Toon Dimension

Interim Report

Peter Bucher
Christian Schulz
Nicola Ranieri

March, 2009

Table Of Contents

1. What's new?

1.1 Graphics

Although our original schedule did not include shaders until the alpha release, we hav first created a cel-shading light model and a silhouette shader. Originally we thought that the silhouette shader would be sufficient. But shortly after it was obvious that this technique alone would not enable us to reproduce the graphics that we had shown in our prototype video. To account for this shortcoming a second edge shader was introduced. Together the silhouette and edge shader support one another and produce quite nice results as it was actually intended.

1.2 Menu

A Menu has been added that allows the basic task of selecting new maps or replay previously finished ones. Its model is in 3D but uses sprites and fonts to draw dynamic content. This enables a toon shaded menu matching the toon world but also drawing map previews and names.

1.3 Models

On the model part a lot of work has gone into creating a simple story map and a multi-player map in 2 different versions for two or four players. As seen in Figure 1-3, all captured from the map editor. The story map will be used as the first map in the game and serves as an introduction/tutorial map to make the player familiar with the core concept of the dimension and the required team work.

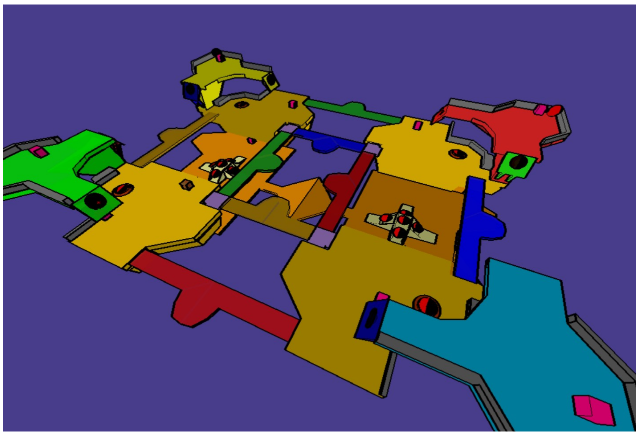


Figure 1: 4-Player version

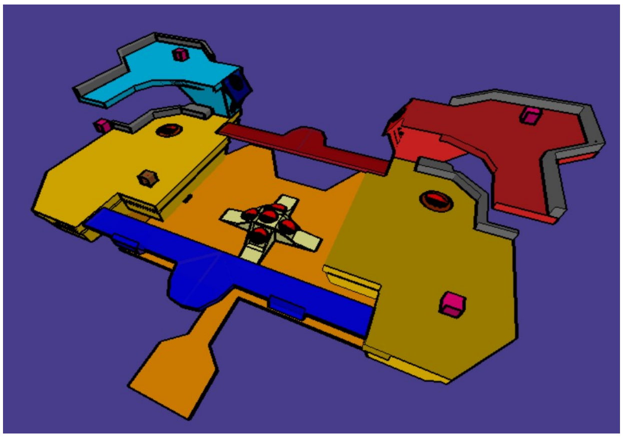


Figure 2: 2-Player version

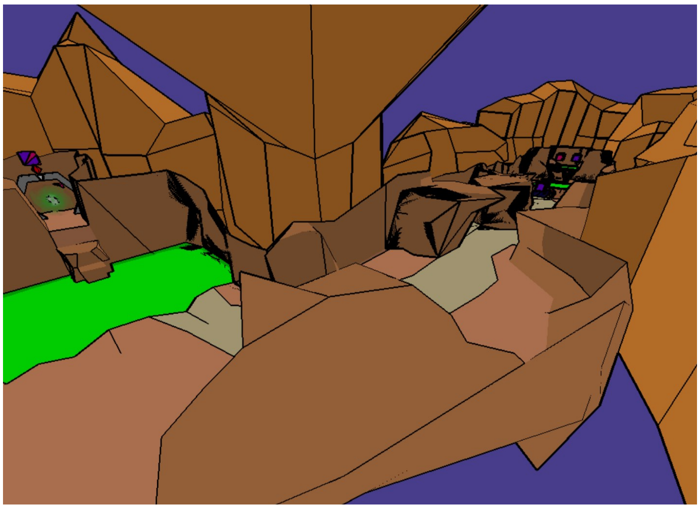
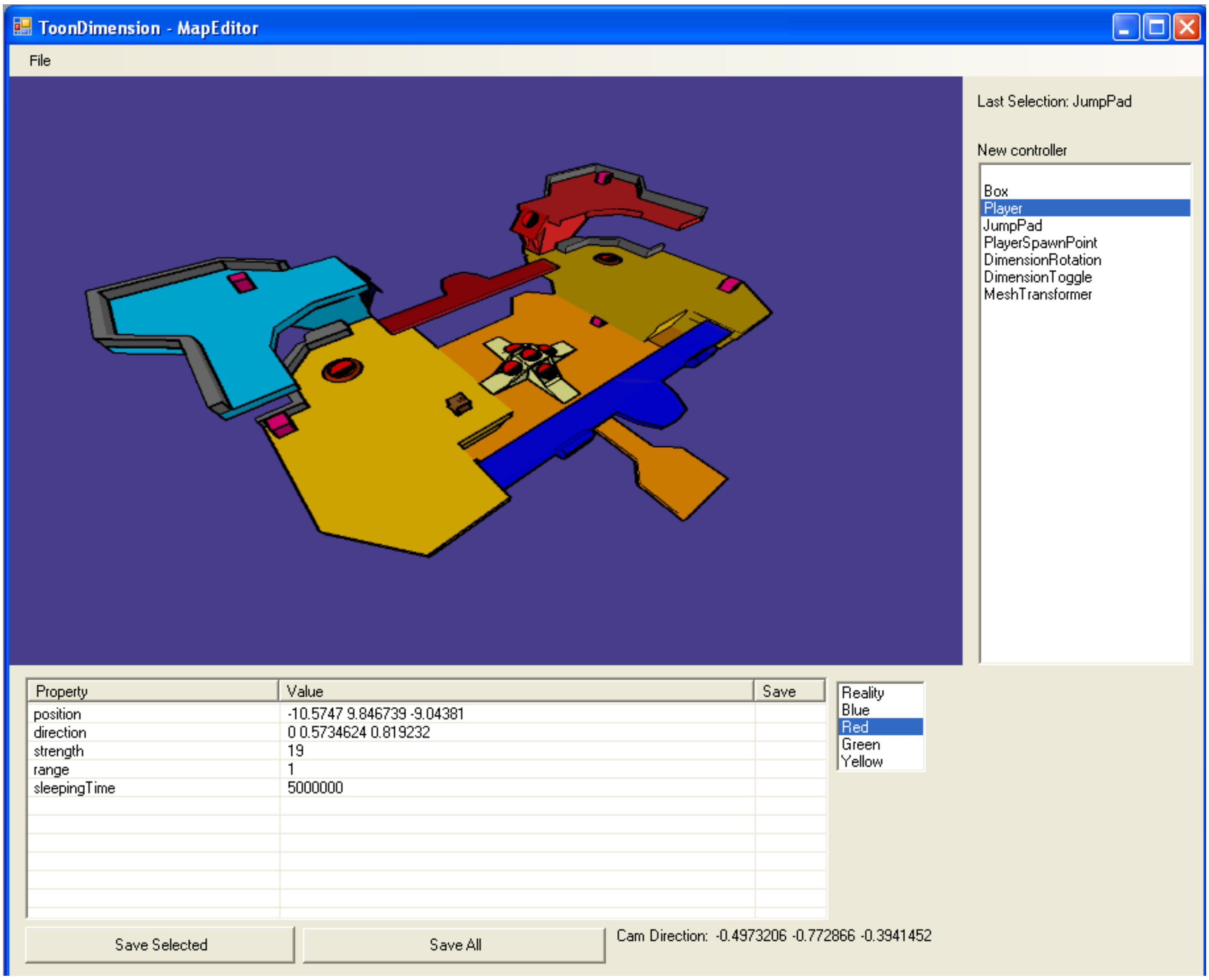


Figure 3: Introduction map

