

**Toon Dimension**

**Formal Game Proposal**

Peter Bucher  
Christian Schulz  
Nicola Ranieri

February, 2009

## Table of contents

1. Game Description.....	1
1.1 Idea.....	1
1.2 Story.....	1
1.3 Gameplay.....	2
1.4 Implementation.....	2
1.4.1 Design.....	2
1.4.2 Classification.....	3
1.5 Visual Aspect.....	3
2. Development Schedule.....	4
2.1 Team.....	4
2.1.1 Peter Bucher.....	4
2.1.2 Christian Schulz.....	4
2.1.3 Nicola Ranieri.....	4
2.2 Tasks.....	4
2.2.1 Functional Minimum.....	4
2.2.2 Low Target.....	5
2.2.3 Desirable Target.....	5
2.2.4 High Target.....	5
2.2.5 Extras.....	5
2.3 Schedule.....	6
3. Assessment.....	7

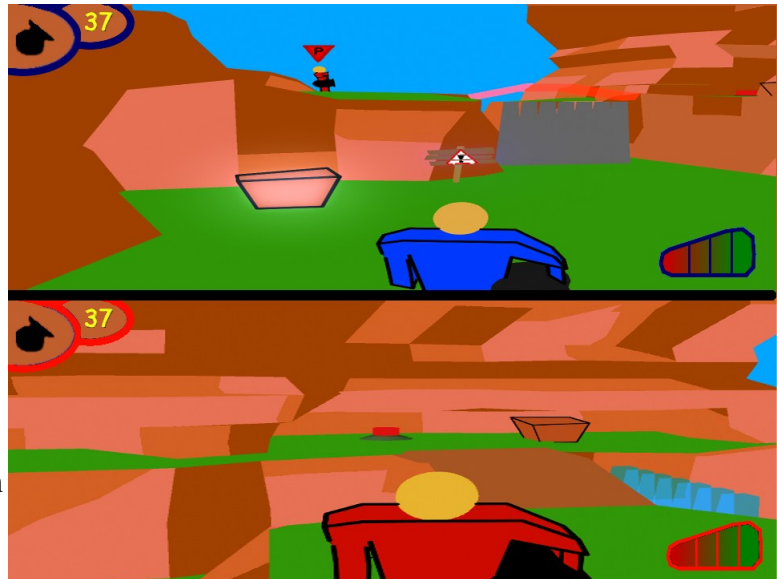
# 1. Game Description

## 1.1 Idea

Our goal is to create a game, in which multiple players try in a cooperative and or competitive way to accomplish a certain task. Each one takes control over a toon character and moves him in jump'n'run style through a map, collecting power-ups, defeating enemies and solving puzzles.

Each map consists of stairs, elevators, doors and interactive elements such as switches. It also contains enemies, which try to complicate the task of each player.

To defeat them and other players, each player has access to weapons as simple bombs or small toon bombs running towards the enemy.



There will be several power-ups, sometimes hidden in destructible boxes, increasing a players abilities.

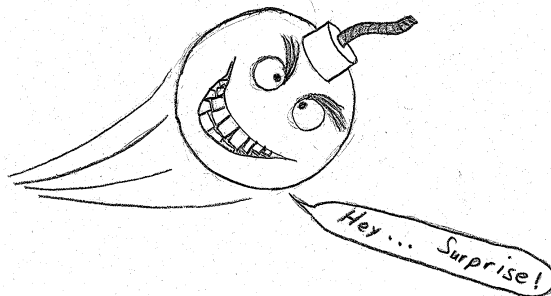
A level can consist of several dimensions, in which only one or the other player can take influence on interactive elements or enemies. This is used to create a cooperative puzzle mode in which the players try to reach together a specific switch or door.

The genre of the game is a mix between third person shooter and action adventure.

## 1.2 Story

The purely evil Dr. Morbo wants to take over the world. He developed a machine that can split the the toon world into multiple dimensions. With this machine he evokes ultimate chaos among the inhabitants. In this era of anarchy it'll be a effortless task to enslave each and every living creature.

But evil Dr. Morbo once again underestimated the courage and power of our toon heroes, who will come alive from the big pen every new episode. With the help of their friends at ACME laboratories they set off to confound Dr. Morbo evil plans and reunite the toon world so everyone can live happily ever after.



