



BATTLE BALLS

FINAL REPORT

SUMMARY

The game “Battle Balls” is a multiplayer party game. Each player is represented by an alien-vehicle in form of a tremendous ball. The game world is restricted to a rectangular area that represents a block of a giant city. At the edges of this rectangular game world are holes. The goal of the game is to survive battles between these alien balls. Players can push the other players off the game platform into the abyss or kill them using one of the power ups that are hidden in the game world.

Project Team: Daniel Egger, Matthias Bühlmann, Benjamin Berger, (All at ETH Zurich)

BATTLE BALLS

Final Report

CONTENTS

1	Introduction.....	3
1.1	BACKGROUND STORY.....	3
1.2	Overall Gameplay.....	3
1.3	Genre and related games.....	4
1.4	Uniqueness.....	5
2	Design.....	6
2.1	Concept arts and mock-ups.....	6
2.2	Game Elements.....	9
2.3	Technical Elements.....	14
3	Development.....	16
3.1	Project Phases.....	16
3.2	Special technical features.....	19
3.3	Technical difficulties.....	20
3.4	Used Software tools.....	21
4	Playtesting.....	22
4.1	Sessions.....	22
4.2	General conclusions.....	22
5	Conclusion.....	24
5.1	Biggest technical difficulty during the project.....	24
5.2	What to do differently in the next game project.....	24
5.3	your greatest success during the project.....	24
5.4	Are you happy with the final result.....	24
5.5	Did you meet your project plan and milestones.....	25

BATTLE BALLS

Final Report

1 INTRODUCTION

1.1 BACKGROUND STORY

Some strange aliens who are addicted to playing games are coming to earth. They play a game called "Battle Balls" which is similar to a battle royal in wrestling. These Aliens really love mass destruction so they cut out a part of a big city and lift it up to the sky to use it as their play field. Directly around this field is "nothing" than the sky so that the gladiators in their fighting machines can fall out of it.

1.2 OVERALL GAMEPLAY

The gameplay and the goal of the game are easy to understand: During some battles, the players can destroy others by attacking those using weapons or push each other off the map into the abyss. Hence, a player just needs to survive as many battles as possible to win the game.



(In-game screenshot, two dead players in UFO and one ball just exploding)

BATTLE BALLS

Final Report



(Screenshot of another four player battle, of the actual game)

The buildings on the map can be destroyed by crushing into them with the player balls. Inside these buildings are hidden power-ups that appear after the building collapsed. These power-ups are mainly weapons to hurt and kill the other players. But there are also some others power ups, explained later.

The balls have a mass-property that give them more power in collisions with other balls and also the buildings collapse faster when colliding with a heavy ball that has a lot of mass. So it's important to have a ball with as much mass as possible to not get pushed away too much by other players. It's easier to survive if one is heavy.

But the mass is even more important, because it can also be transformed into some extra power, so that the ball is much more agile and faster to control. But this feature burns mass and the ball shrinks the longer one uses this mass transformation. To regain mass, one has to destroy a building and collect the falling building parts. Then the ball grows again.

1.3 GENRE AND RELATED GAMES

One can order our game in the genre of action games with some elements of game of the fighting genre. The basic idea of our game is similar to these other party games called "Micro Machines", "Super Mario Karts" and "Bomberman" where one can just distribute controllers among the attending players and immediately have fun together for quiet a long time. It is important to note that these games are all very simple regarding the controls, so

BATTLE BALLS

Final Report

that even a player who's new to the game can immediately engage, without having first to learn about how to control his game character.

The main factor why these games are fun is the personal competition between the players, so that in the end everyone wants to be better than the rest. This gives the winner sort-of satisfaction. The other fun thing is the possibility to "knock others out" of the game. That is especially fun when all the players are sitting together in the same room and knocked out players can insult their tormentor and vice-versa the others can laugh at the losers.

Our game also combines some strategic element (mass/energy dilemma) in the basic party game gameplay so that it will not become boring too fast even for experienced players, what was the major drawback in the games mentioned above.

The mass collection feature and the idea of "giant balls in a city" could in some way be related to the game called "Katamari".

1.4 UNIQUENESS

Our game has some special things that distinguish it heavily from other comparable, related games:

- We have a destructible map with nice looking building collapses
- We have a dead mode where also "dead players" can still engage into the battle
- We have a wired game setting with alien balls and the captured city

BATTLE BALLS

Final Report

2 DESIGN

2.1 CONCEPT ARTS AND MOCK-UPS

In the beginning of the game design process we created a lot of concept arts and mock-ups, to get an idea how the planned effects and game elements should look like.

The idea was to create all the models as appropriate as possible in advance, so we could easily reuse the produced concept artwork for the final game and don't need to create another collection of models.

2.1.1 GAME MAP MOCK-UP

First we created a concept art of the whole game play.

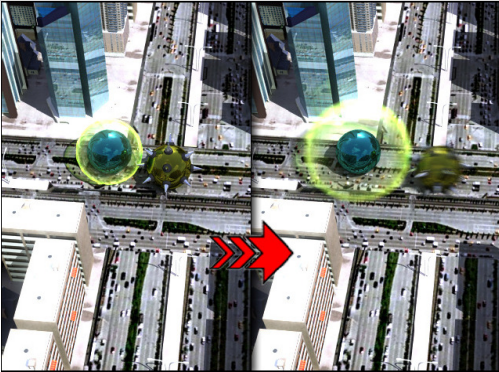


2.1.2 POWER UP MOCK-UPS

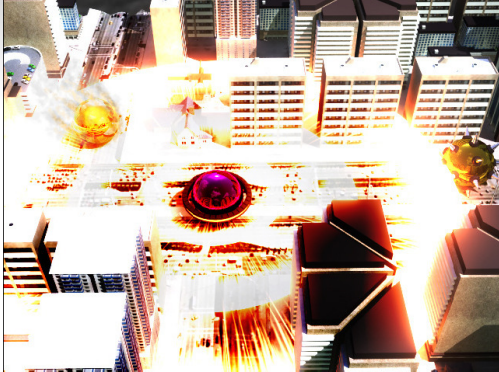
We also created a lot of mock-ups how the planned power ups could look like, even we then later decided to not implement all of them:

