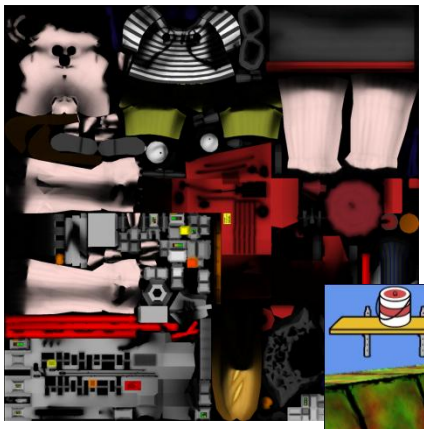
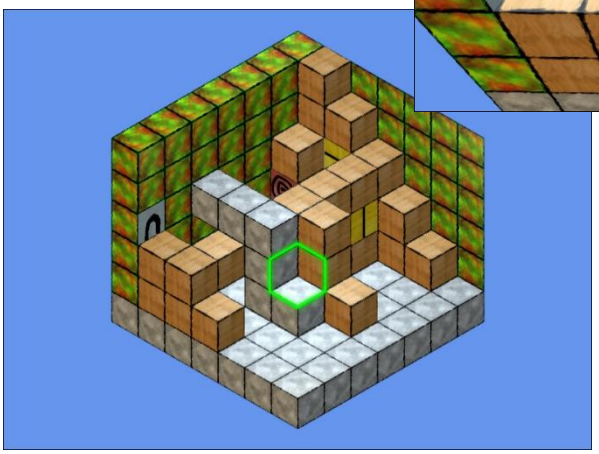
2010

YetAnotherWorkingName





Bayer Jeronimo, Flierl Matthias, Kiser Thomas

Game Programming Lab – ETH Zürich

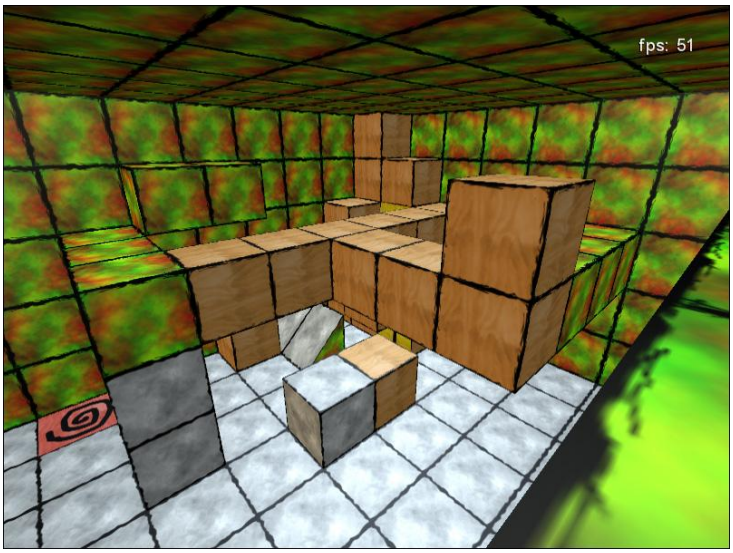
19.04.2010

TABLE OF CONTENT

1. PROGRESS

1.1. WORLD

We have implemented the cube world as described in the proposal. The player is able to move around in the common first person view by walking and jumping.