1.4 Map Editor

The map editor has made a great leap forward. We can now load and save maps on the fly, create new controllers, move them around freely and assign parameters to individual or multiple objects at the same time. The initial decision to link the editor that close with the game itself that forced us to include some editor exclusive functions into the game really paid off. Like changes to controller attributes are instantly visible and can be easily compared. Populating and changing the map can now be done in a fraction of the time we would need editing XML files by hand.



Last Selection: JumpPad

Property	Value	Save
position	-10.5747 9.846739 -9.04381	
direction	0 0.5734624 0.819232	
strength	19	
range	1	
sleepingTime	5000000	

Cam Direction: -0.4973206 -0.772866 -0.3941452

Figure 4: map editor

1.5 Sounds

We tried to include as many sounds as possible. A lot of them are mere place holders. Especially those that come from other games.

1.6 Characters

The player is now represented by an animated character. In this release only one character model (the bomberman) is included, but instances of this model can have different colorization, which is needed for indicating that a player is in a certain dimension. Character modeling (meshs & materials) and character rigging (i.e. bone and skinning setup) was done in 3ds max, while the skeletal animation was produced in MotionBuilder. Animation is initially solely data-driven due to low performance costs and visual convincing results. The character is able to walk in a prescribed direction and tries to match discrete position constraints produced by the player controller system.

