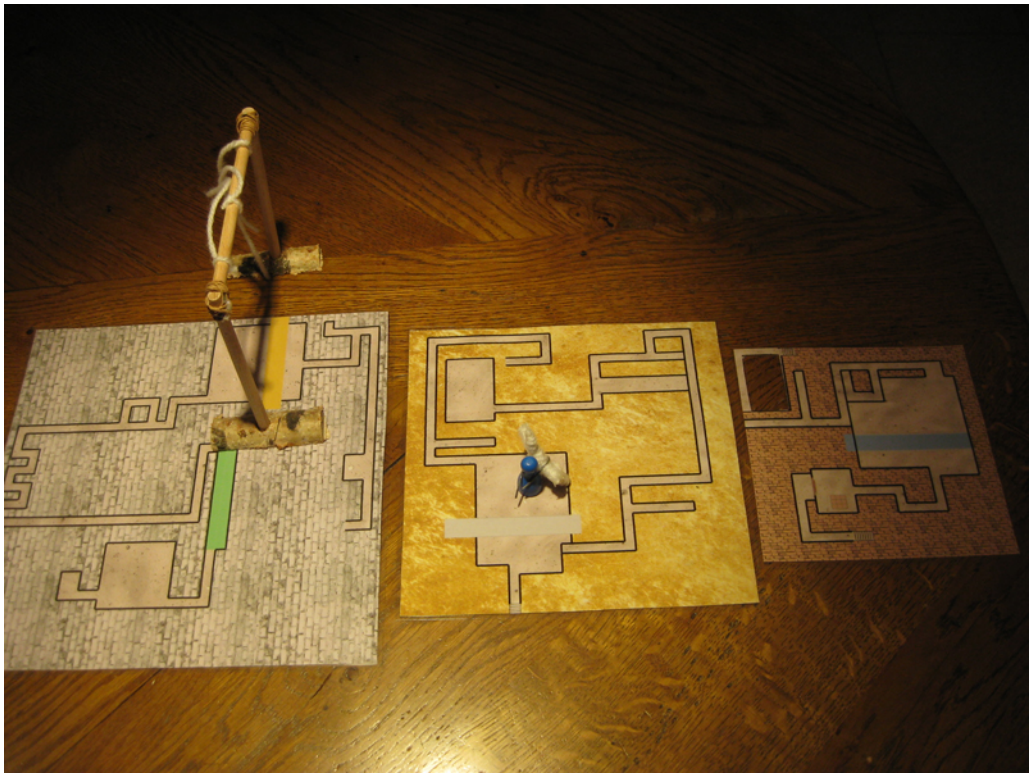


Figure 1: The different playground levels.

1.2 Gameplay

The game can be played with at least two players where one of them simulates the computer and the other is the actual player. The player is represented by the blue token carrying the mummy as shown in picture 2 which is represented by the doll made out of a pipe cleaner and wool. The goal is to get to the tomb with the mummy and the token. To get to the tomb several puzzels have to be solved but the spirit of the pharaoh and the knowledge of the egyptian culture will help you to solve them.
If the player did not solve a puzzle correctly he has to redo it, that's possible because the spirit of the pharaoh saves you from dying.
If there are more then two players they can solve the puzzles cooperatively.

Figure 2: The blue token with the mummy.

Figure 3: The woolen pharaoh.

1.3 Rules

- All puzzles need to be solved.
- To win the game the pharaoh need to be brought to the tomb.
- If a player fails to solve a puzzle he needs to redo it.
- Anytime the pharaoh can be asked for help.

1.4 Puzzles

It is possible to solve the different puzzles either by re-enacting the scenes and trying to solve them in person or by drawing the solution on a paper. Especially the physics puzzles are easier to draw on paper and this also allows complexer setups.

Player A, who simulates the computer, first stages the scene which includes the goal as well as the existing objects which can be used. While the other player(s) tries to solve the puzzle, player A can give hints and can restrict the solution domain to the feasible ones. The images 4 and 5 show two puzzle setups and one possible solution each.

