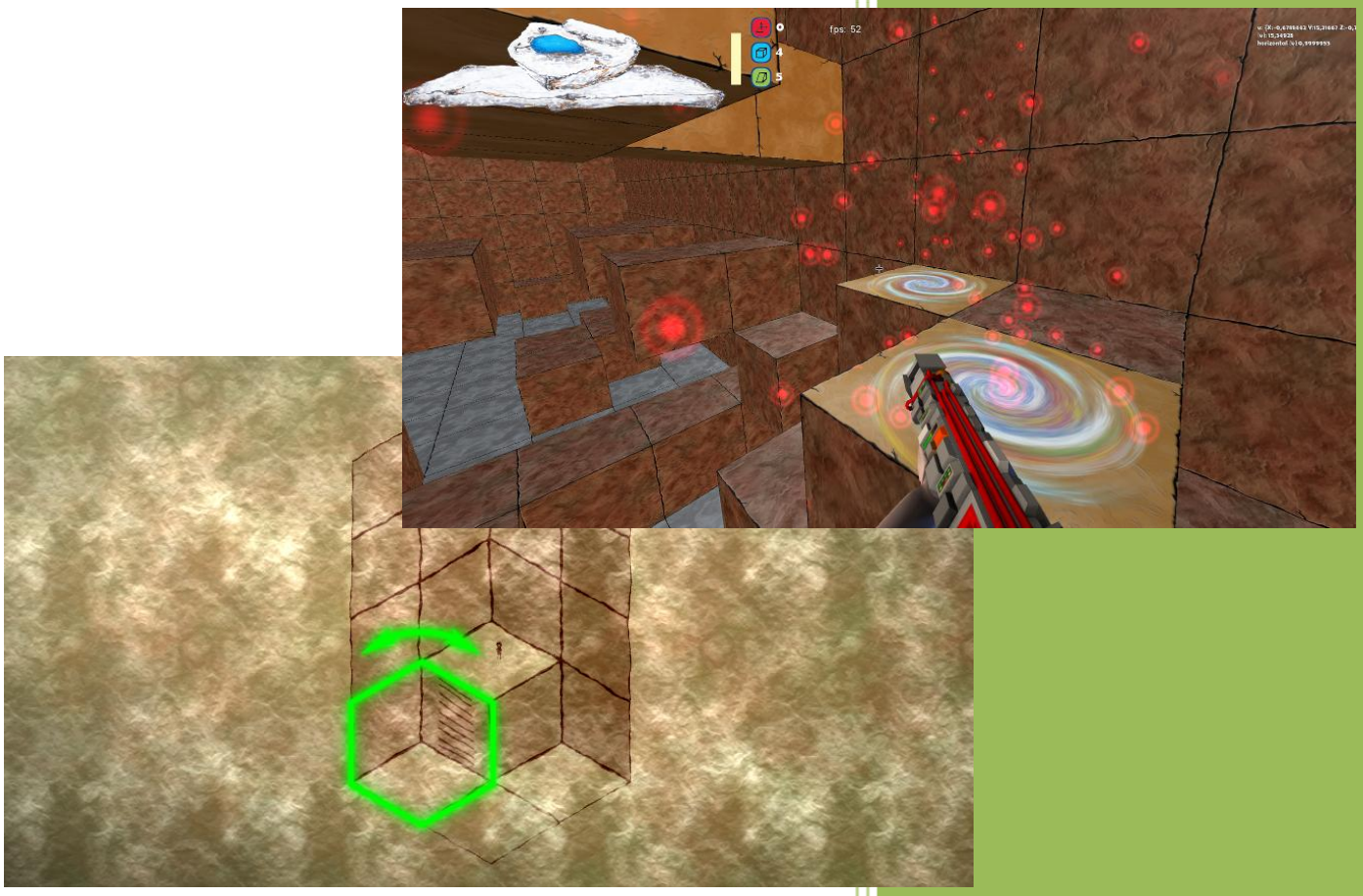
2010

UNSTABLE TALENT

An artistic treasure hunt through time



Bayer Jeronimo, Flierl Matthias, Kiser Thomas
Game Programming Lab – ETH Zürich
10.05.2010

TABLE OF CONTENT

1. PROGRESS

We have implemented almost all features of our desired target, and also the achievement feature listed as high target. As we already wrote in the interim report, we did not keep track of the exact man hours we invested, but they are still consistently higher than listed in the timetable of the initial game proposal.