BATTLE BALLS

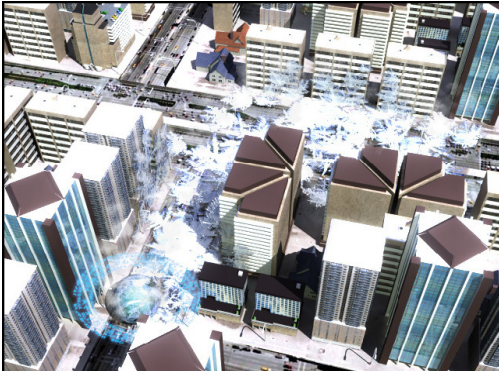
Final Report



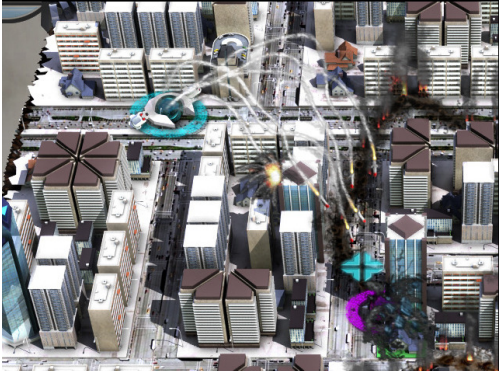
Energy Bumper



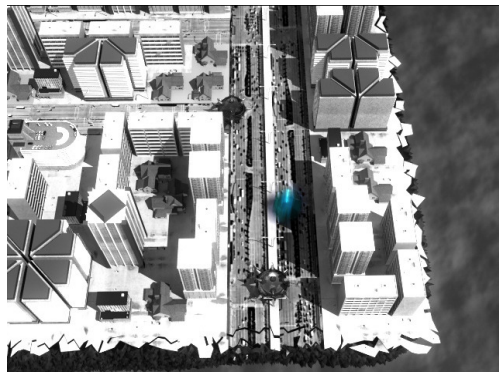
Fire Thrower



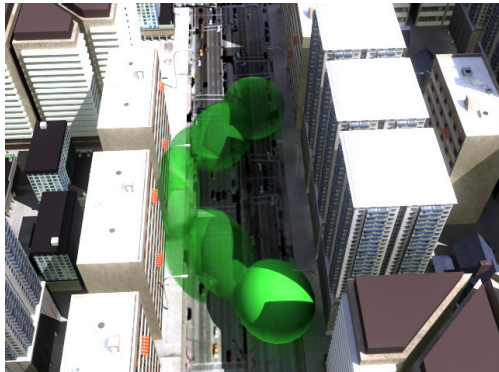
Ice Ball



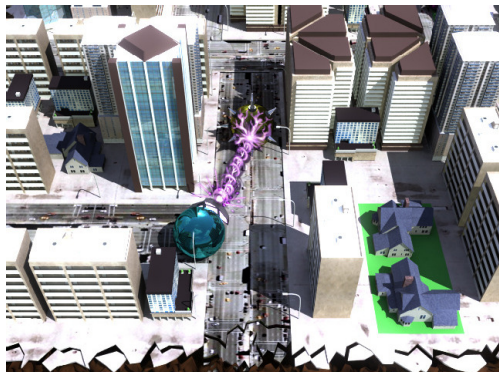
Rocket Launcher



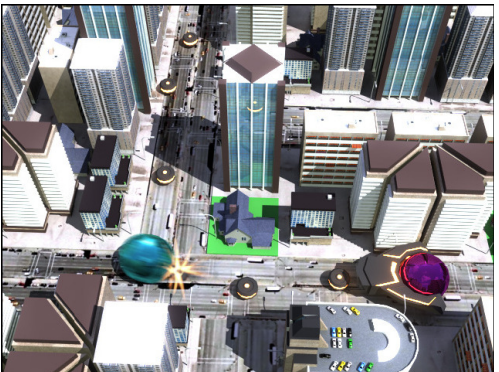
Hyperspace Controller



Bouncy Ball



Dark Matter Absorber

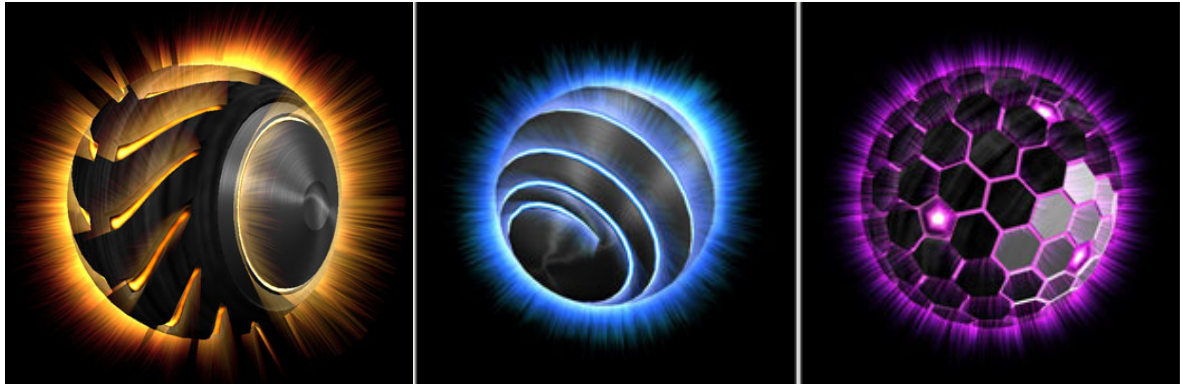


Mines

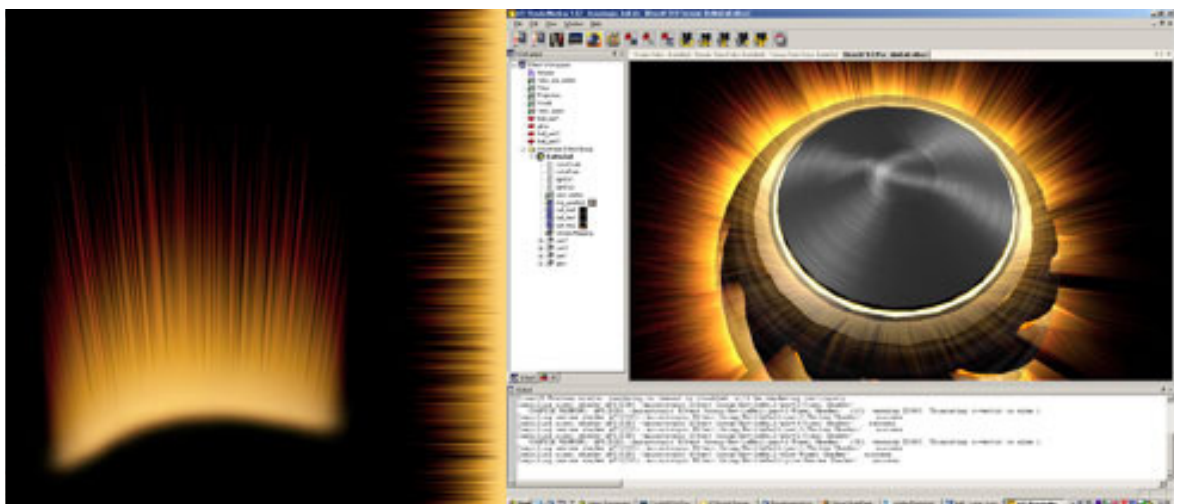
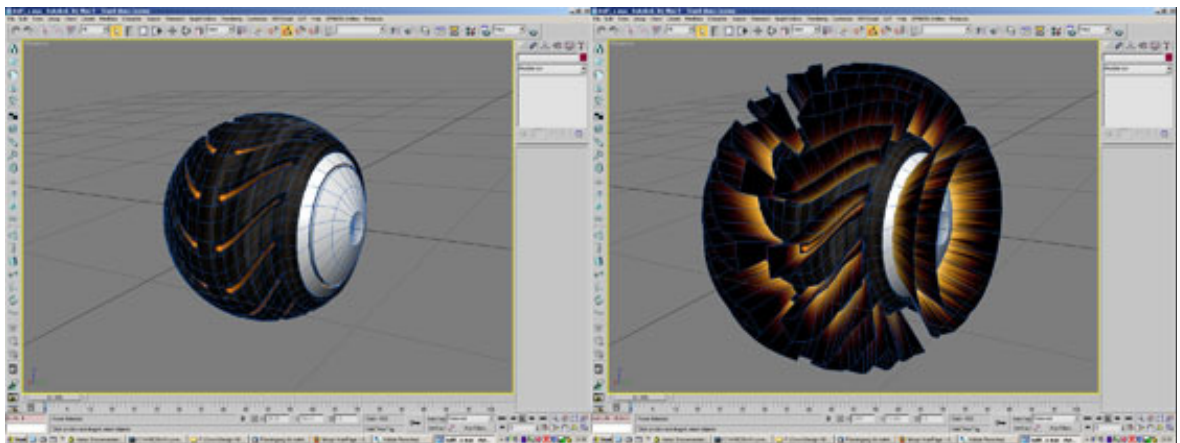
BATTLE BALLS

Final Report

2.1.3 PLAYER BALL DESIGNS



This mock-ups above are rendered with “Render Monkey“. The models and the effects were done in 3D Studio Max as you can see below, where also the single steps of the design process are documented. First image to the left: the pure ball and to the right the ball inclusive the “glow model” around it, rendered without alpha blending.



Left side: the glow texture

Right side: The final result

BATTLE BALLS

Final Report

2.2 GAME ELEMENTS

In this section all the logical pieces of the game are described in detail.

2.2.1 PLAYER

Each player can control one of the balls. Up to 4 players can challenge each other. The ball of the player can't be controlled directly by the joystick of the pad in the sense of directly moving it around, but each player has control over the acceleration of the ball. This will lead to a somewhat indirect control over the ball which is important for the "feel" of the game.

2.2.2 BALL

Each player has three main resources, which are related to his ball:

- The ball's mass
- The ball's health
- The maximal power that the player can act on his ball

The ball's mass and his maximal power are crucial for the steering behaviour and the mess the ball can produce. *The destruction vehicles are powered by dark matter, which is present everywhere in space, but cannot interact with normal matter except for it's mass. A main part of the balls interior consists of dark-matter collectors, which absorb dark matter and constrain it to the body of the ball, increasing it's absolute mass.*

To cause more damage, the player must rise the mass of his vehicle. The bigger the mass of a ball the more power one needs to still be able to control the vehicle. On the other hand the player has only limited maximal amount of power, that he can act on the ball. Therefore one conflict of the game is to have a good balance between power and mass which leads to a balance between the controls a player has over its destruction device and the damage the ball causes on other players and the environment.

Mass

The gain of mass happens by transforming bright matter into dark one by collecting destructed building parts and rubbish around the map. This is mainly done by placing the ball under a collapsing building, so all the part fall onto the ball and it is able to absorb the parts and transform it into mass.