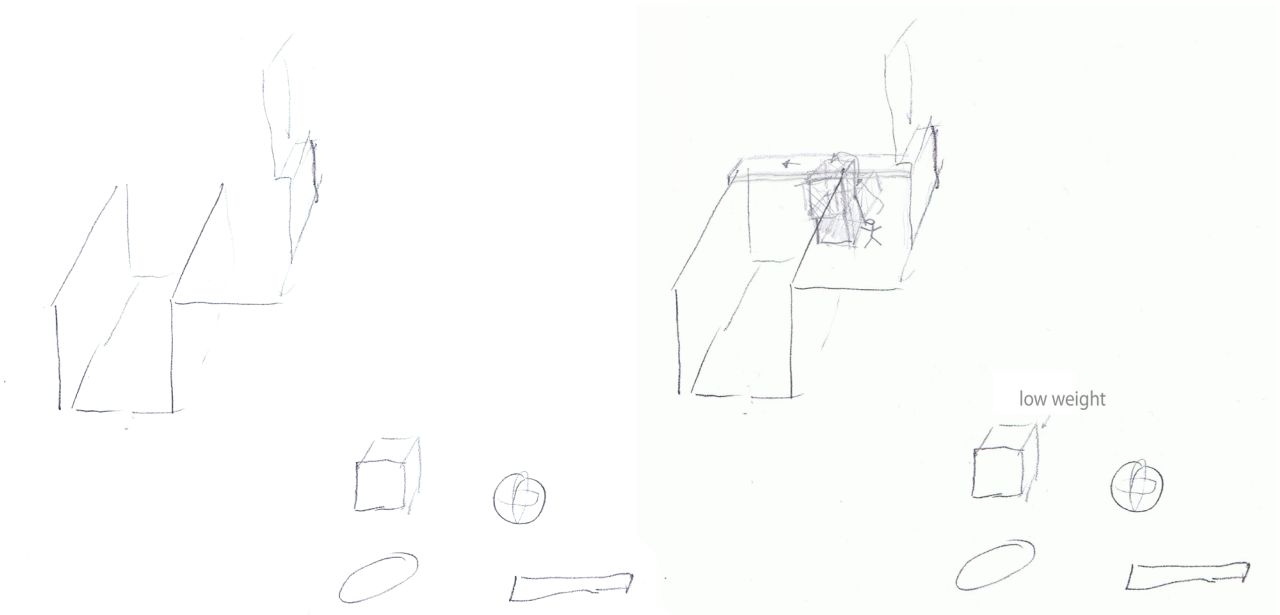


Figure 4: Physics puzzle 1. The left image shows the scene drawn by player A (computer) and the right image shows the scene with the solution by the other player.