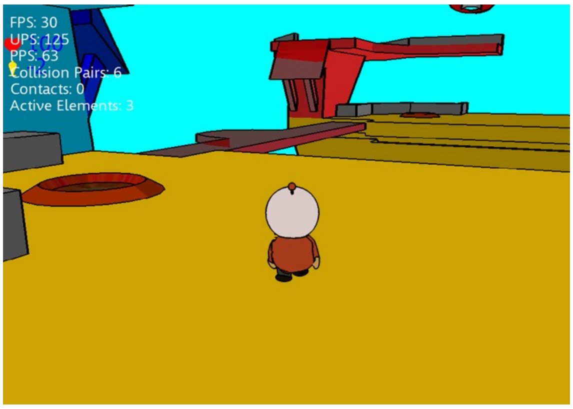


Figure 5: player

2. Problems

2.1 Physics and map limitations

We've encountered one problem with the physics, where the player can climb up on some unwanted positions, like small edges. This results in the player being able to leave the map. On one hand we can address this by creating maps that are designed to make the player be fully 'in a box' or by introducing invincible barriers. The current story map is designed like the first approach and the multi-player map is fully open but will punish the player for dropping off the platforms. We hope to design future maps in such a way that we can combine both approaches, because using only one makes the player feel contracted in an area or restrained by unnatural borders. The later will require us to justify the restriction with props like fences or handrails.

2.2 Implementation Issues

On the Xbox the shaders didn't behave like expected at first. Since all render targets on the Xbox share the same memory, old content is deleted on calls to 'setRenderTarget'. The persistent render targets supported in XNA 3.0 seemed to cause much trouble, so we did a work around.
A further implementation issue is caused by the multi threading. It is absolutely necessary to have at least three threads for performance reasons. However, ressources allocated in one thread may cause problems when used in another thread. Additionally there are the known criticall section and dead lock problems.

2.3 Modelling

2.3.1 Maps

We underestimated the time needed to create a map mesh. Even when the mesh is more or less finished, it needs a lot of tweaking and testing afterwards when it doesn't play right or allows to player to find glitches. For the coming development cycle we hope to reduce the time needed, due to the gained experience with our engine, the tools and the less elaborate texturing.

2.3.2 Characters

Successfully getting a character into the game is mainly about managing the complex data structures correctly. Import and export filters of DCC tools tend to be implemented rather lousy, or they are not able to match their internal data representation. At the end of the content pipeline some adjustments had to be programmed to account for the problem that different characters or different software tools produce varying results.

3. Revisions

3.1 Game Design

There wasn't enough time to add a second weapon as we wanted for the low target. But it's still planned to be included for the final product. Besides this we don't see the need to change our goals. In case the time would get short, we would probably first drop a story map as those require quite a large amount of time to be modelled.

3.2 Schedule

The only notable change in the schedule is the reduced time for creating textures. This is the case because as for now we didn't need very complex textures and noticed how we could create somewhat retro, but still quite unique graphical-style that also matches the one presented in our prototype.

4. Schedule

The term (x/y) stands for x hours expected, in y hours realized. Green entries are additional work which has been completed, red terms are partially or not at all complete.

Date	Course Items	Peter	Christian	Nicola
24.02.		- FGP (3/3) - Sketches(2/2) - Mock up Scene (5/7)	- FGP (-/8)	- Design (-/4) - Skeleton (-/4) - Physics (-/16) - FGP (-/8)
03.03.	Formal Game Proposal	- Simple Map(10/8) - Meshload (5/4) - Ghost cam (-/3)		- Character Controller (20/15) - Power-Up Prototype (-/10)
10.03.	Mutual Project Critiques	- Project Critiques (1/1) - Simple Map editor (20/20)	- Project Critiques (1/1)	- Project Critiques (1/1) - Dimensions (15/15) - Box Controller (4/10)
17.03. 09	Game Prototype, Functional Minimum	- Map-Modelling (5/3) - Map editor (15/13) - Refactoring (-/2) - Separating editor solution (-/5)	- Character modelling (12/4) - Prototype video (-/24)	- Bomb Controller (20/15) - Refactoring (-/15)
24.03. 09		- Textures (5/1) - Mapeditor (15/15) - DM-Map (10/10)	- Animated character (12/6)	- Menu (20/12) - Character Controller (-/8) - Shader (-/3)

31.03.		- Textures (5/-) - DM-Map (15/-) - Introduction Map (20/20) - Map-Save/Load (3/3) - Mesh-Controller (4/4)	- Animation clip production (20/5)	- Multi Player (20/10) - Character Controller (-/10) - Pre–Alpha Tests (-/2) - Shader (-/3)
07.04.	Interim Report, Low Target	- Sound Effects (10/10) - Props-Modelling(10/3) - Map adjustments (5/5) - Interim Report (3/3)	- Realistic locomotion generation (20/20)	- Power-Ups (10/5) - Weapon (5/-) - Interim Report (5/1) - Interface (-/3) - Model Binding (-/5) - Menu (-/5)