It was a high target of the project to create buildings that produce rubbish to collect, but we saw that this was easy to implement, so we directly implemented it, instead of creating a kind of a mass-power ups as planned.

It is important to note, that the masses of the balls define the ability to push others away at a collision (i.e. the heavier ball will of course have more chance to push the other around).

Power

The player is able to spend up to a maximal amount of power as a force to the ball to control its movements. This can be done by holding the primary analogue stick with an appropriate amount in the desired moving direction. How much force is influencing a ball is visualized by a glow halo around the ball. The player has a base force he can act on the ball and additionally the balls have energy converters, which allow transforming some of

BATTLE BALLS

Final Report

the dark matter into energy. This energy boosts up the power, a player can maximally bring up to steer the ball. This action is done by the analogue shoulder key of the gamepad with that he controls how much additional force is converted. That gives temporarily more control over the vehicle (faster response) but also decreases the mass over time.

Health

During the play tests of our game we noticed it would be fun if the weapon power ups are not only enable the players to push other off the map, but instead the balls should get destroyed when getting hit by a weapon. So the goal is now not just to push other players off the map as we first planned. Now you can also kill the others by destroying their ball by taking all of their health by attacking them with weapons. If a ball loses all of his health, he turns into a huge explosion and the player disappears.

2.2.3 POWER UPS

One of the main reward factors of the game are the possible power-ups the players can gain. They are graphical eye candies and give the player the feeling of being powerful and mighty. The power-ups in the game map are hidden in the buildings of the city. (Remember, the game map consists of a rectangular part of a city). A player has to destroy a building to uncover its hidden power-up. Then he is able to collect that power-ups by touching it with the ball.

All the power-ups have their unique gameplay feeling. The power-up a player currently holds is displayed in its status bar visible for all players. The player's ball is also equipped by a nice looking model that represents the current weapon. By pressing the button a power-up can be consumed and its abilities are released immediately. Some power-ups give new possibilities of controlling and other are passive and act automatically as the ball moves.