Due to the limited man power (see section 3.), we dropped some features that were initially planned.
One is that we intended to make gravity fields larger by re-painting its base a second time. Although this might have been fairly simple to implement, we decided to focus on the more essential features. While creating levels, we never missed this feature.
We also cancelled the undo/redo functionality. We realized it would be too complex to implement (it would have required a snapshot of the complete physical world) in a reasonable amount of time and therefore abandoned this feature.

On the other hand, we also built some features that were not listed in the initial game proposal.

We came to the conclusion that implementing the level selection as a three-dimensional room instead of the planned menu would not only be cooler, but would also allow us to reuse the already-existing functionality and might therefore save some work. Needless to say, we went with this idea. We called this entry room the Lobby.
We also decided to display the achievements collected by the player in the lobby. These models were also not planned in the initial time table.

Since the get the player acquainted with the features, we also added a special tutorial level with a HUD to give the player some directions.

1.1. WORLD

We already had the physical model of ladders at the time of the interim report, but it only worked for one orientation. Support for the remaining directions was added.

We have also added the volumes that detect when the player hits an achievement or exit. Visualizations for these fields are in progress and sould be ready until the presentation of the alpha release.

The control that allowed movement in mid-air was also already present, but the velocities quickly added up while the character was off the ground. By quickly jumping again when touching the ground, one could attain a much higher speed than by walking. We found this bunny-hopping feature pretty cool and modified the controls so one wouldn't get high speeds that easily, but could still move two to three times as fast as walking.

1.2. PAINTING

To make the painting of ramps more intuitive, we added a functionality which allows the player to first create a dummy ramp which he then can move and rotate. When finishes he acknowledges the changes by pressing the shoot button again and the final ramp gets created.

1.3. ISOMETRIC VIEW

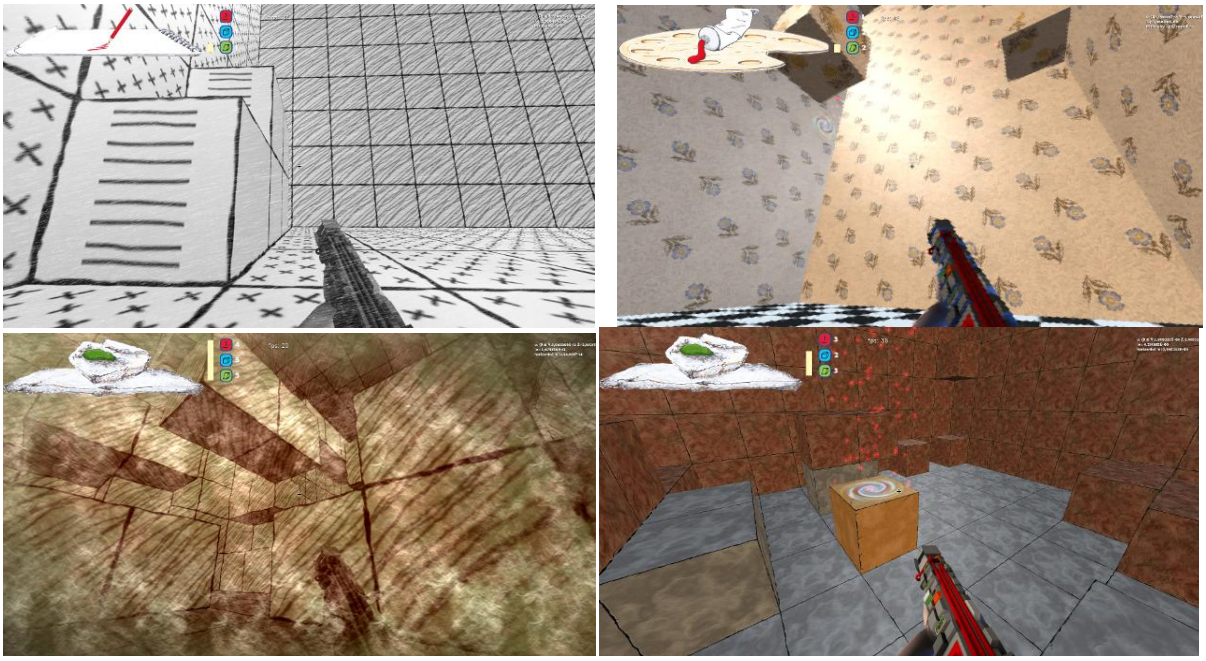
The missing operations in the isometric view are now fully implemented.
These are rotations which remove a cube from the scene, and the modification of the physical elements corresponding to the rotations being performed.
We also changed the sprites to display whether the hexagon will be rotated by 60 or 120 degrees. We intend to allow clockwise and counterclockwise rotations, but at the time of writing, this is not supported. (The functionality is present but untested, what's missing is mostly handling the corresponding button presses. In the optimal case, adding this feature will take less than five minutes.)