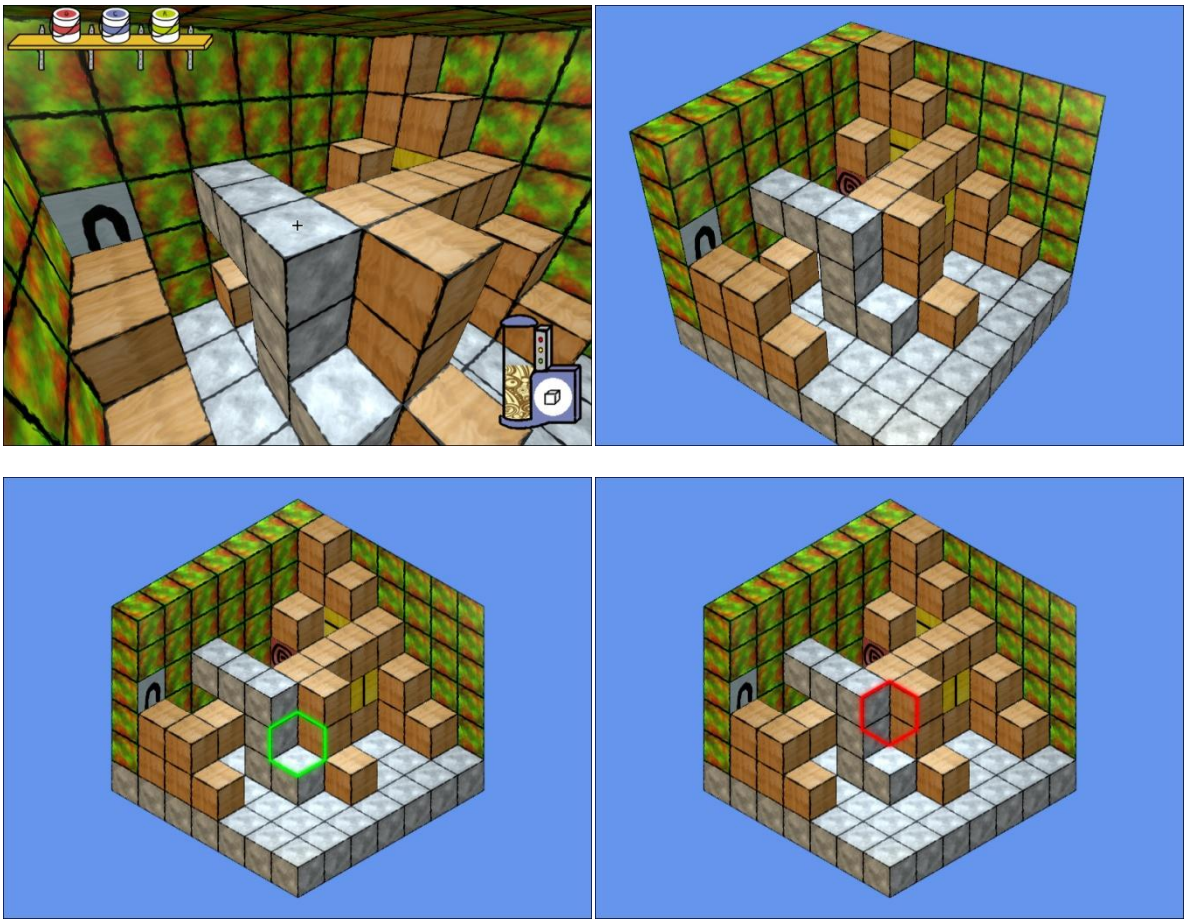


1.2. PAINTING

With his paint gun, the player can shoot around two different things: Cubes and gravity fields. The cubes are automatically painted in the same color as the hit point and fit nicely into the map. The gravity fields take an effect on the player and on dynamic objects in the world.

1.3. ISOMETRIC VIEW

The switch to the isometric view and back is implemented and looks quite smooth in most cases (the transition code is not yet final). The state displays a movable hexagon which indicates whether the corresponding cube can be rotated or not, although some checks are yet to be implemented, as is the rotation itself.



1.4. GRAPHICS

The HUD indicates the selected color and the remaining “ammunition”, although the sprites will need to be redone to match the level's visual appearance. One can also see the hands and the gun model, and the full character can be seen in the isometric view. However, this is not perfected yet and still in progress. A shooting animation is also done, but not incorporated in the game yet.

We have also implemented three shaders which will indicate different eras in the final game.

Since we have not implemented a level selection mechanism yet, a button is used to cycle through them.

Impressionism

Unfortunately, this shader exhibits fairly strong flickering on sharp edges, which is even enhanced by the black borders of our current textures. This will need to be considered when creating textures for this era.

Cave painting

Pencil Shader

2. PROBLEMS

2.1. VISUALIZATION OF THE MAP

As was to be expected, we experienced severe performance issues with a naïve implementation of the cube rendering. Instead of drawing every cube as a model, we used vertex and index buffers to draw the whole map in one go. As a further step, we fill only those triangles into the buffers which are on the outside and therefore may be visible to the user.
This quite sophisticated system eliminated all performance problems of the map visualization.