The number of power ups of each kind should be always the same, so that there is no map with more power-ups of kind power-up than of kind power-up Y.

Here are the power-ups we implemented in the final game:

Fire Thrower: The fire thrower is able to burn enemy balls and overheats them by force them to touch the fire burst. When the enemy is overheated his ball drives in crazy mode where he is not able to stop the ball anymore and it accelerates with full speed and also the controls of the player are inverted. Like that it is very hard to stay on the battle field.

Rocket Launcher: Several rockets get shot up to the sky and launch into your target. While the player is aiming toward the target of the rockets, he can't steer the ball anymore until he releases the aim control of the rocket launcher. Rockets can destroy buildings, and also will take some health from the ball that gets hit.

Energy Bumper: An energy-deflector around the ball which can be expanded at the push of a button to bump away other vehicles, hopefully into the abyss.

Hyperspace: Stops the time for the rest of the world. The player that gets this power up will remain the only colored ball and the whole rest of the map turns into gray and everything stands still. In some little time the only thing that moves is the players ball, all other stand still. So the player can position his ball optimal for a knock-out when time continues.

BATTLE BALLS

Final Report

Mines: At the balls it opens a little hole where the player can put out mines on the floor. After that, when a ball hit such a mine, the ball loses some health by the produced explosion of the mine and it get bumped away in the direction of the ball / mine distance.

Machinegun: As the name says it, you can fire on your opponent with a heavy gatling gun. The ball and also the buildings that get hit by the bullets of the gun loses some health. Its not easy to aim with that power up , but when it hits the target, it produces an ugly mess.

2.2.4 MAP

Initially the game map consists of some buildings that contain random power-ups, some streets and in some previous defined spawn positions the player balls. The building can be destroyed by a player by crushing into it with the ball or by using some of the weapon-powerups (see above). The destruction of the building look very realistic, consisting of our building collapse physics together with a good graphical particle system. However - the destruction physics for the buildings are only eye-candies and not part of the gameplay itself. The falling building parts are just to regain mass for the ball, and do not affect the game in another meaning.

The map may also contain holes, where players can fall into. The map is not forced to be rectangular, but its built up of little quadratic pieces. So the shape will be mostly rectangular like.

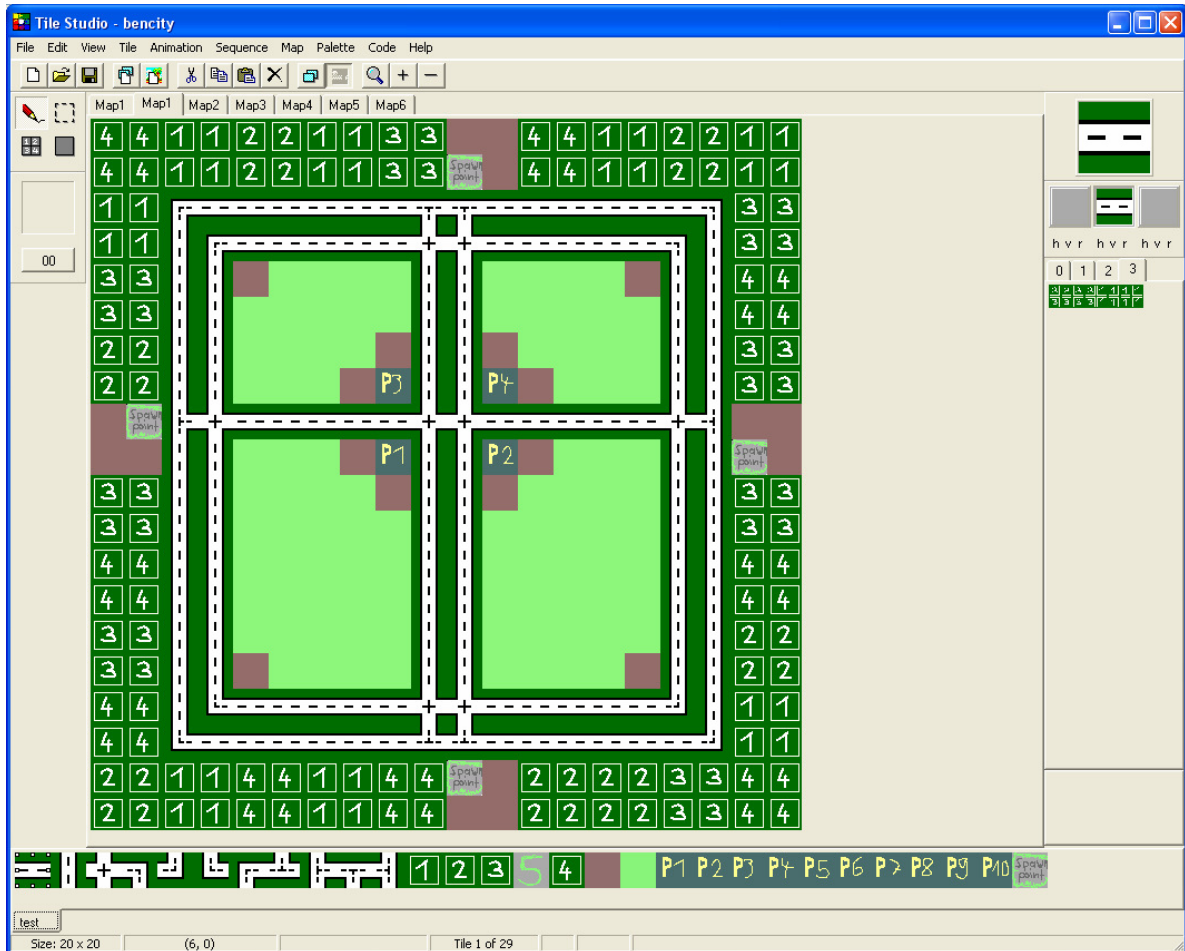
We included about eight different maps into the game, that are chosen randomly when a new round starts. (6 examples below)