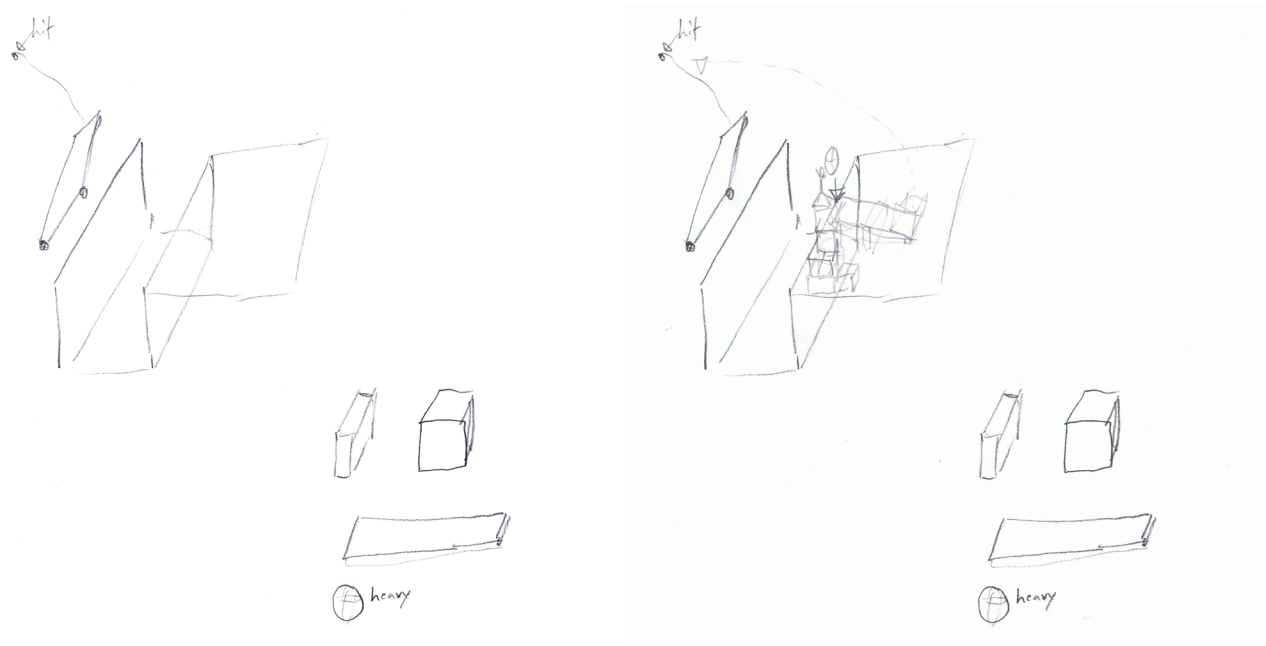


Figure 5: Physics puzzle 2. The left image shows the scene drawn by player A (computer) and the right image shows the scene with the solution by the other player.

The image 7 shows the physical prototype of the path of death puzzle. First a sequence of 5 images is shown to the player for 4 seconds eachs. Afterwards a path is laid-out at the floor with 8 other pictures. The player has to try to follow the 5 images in the right order.

Figure 6: Path of Death puzzle.

The image **??** shows the physical prototype of the spikes puzzles. The player has to try to walk along a path which is barricaded by several people armed with brooms acting as moving spikes.

Figure 7: Spikes puzzle.

1.5 Conclusion

One of the conclusions we made out of the physical prototype is that physics puzzles get difficult very soon. Thus we need to provide hints in different steps, including an intuitive scene setup, which highlights the possibilities.

For the path of death puzzle, memorizing 5 out of 13 pictures is possible if the pictures are easily distinguishable. With 4 second intervals when showing the picutres a longer sequence is still possible to memorize, however reducing this interval makes it a lot harder.

The spikes puzzle is quite hard in person, but makes a lot of fun. In the game, the character is more agile which allows faster moving spikes and more difficult scenarios.

2 Development

2.1 Status

We currently have a working I3D loader, which supports the basics attributes such as the scenegraph structure, translations, rotations and geometries. There is also a basic renderer which traverses the scenegraph

and displays the scene with the BasicEffect class. The BEPU Physics Library was included and integrated into the I3D loader pipeline to automatically create dynamic, kinematic and static objects out of the attributes stored in the I3D files. There is also a character controller which moves a cylinder through the environment with appropriate collision detection. We also used the camera controller by Thomas Oskam to have a smooth 3rd person camera movement.

To finish the functional minimum we need to add some more I3D loader and scenegraph features, to create the character model as well as the test environment with two puzzles. To make these puzzles playable we also need the environment interaction.

Image 8 shows a screenshot of the current development state. In image 9 and 10 is the same scene shown inside the GIANTS Editor and Maya.