2.2. SHARED CODE

There are different ways for sharing code which work quite well. But they don't work well if you mix them. In our group, there was such a mismatch of code sharing techniques, which led to some quite frustrating problems.

On one hand, a group can separate their code into pieces, with as little overlaps and communication between the parts as possible. Everyone has his components of the code in which he and only he is the expert. The advantages are that no one has to look deeply into the code of the others and can use his full time for the design and implementation of his own parts.

On the other hand, a group can work in a very repository-centered way in which everyone tries to be as close to the repository as possible. People still have their assigned tasks, but the coupling between the code sections is tighter. This has the advantages that everyone sees what the others are working on, and more eyes can see problems faster. Since there are no local code branches, everyone can change other people's code without causing any conflicts. Another advantage is that it requires only little merging effort to commit code to the repository, since the users frequently update their local copies.

In our group, as mentioned, we had a bit of a mix of those two attitudes. One part of the group was frustrated because there were changes in the code which they didn't expect, which caused merging problems. The other part of the group missed the communication of what the first part was doing, and often didn't exactly know what sections they were working on, if at all. Therefore, the second part of the group tried to restrict their work to the "safe" code sections, with limited success.

To solve this problem we assigned every class to an author. People who want to work closely to the repository and share their code for debugging can do this, but only in the classes which are not assigned to authors who prefer the loosely-coupled way. We hope that this solves all future code-sharing problems in our group.

3. TIMETABLE

We did not keep track of our man-hours. For most tasks, we used a lot more time than listed in our initial timetable, in some cases more than twice as much. We were all aware that we would invest a lot more than 300 hours per person, but still based our time table on that amount. Therefore, the numbers were not really suited as a realistic estimate from the beginning.

= Works
= In Progress
= Not Started

	Phase[1]	JB[2]	MF[3]	TK[4]
23.-29.3.				
World with ladders and doors	FM	12		
Physics	FM		18	
Character control	FM		6	
Map handling (load, maybe save)	FM	12		
Creating a player model for isometric view	FM			24
30.3.-5.4.				
Character control	FM		4	
First person view	FM		4	
Interactions with ladders, doors and force fields	FM		12	
Creating a player model for isometric view	FM			24
HUD	LT	8		
Creation of force fields plus simple visualization	LT	8		
Creation of objects (blocks and ramps)	LT	8		
Yet unknown problems			4	
6.-12.4.				
Isometric view	LT			8
Rendering mechanism (e.g. painted/comics look)	LT			16
Two different simple levels	LT	8		
Saving with stats (e.g. timer)	LT	8		
Yet unknown Problems		8		
Menu for level selection	LT		8	
Textures for one era	LT		16	
13.-19.4. (Interim report due 19.4)				
Detecting possible transformations in isometric view and applying them	DT	8		8
Highlighting (2D masks for selection) in isometric view	DT	8		8
Textures for at least a second era	DT		16	
Prepare interim report		8	8	8
20.-26.4.				
Detecting possible transformations in isometric view	DT	16		16

[1] FM = Functional minimum, LT = Low target, DT = Desired target
[2] Jeronimo Bayer, working time in hours
[3] Matthias Flierl, working time in hours
[4] Thomas Kiser, working time in hours

and applying them				
Transformation to/from isometric view by camera movement	DT	8		8
Effect when creating objects	DT		16	
Yet unknown problems			8	
27.4.-3.5.				
Visualization of Transformation	DT	8		
Trailer video Brainstorming		8	8	8
At least two non-introduction levels	DT	8	16	16
4.-10.5.				
Sound	DT	16		
Enlarging force fields	DT		4	
Undo/Redo of operations	DT		12	
Effect/Shader of anti-gravity regions	DT			16
Prepare alpha release		8	8	8
11.-17.5. (Alpha release due 11.5.)				
Trailer video		16	16	16
Playtesting		4	4	4
Polishing		4	4	4
18.-24.5. (Playtesting presentation 18.5)				
Trailer video		12	12	12
Presentation		8	8	8
Conclusion		4	4	4
25./26.5. (Debriefing and final presentation)				