BATTLE BALLS

Final Report

2.2.5 MAP EDITOR



We used a little shareware tool called “Tile Studio” to build up our maps. It is a little tile editor, where one can edit a 2D map and fill up some tiles with predefined textures, that represent streets, buildings, spawn points, holes etc... After designing the map one can export the 2D map project into a file, based on a predefined rule set, that defines what format that file should have. We then parse such files and load a map corresponding to their definition. Such a file looks like that (only first half of the file below):

```
12;12;2;2;13;13;13;13;13;2;12;12;13;12;12;13;12;12;13;13;13
12;12;2;2;13;13;13;13;13;2;12;12;13;12;12;13;12;12;13;13;13
12;12;2;2;13;13;13;13;13;2;13;13;13;13;13;13;13;13;13;13;13
12;12;2;2;13;13;13;13;13;2;13;13;13;13;13;13;13;13;13;13;13
12;12;2;2;13;13;13;13;13;2;13;13;13;13;13;13;13;13;13;13;13
12;12;2;2;13;13;13;13;13;2;13;13;13;13;13;13;13;13;13;13;13
12;12;2;2;13;13;13;13;13;2;13;13;13;13;13;13;13;13;13;13;13
12;12;2;9;1;1;1;1;1;3;1;1;1;1;1;1;1;1;1;1;1;1
12;12;2;2;13;13;13;13;13;2;13;13;13;13;13;13;13;13;13;13;13
12;12;2;2;12;12;13;13;13;2;13;13;13;13;13;13;13;13;13;13;13
```

BATTLE BALLS

Final Report

2.2.6 MULTIPLAYER GAME CYCLE

We planned to implement different game types to choose from, that specify the long term goal of the game. It would be possible to integrate team based game types, but for the moment the most obvious and the only implemented game type is “championship”, where all the players fight each other and try to win as many battles as initially specified, for example 10 battles. The best performing player over all rounds e.g. the one that survived the most battles then wins the whole championship and hence is the champion of the game.

BATTLE

A battle begins and all players can move around and collect power-ups by destroying buildings etc. as described above. The battle is over when there is only one final surviving ball on the map that will win this round. After each battle, the game loads a new map in its initial state, all balls respawn at initial positions and a new battle begins. This cycle continues until the championship is over.

GHOST OR DEAD MODE

As soon as a player’s ball gets knocked off, he changes to the ghost mode where he’s just able to indirectly influence the game by firing from time to time a massive ion- cannon (mounted to an UFO, where the player’s ball sits in). This destroys the buildings at the target position and breaks the city ground. This produces a new hole in the terrain, where players can fall into. The target position is first marked for some seconds, before it finally fires. So players can be aware of newly appearing holes.

Like that even the already finished players can still have fun by destroying the city and try to influence the others play instead of just boring wait until the battle is over. To be able to fire the cannon the player must wait until his ball lights up, and the ball is ready to fire.



UFO that uses its ion cannon.

2.2.7 SINGLE PLAYER

We wanted to implement also a single player mode with some enemies that try to knock the player out, controlled by an AI. The primary goal of the single player mode is to practice for the next multiplayer match against other, very skilled human opponents.

Unfortunately the AI is kind of complicated to be implemented in an intelligent manner and the AI balls don’t just fall off the city. So we decided to implement the AI a bit later, just for the submission at the Microsoft contest. It doesn’t matter we didn’t do it so far, because we wanted to create a nice multiplayer game, and that’s what we got.

BATTLE BALLS

Final Report

2.2.8 GAME DISPLAY

The game is in 3D and played in a vertical top-down perspective in which you can see all players participating in the game (we want to avoid split screen, since we don't want that every one "plays a multiplayer game alone" but that the action is very concentrated and the players are always in direct conflict). During the game the view direction of the camera will be fixed and is not moving, to let the player keep their orientation.

2.3 TECHNICAL ELEMENTS

2.3.1 GAME LOGIC AND STATE MANAGEMENT

One of the major elements of the game is of course the game logic that consists of the gamestate management (championship, begin battles etc.), handle dead players (ghost mode), definition how the power-ups influence the player and the world (e.g. different control style while guiding a missiles), controller input, force feedback and everything else that's not covered by the other points below.

2.3.2 BASIC PHYSICAL SIMULATION

This point covers the "world physics", that are crucial for the gameplay of the game. This contains the general Collision Detection, Ball Physics and the Deformable Terrain.

We plan to not implement a general purpose physics engine for that project because of the very limited available time. We primary just simulate the balls and handle collisions between them and the rest of the world, because these are the only rigid things that move in the scene. We will implement the ball as a point mass (not a simulated sphere that rolls), that receive a force controlled by the player. We need to implement collision detection and the responses primary for the following cases:

- Ball/Ball collision - Calculate appropriate bounce for both balls
- Ball/Building collision - Damage the building at the hit point and bounce the ball
- Ball/Ground collision - Simple heightfield query at the balls position

For other purposes of the gameplay where physics matters like the power ups, which some also need some collision detection and physical behaviour (rockets, mines, ...) we will implement special, sometime not necessarily physically correct physics, because our main goal is to create a fun game, not a scientific correct physical simulation.

2.3.3 RENDERING SYSTEM

The rendering system primary needs a scene organization like a scene graph where all the displayed objects can be organized in a hierarchical way. We will also address the generations of shadows (in a high target we would like to implement shadow volumes that even influence particles). Another element is a proper "Render Pass Manager" where we can define different passes for a material, which allow a nice integration of shaders into the system.

BATTLE BALLS

Final Report

2.3.4 PARTICLE SYSTEMS

One main technical element will be a particle system. There is a lot of action on the screen, like buildings which are destroyed and the unleashing of some powerful powerups. All these things can be displayed with particle systems. This will lead to a flexible system which can handle all of the required uses like rings of fire dust of destructed things and react on spacewarps like near explosions, wind and so on.

We will first implement a prototype that evaluates the challenge of implementing a particle system based on a vector field that can influence the particles for example by an explosion.

2.3.5 BUILDING PHYSICS

Another main part of the game's rocking appeal consists of the destruction of the buildings. To accomplish this we implemented an appropriate physical simulation. But our main goal is to make this look good and not to make it act realistically. Once the building is destroyed it will not have any impact on the game.

We restricted ourself to this as we know that it is very hard to implement a physical system which works and is realistic. We built it up until it looked good enough for our purposes by playing around with it.

BATTLE BALLS

Final Report

3 DEVELOPMENT

During the development process we had different phases. They are listed in chronological order, with the first one at the beginning:

3.1 PROJECT PHASES

3.1.1 BRAINSTORMING PHASE

Where we defined the whole game how it should work etc. We also created a gameplay prototype in Adobe Shockwave that demonstrated us the game in a simple form to evaluate the different ideas we had.