1.4. INPUT HANDLING

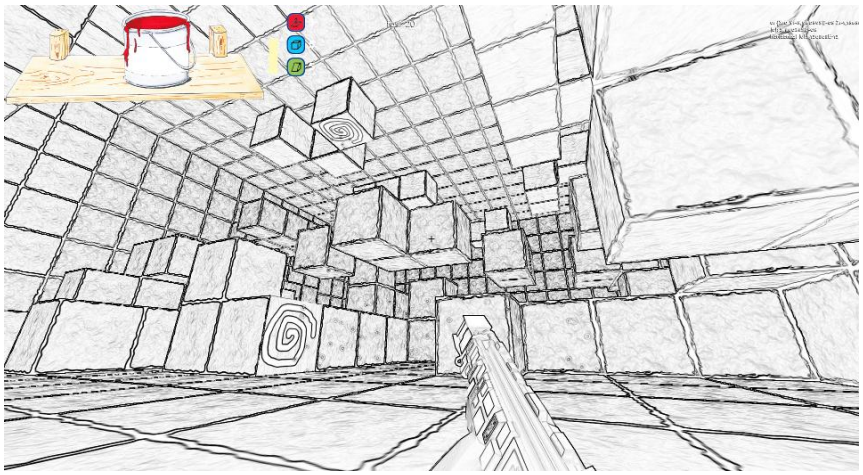
While progressing in the code, input handling was scattered all over the place, so we decided to centralize all input requirements in an input depot. Therefore we were able to introduce a button and key change mechanism to personalize the gameplay to the desires of the player.
Not just single button events but also one- and two-axes movements are handled by the input depot. A triggered movement method was added, which allows to get the value pressed by the user just at discrete time steps, allowing better handling of cursor actions e.g. in isometric view. The trigger frequency is also dependent on how far the thumbstick is pushed.

1.5. GRAPHICS

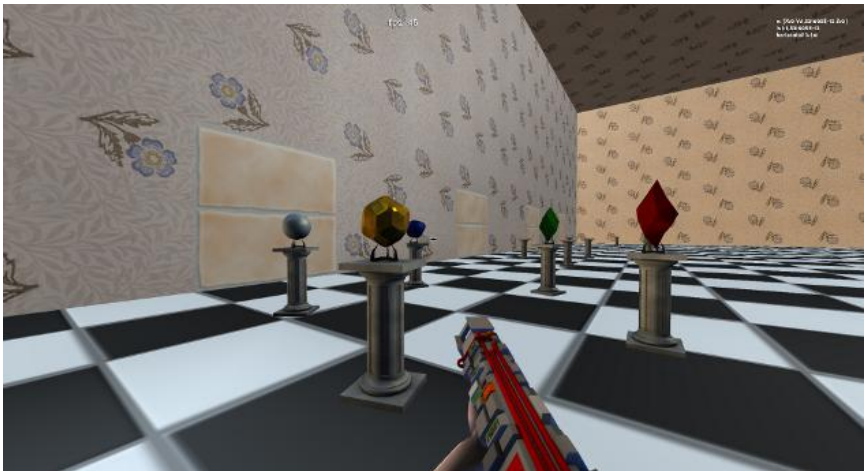
To represent that the player is traveling through different times, we added matching sprites to the individual shaders. For the Stone Age Era for example, the color is held on a stone plate. Every era was also complemented with its own texture set.
Here the current looks of the different painting eras:



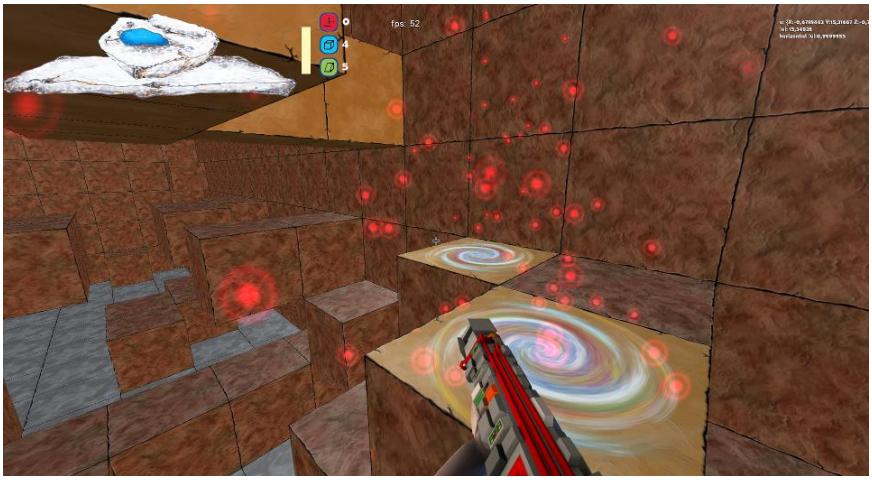
There is also a fourth painting Era for modern art, realized by an edge detection shader, but there are missing adequate textures. A snapshot of how it could look like here:



There are achievements which the player can collect in several levels and are then displayed in the lobby.



We also added a particle system computed on the GPU to illustrate special regions, i.e. gravity fields, doors and volumes that need to be touched to collect achievements.



We have also integrated the shooting animation into the game, and added a bobbing camera movement when the character is walking.

1.6. MENU

To give the player a entrance point, we added a menu to the game which allows to start a new profile, to load a saved profile or to change some game values such as the music volume or the gamma value of the graphics displayed. As mentioned in "Input Handling", we are now also able to let the player choose key/button settings. The choice is not completely free, so for the XBox controller for example one can exchange the two thumb sticks with each other but not with arbitrary buttons. All other buttons can be used interchangeably. These features are brand new and not committed to the source code repository at the time of writing, but we hope to get everything completed until the presentation of the alpha release.

2. PROBLEMS

2.1. LIMITED MANPOWER

Some days after the interim report, Matthias noticed increasing headache, ending in a incident where he lost consciousness for some time. He still suffers from that incident until now, tried to contribute as much as he could to the project, but was really restricted in the amount of time being able to work. Also focus was limited, so complex work was partially overtaken by other group members. He is really sorry for the team, that he wasn't able to work on the project as much as there would have been needed, but is looking forward to catch up again during the final three weeks of the project.

2.2. PHYSICS ISSUES

Right before the interim presentation, we noticed a severe performance issue when several dynamic objects were affected by gravity fields. As we soon suspected, the callbacks triggered when an object enters a field were causing this problem. We finally solved this issue by deriving a new force field class from BEPU's, and tagging entities with information about when they were last seen in a gravity field. This allowed us to perform the operations necessary when an object enters a field.

We also were experiencing several problems with the character movement. One was that the position would sometimes flicker because the character was repositioned on the floor, but the height was incorrect. It turned out that the origin used for the corresponding raycast sometimes caused inaccurate results. The BEPU library generally didn't make a good impression related to raycast accuracy, even before the interim report we already had issues with bad normals at the hit position.
Another issue was that jumps against the wall sometimes quickly pushed the character upwards and out of the map. This was also related to this raycast and the subsequent repositioning on the "floor": When a raycast starts inside an object, Bepu reports a hit directly at the ray origin. This occurred when one walked or jumped against a wall, and apparently the extent of the large boxes we use as physical walls is so inaccurate it caused this problem. We manually worked around this issue by checking how far the ray origin is "inside" the cube, and if it's closer to a wall than the floor or ceiling, don't use that object to reposition the player on the ground.

We also had problems with detector volume callbacks on ladder volumes. These incremented a counter when the player character entered a volume, and decremented it when he left it. Sometimes the count was incremented once more than it was decremented although the player was not close to a ladder any more, which allowed the player to climb even when there was no ladder present. We worked around this issue by using a similar trick as for the gravity volumes, i.e. we completely stopped using Bepu's detector volumes and used a derivate of the force field class. That didn't quite fix the issue, until we finally realized that the collision collector object located at the character's feet was not properly repositioned there when he was considered to be in a ladder volume.

2.3. ANIMATION

Integrating the shooting animation for the model caused some problems. Although the framework provided by an XNA sample worked well on the PC, porting it to the Xbox was difficult. We finally got it to work. One issue was that it simply doesn't work when the project is stored on a network drive (as is the case on ETH computers) - with an error message that certainly doesn't tell what the problem really is, of course. Another issue is that the required custom content processor needs to be configured to build for x86 even when the project is built for the Xbox.

3. CONCLUSIONS

At the time of writing, several smaller items are on our to-do list. However, we hope to get these fixed by the time of the presentation. Some have already been implemented, but are not committed to the repository yet.