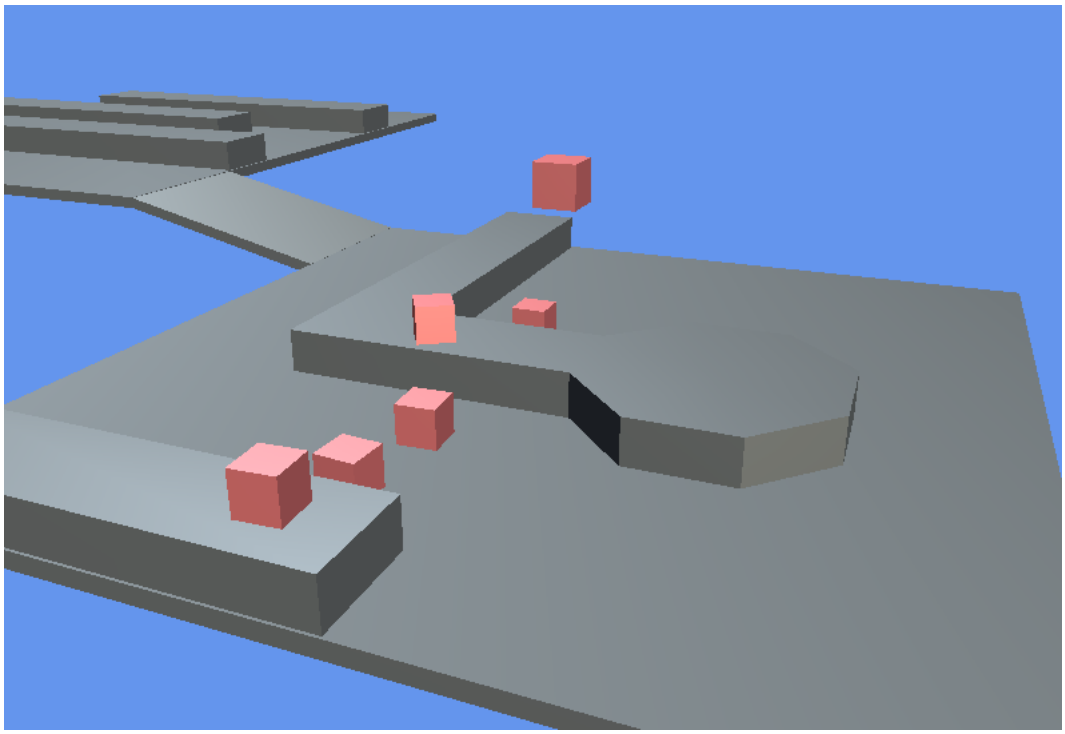


Figure 8: The image shows an ingame screenshot of the current development state, with the map.i3d loaded.

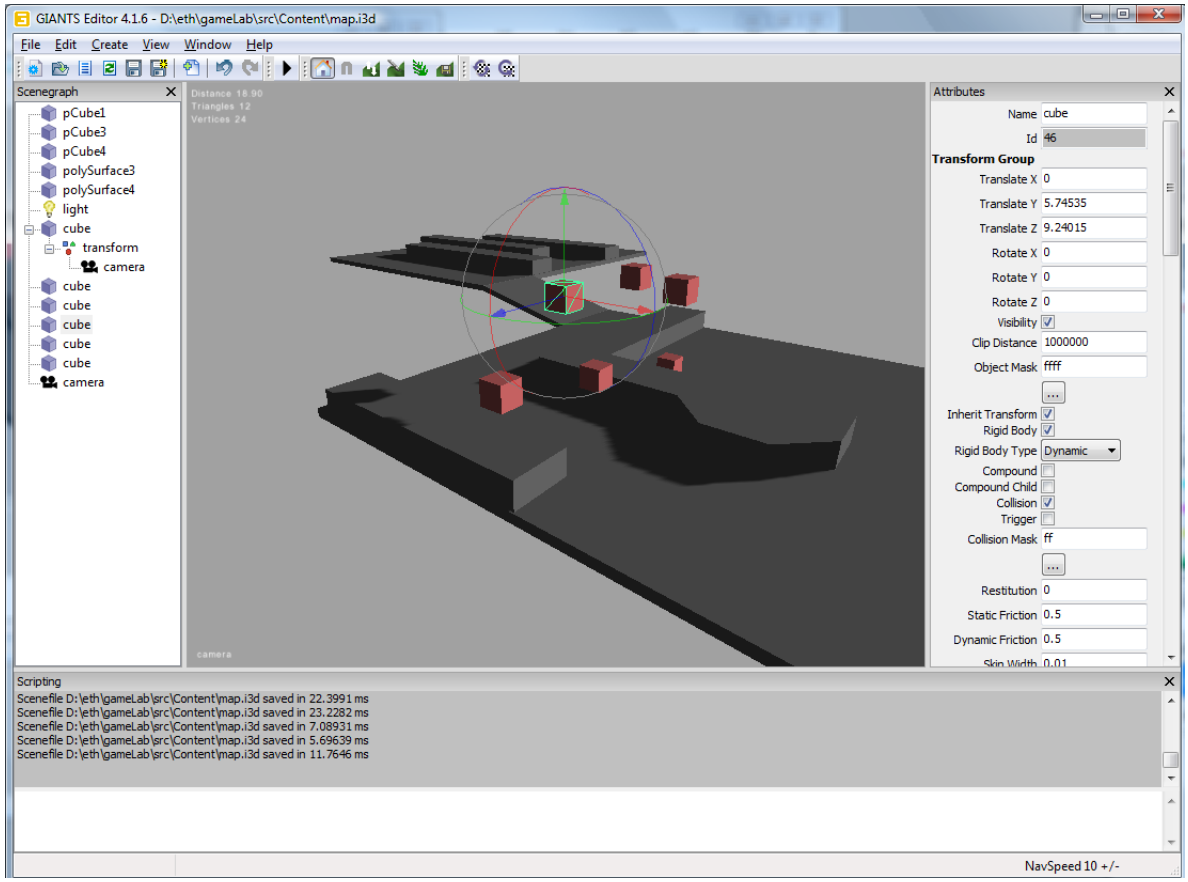


Figure 9: The image shows the map.i3d inside the GIANTS Editor which we use to modify the environment.