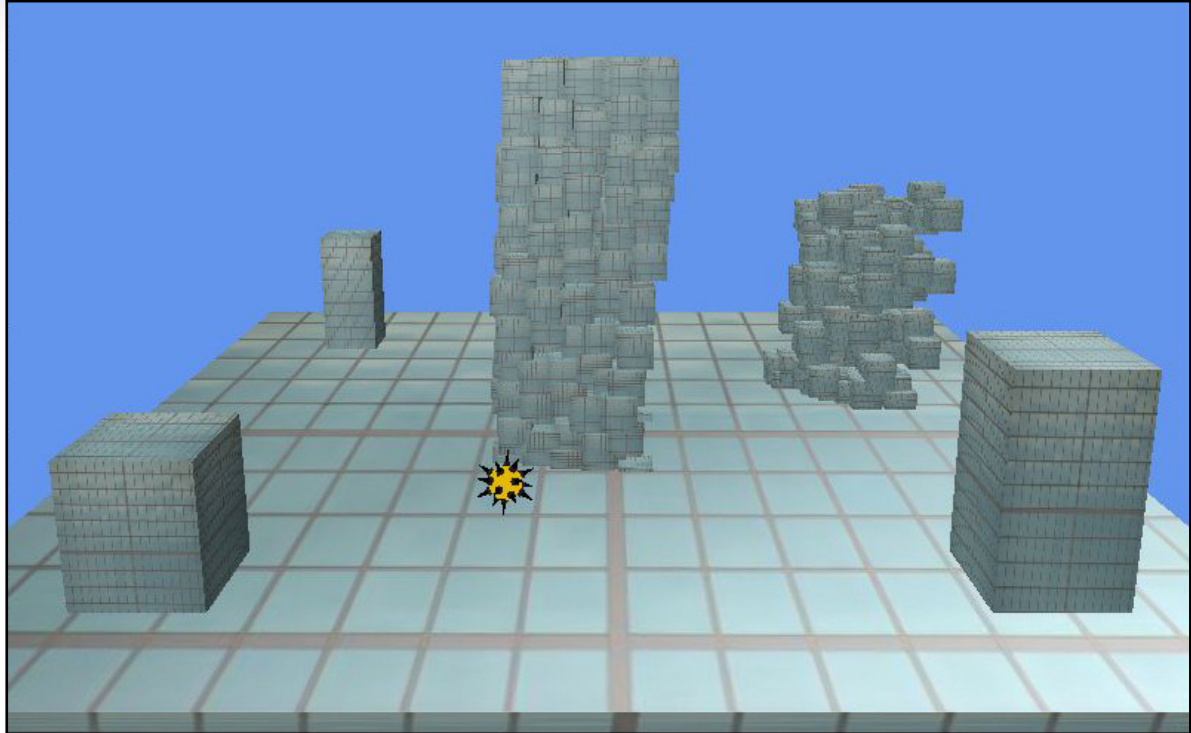


3.1.2 PROTOTYPING PHASE

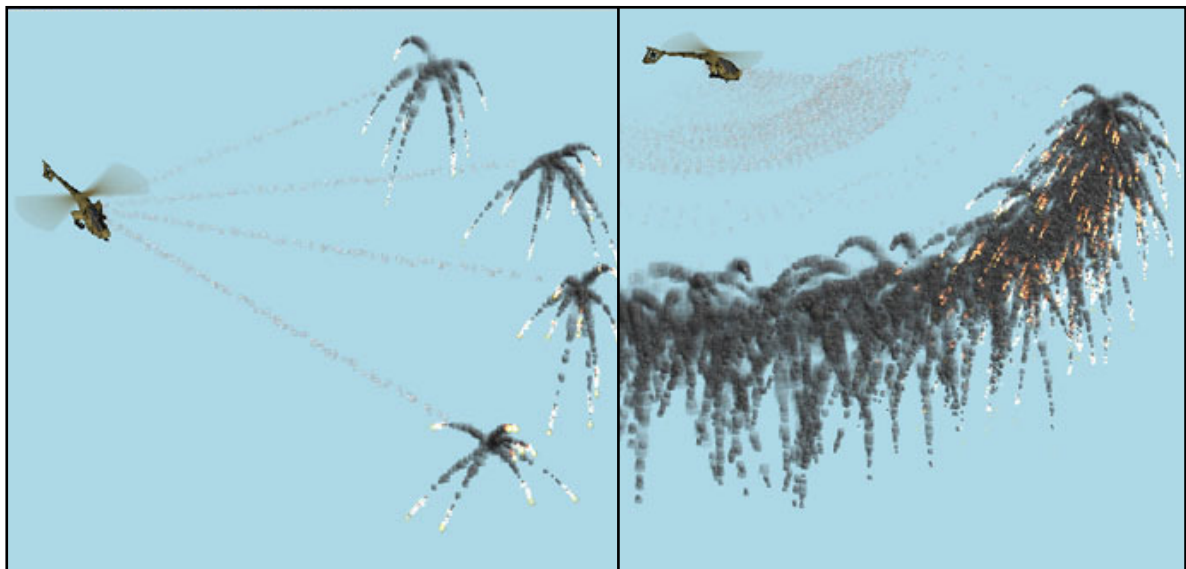
We split up the project into two main technical difficulties, about that we wanted to be sure that we are able to implement it in the final game. That was: the building collapse and a particle system that can react on users movements and explosions etc. So we created two more prototypes, but this time already in C# using the XNA framework.

BATTLE BALLS

Final Report



Screenshot of the building collapse prototype



Screenshot of the particle systems prototype

BATTLE BALLS

Final Report

3.1.3 LOW TARGET

The goal for the low target stage was, that in addition to the above gameplay Shockwave prototype, the game board will look more like a city. The particle system is implemented and it will look good if a player destroys a building. Also some of the weapon power-ups are implemented and will work well. We also implemented 2-3 power ups that already work fine, but maybe do not look perfectly.

After we prepared everything with our prototypes and long brainstorming discussions, we were able to implement the final game. It needed a lot of code and also a lot of artistic artwork in 3D modelling software and Photoshop etc. to finish the game. We first created a basic engine, that was able to:

- Render the scene with a ball
- Create and load a map out of a file
- Collision Detection between player balls and buildings
- Implemented the building collapse from prototype
- Player input handling / ball movements

In the end of that stage we was able to test and tweak some behaviours of the ball and also make the collapsing buildings look more realistic.

3.1.4 DESIRED TARGET

After we created the main game engine we were able to implement the parts of the gameplay like for example the powerups and also the logic of the game like the gamestates.

In the end of the stage “desired target”, the game board will look like a city with a lot of high quality models and used shaders. The sounds and the music are implemented and work well together with the rest of the game. The game is quite balanced and it will be enough fun to spend some time with it and is also fun as the player gets more experienced.

Also all the necessary screens and highscores are implemented, the player management works well in the sense that each player can join a game and its won rounds are counted etc. In a sentence the game is finished and looks like a game.

The end of that stage was the alpha release, which we demonstrated to the class.

3.1.5 HIGH TARGET

As the high target we defined in the game proposal: The crowd simulation is implemented and the city will look like a living city moving cars and people. Also there is more realistic looking rope-simulation implemented for the hookshot and the particle systems react on spacewarps and deflectors. Also the collection of building parts, that fall to ground will be implemented as high target. There is also a simple single player mode implemented in the high target, which is maybe the most important part of the high target.

We didn't implement the singleplayer mode and also not the pedestrian simulator and the hookshot. But we got a particle system that react on spacewarps and we implemented also the collection of mass by colliding with building parts.

BATTLE BALLS

Final Report

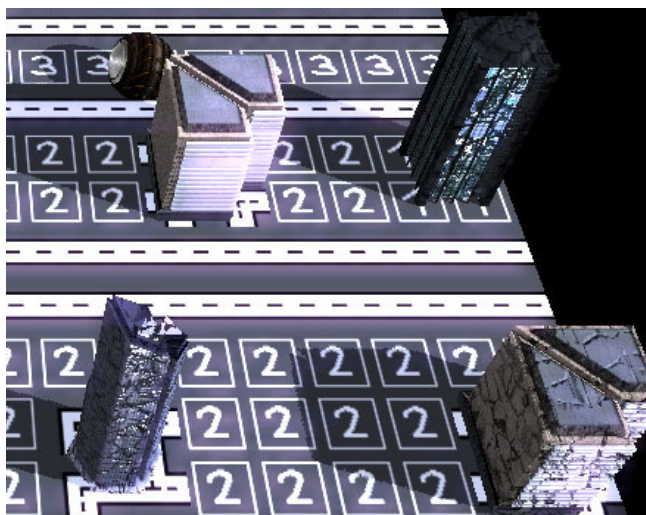
3.1.6 EXTRAS

We specified in the game proposal: The physical simulation is expanded and the things which you can do with it like destroying buildings and throwing parts and/or cars to each other would have an impact on the gameplay itself. – We didn't implement that.

3.2 SPECIAL TECHNICAL FEATURES

3.2.1 BUILDING DAMAGE SHADER

To easily simulate and visualize the building damage, we implemented a shader, that creates a crack pattern over the texture of the building to show that its going break soon.



3.2.2 METAL PLATE OF THE BALL

For the metal plates of the balls we created an anisotropic shader using the paper "Efficient Rendering of Anisotropic Surfaces Using Computer Graphics Hardware".



3.2.3 SIMPLE SCENEGRAPH

We have implemented a simple scene graph which handles all our graphics with shaders. It includes a global matrix stack similar to the one in OpenGL. The main goal of this scenegraph was to support model and inter-model hierarchies. The scenegraph works very well. It took quite a long time (more than 2 whole days) to debug it and make it really work. See also the "technical difficulties" below which explains the problem we had with the bone hierarchy. This problem alone took more than one day to resolve. The implementation we have now works very well.

BATTLE BALLS

Final Report

3.2.4 FLEXIBLE INPUT SYSTEM

We implemented a flexible input system. Our input-system abstracts away the difference between the keyboard and the gamepad. You only need to change on line in the code to switch from gamepad- to keyboard-input and vice-versa. It is very useful to have such a flexible system because we do our main development work on a PC often without a gamepad in reach...

3.3 TECHNICAL DIFFICULTIES

3.3.1 GRAPHICAL PERFORMANCE PROBLEM

From the beginning our game had some performance problems and run quite choppy from time to time. But it wasn't too bad. We didn't pay too much attention to it according to the famous proverb "premature optimization is the root of all evil". But when we tried the game on the X-Box it was even worse. The game run very crappy with even less then one frame per second. And that although we only displayed 10 buildings with only 10000 polygons each (Well we know, that are a lot of polygons for one building, but the X-Box should still be able to handle this polygon count). After tweeking around a bit with our setup we could nail down the problem to the graphics.

It was only the when we discovered the NProf profiler. The profiler gave us some interesting results. It turned out that about 80% of time was eaten up by a call to the method `Effect::Begin()` although we almost never called it directly.

We discovered that we cannot use the code from the XNA help file "how to render a model":

```
private void DrawModel( Model m )
{
    Matrix[] transforms = new Matrix[m.Bones.Count];
    float aspectRatio = graphics.GraphicsDevice.Viewport.Width / graphics.GraphicsDevice.Viewport.Height;
    m.CopyAbsoluteBoneTransformsTo( transforms );
    Matrix projection = Matrix.CreatePerspectiveFieldOfView( MathHelper.ToRadians( 45.0f ),
        aspectRatio, 1.0f, 10000.0f );
    Matrix view = Matrix.CreateLookAt( new Vector3( 0.0f, 50.0f, Zoom ), Vector3.Zero, Vector3.Up );

    foreach ( ModelMesh mesh in m.Meshes )
    {
        foreach ( BasicEffect effect in mesh.Effects )
        {
            effect.EnableDefaultLighting();

            effect.View = view;
            effect.Projection = projection;
            effect.World = MyWorldRotation * transforms[mesh.ParentBone.Index] *
Matrix.CreateTranslation( Position );
        }
        mesh.Draw();
    }
}
```

The call `mesh.Draw()` to the `ModelMesh::Draw()` method will internal call the `Effect::Begin()` which is very slow even on the most current graphic hardware. Our building consist of about 100 submodels for the building parts. We found in a forum-entry that a modern graphic card can handle about 200 call to this method per frame. A number which we were vastly surpassing with our 11 * 100 submodels...

BATTLE BALLS

Final Report

We only need one effect file for the whole building model and when we eventually put the `Effect::Begin()` once per model and not once per submodel the performance of our game augmented about 5 times.

3.3.2 SHARING OF MODEL BONES

To animate our building when they get destroyed we use different submodels for the building which are connected with bones to the main model. After some time we succeeded to animate the model by changing the transformation matrices of these bones. But the success didn't last for a long time. As we were trying our code with different buildings we discovered that the matrices for the `ModelBones` are shared across all instances of the model. That means that if we destroy one building the other one although not connected at all with the first building we also get destroyed (at least graphically) because changing the bone-matrix of one model automatically changes the same bone-matrix of the other model. To keep our model animation code we had to manually copy the `ModelBone` hierarchy with our own `Bone` class. With this hack everything started to work.