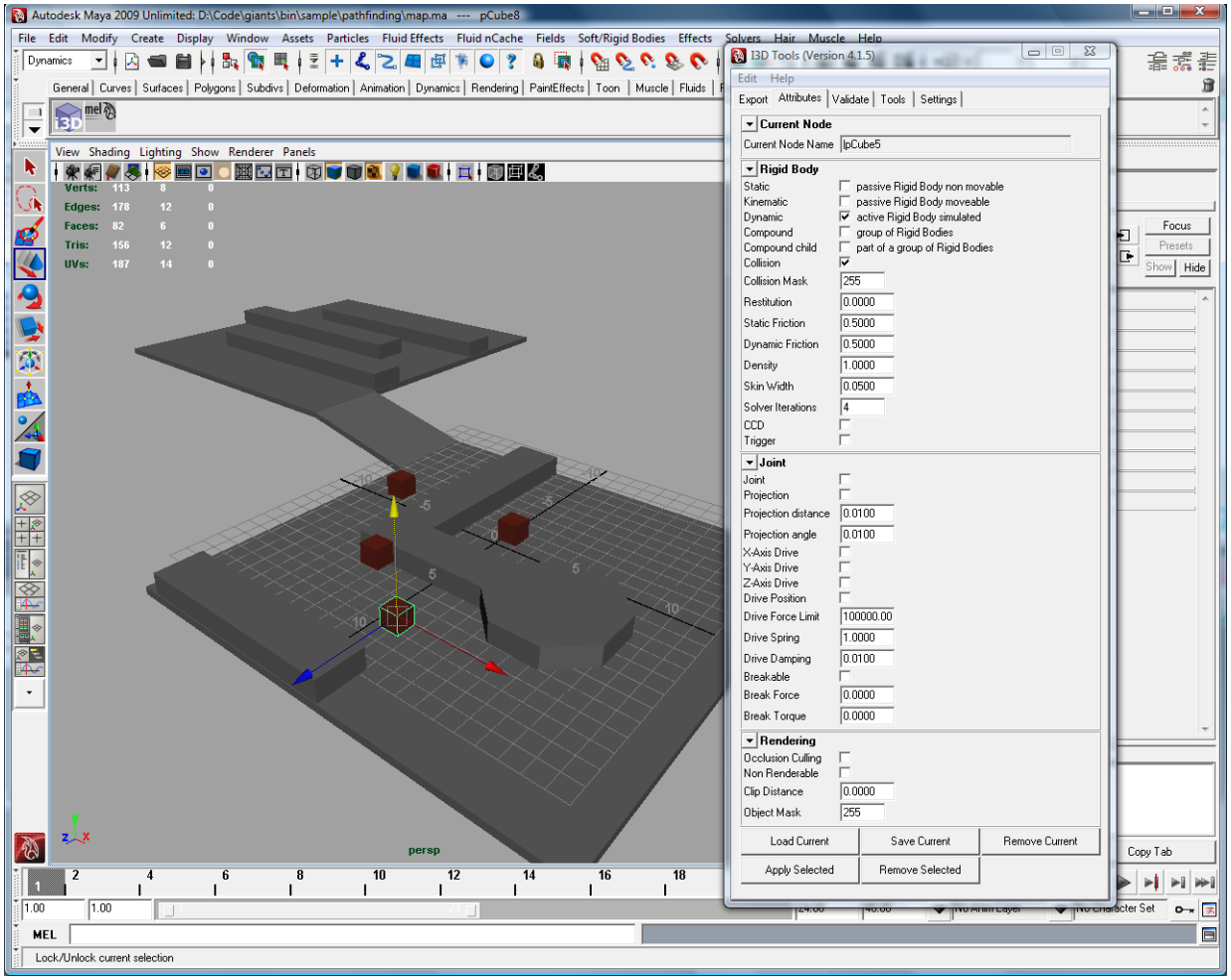


Figure 10: The image shows a sample scene in maya with the I3D exporter options which we use to create the models.