3.4 USED SOFTWARE TOOLS

- **XNA Studio / Visual C# express** - Code
- **3D Studio / Maya** - 3D modelling
- **Photoshop** - Textures
- **Rendermonkey** - Render concept arts
- **Tile Studio** – Map design
- **XACT** – Sound
- **Tortois SVN** – SVN client

BATTLE BALLS

Final Report

4 PLAYTESTING

We did our playtesting in a rather informal way. One reason for this approach was that some key elements like the power-ups were not implemented as a whole in the playtesting phase. Another reason was that all the three team members did the playtesting separately with different person and user groups.

4.1 SESSIONS

Here is a list of the people we recruited to test our game:

- **team of 11 years old** (the brother of one of the team members and school colleague of the brother)
- **team of women**
- **fellow master students**

The 11 year old found the game quite entertaining. They didn't really care about the graphics but still had quite a lot of fun, although they didn't understand the main game concepts, to push the other players out the field, right from the beginning. But after a few rounds they did understand this concept and tried to win the game against each other.

Also the other play tester had in general a lot of fun. They often praised the graphical appearance of the game and they liked the simple game concept. The steering of the balls and the look and feel of the game reminded one of the game testers of the classic game Marble Madness which was one of his favourite games of the 80's. This was a very good compliment to our game, because it showed that we somehow succeeded in implementing our design goals.

4.2 GENERAL CONCLUSIONS

4.2.1 GOAL OF THE GAME

By watching our game testers play we found out that with our initial game maps and the initial game concepts the players some times tried to avoid falling down from and losing the game by camping in the middle of the game field. We tried to avoid such behaviour by implementing an additional game goal: Each ball has no some life points which are shown in an additional life bar on the status display. This life points will get lost by the effects of some powerups like the rocket launcher and the mines. If the life points are all gone the ball will explode which end in the same manner as if the ball would fall down the playing field.

4.2.2 MAPS

We also revamped the playing fields by adding some more interesting game maps. The new game maps also contain holes in the middle to avoid the camping behaviour of some players. Although the addition of the holes in the middle enhanced the gaming experience

BATTLE BALLS

Final Report

in a big way it also added some new problems, which were uncovered in other testing sessions: It was sometimes graphically not very visible where a hole is and some player would fall down the holes by accident. We plan to eliminate this problem by using some 3D-models to show the edges instead of the flat tiles we are using now.

4.2.3 WHICH ONE IS MY BALL?

Another thing we have seen in play testing sessions was that the players didn't know in the beginning of the round which balls they control, because some of the different balls look very similar.

4.2.4 WOMEN NOT THE FOCUSGROUP OF CONSUMERS

A last thing we noticed in the testing sessions was that the more players (up to 4) played the game, the more fun each participant had in the game. This is also one of the main reasons why we did our public presentation with four players.

As an interesting side remark one general impression that the game testing but also some other informal game presentation sessions revealed was that women which were not exposed to video games before were generally very unimpressed of our game. The game testing session showed that after about 5 rounds of play on of the woman in our testing group literally begged to not have to play anymore. Other women were only impressed by the cute sounds of the game. The conclusion of result is that we don't bother. :-) Women were not our target group in the first place. Still it is very interesting to notice that games solely designed by males which impress other men generally leave out a whole batch of a big potential user group...

One of the reasons this could be is because our game is still very fight and the goal of the game is to win by killing the others. Girls and women don't like action games that much this might be an explanation.

BATTLE BALLS

Final Report

5 CONCLUSION

All in all we can say that this project was a big success and also a lot of fun for us. The first place we got in from the public voting was a very nice profit for all the spend days and weekends!

Now let's go to the details of some aspects:

5.1 BIGGEST TECHNICAL DIFFICULTY DURING THE PROJECT

One of the hardest technical problems was to get started on XNA. It took some time and effort from our side, because the supporting material from the assistants came too late. Once we managed this hurdle we run into some problems on how bones were handled in XNA (see TODO).

A last and everlasting technical difficulty we ran into was the difference between graphics cards, namely differences between ATI and Nvidia. Our lead graphic programmer has an ATI-card and it turned out that some parts run differently on the Nvidia cards the two other member of the team run.

5.2 WHAT TO DO DIFFERENTLY IN THE NEXT GAME PROJECT

This project was a very valuable learning experience but as a matter of fact we wouldn't do so much things differently in another project. Maybe if we would have some more time we would invest a little more time into the design phase and try to make the code more reusable for future projects. But apart from that it worked out very well.

Maybe in another game project we would not use XNA, simply because it doesn't support online multiplayer capabilities yet. We would like to make an online multiplayer game in the future.

5.3 YOUR GREATEST SUCCESS DURING THE PROJECT

The greatest success of this project was to see our game idea evolve into a real fun and playable game. Starting from the concept on paper it was a very nice experience to see the idea grow, first into some tech demos and then into the (almost) finished game as it's now. The feeling of creating something presentable in this project was the greatest success.

This project was also a very good experience in team work. The distribution and the merging of the work load was very good. Each member was responsible for another part of the project which he could implement with the help of the other team members. Using software like subversion in a multi user environment was new for some team members but it worked out very well.

5.4 ARE YOU HAPPY WITH THE FINAL RESULT

We are very happy with the final result of our project and as the public presentation and the win of the "people choice award" showed also other people like our project! And that was finally our goal, so we consider our project as a great success.

BATTLE BALLS

Final Report

5.5 DID YOU MEET YOUR PROJECT PLAN AND MILESTONES

The project plan we made in the first few weeks was the driving engine of the project. The distribution of work and the initial assumed time line was surprisingly accurate except the hour estimation which turned out to be 2 to 3 times higher than expected (an old Software Engineering law proven again...).

We could meet our milestones more or less as expected.

In some weeks the schedule was very compressed and it worked only out because we invested a lot of time into the project and sometimes even neglected other ETH courses. If we wouldn't get 10 ECTS points from this course it would be too much work...

5.6 COURSE

We were happy to have visited that course and invested all the time into the project. It was nice and it was interesting to do such creative work during a course. Big thanks to the "gamelab team" to organize that course. We all hope the course continues...