## 1.3 Gameplay

The aspect of the dimensions, allowing each player to do different things corresponding to the dimension he is part of, will take a major role in the gameplay. It will require the players to play cooperatively to progress in the game. This will be done with simple mechanics like sections where only one player can defeat the enemies. But it will also incorporate more complicated situations where puzzles have to be solved together. Starting out with easy button and key ones to more advanced physics puzzles. For example where players have to find boxes and stack them up to reach a higher platform or find weights to operate a seesaw. The design of such puzzles will be a very difficult and iterative process and an integral part of the map design.

## 1.4 Implementation

### 1.4.1 Design

Each entity of the game can have a visual representation and a physical representation. It also has an intelligence or logic which defines its behaviour. We therefore instantiate each entity as a controller which contains this behaviour. Each controller can create and control its visual and physical representation.

This design choice perfectly matches the XNA game class logic which separates the update and the draw routine. It is also intuitive, since each entity can decide about its own actions, even when to remove itself from the game.

Starting a game means, that a player selects a new map in the menu, the controller container is cleared and a new map controller is created. The map controller creates all entities of the map and adds them also to the controller container. Running the game means now just updating all controllers.

Since we have to fill a map with controller entities, we need a map editor. The input of the map editor are meshes, textures, effects and controllers, the output is a map description file describing the whole map and all controllers and their parameters. The map editor is not for modelling the world, but for composing meshes and their properties as well as controllers and their parameters.



### 1.4.2 Classification

The map and each entity of the game is represented as a controller. Nevertheless we classify the entities to have a better structure and an easier understanding of the game.

#### Static Entity

This class contains all entities of a map, which do not change position or form during the whole game. Their physical representation is created once and keeps the same and can therefore be optimized. Static entities have an infinite mass. They do collide with other entities but have no further logic. Because of their simplicity, they are directly handled by the map controller. Examples of this class are walls, the floor and stairs. This enables the map to be an arbitrary mesh.

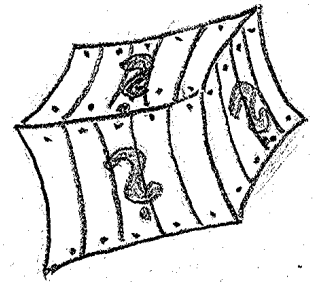


#### Semi-static Entity

All map entities which can move along a given path and react to events are called semi-static. The movement can be cyclic or caused by an event. Therefore they have to implement a certain logic and have to be represented by an own controller. They do collide with other physical representation but have an infinite mass and do not deviate from their given path. Examples for this class are elevators, doors, portals and switches.

#### Dynamic Entity

Entities which are only moved by physics and external forces are called dynamic. They have a certain mass and react on collisions. They can be destructible and also change their physical and visual representation. Examples are boxes and all kinds of bombs.



#### Character Entity



These are the intelligent entities of the world. They can move and interact with the environment and try to complete a task. There are player characters which are the avatars of the players and there are non player characters which are the enemies. Characters have health and can be damaged.

#### Abstract Entity

The last class describes entities without a physical representation. They have a task and query themselves for information to complete it. Examples for this class are spawn points and power-ups.



### 1.5 Visual Aspect

The characters are toons and have to look as if they just jumped out of a cartoon. An option would be to render them using the cell shading technique.

The maps can either play in the toon world or they can be more realistic, using bump maps and other realistic looking shaders. Although there will be many explosions, the game should never look violent but funny and humorous.

Capn Kaboom

## 2. Development Schedule

### 2.1 Team

#### 2.1.1 Peter Bucher

**Semester:** 8  
**Related Courses:** Software Engineering, Java Programming, Visual Computing, Introduction to Computer Graphics, Advanced Topics in Computer Graphics and Vision, Physikalisch-basierte Simulation in der Computer Graphik, Fräsen im Digitalen Publizieren

#### 2.1.2 Christian Schulz

**Semester:** 12  
**Related Courses:** Software Engineering, C# Programming, Programming Multimedia Systems, Computer Graphics, Computer Animation, Digital Compositing  
**Related Work:** C. Schulz, M. Schultz and H. Fricke 2008. A Real-Time Pedestrian Animation System. Proceedings of the 4th International Conference on Pedestrian and Evacuation Dynamics.

#### 2.1.3 Nicola Ranieri

**Semester:** 8  
**Related Courses:** Software Engineering, Konzepte Objekt-Orientierter Programmierung, Visual Computing, Introduction to Computer Graphics, Advanced Topics in Computer Graphics and Vision, Physikalisch-basierte Simulation in der Computer Graphik, Fräsen im Digitalen Publizieren  
**Related Work:** Title: “Tai Chi Chuang using wearable Sensors”, Supervisor: Dr. Dennis Majoe, Prof. Jürg Gutknecht

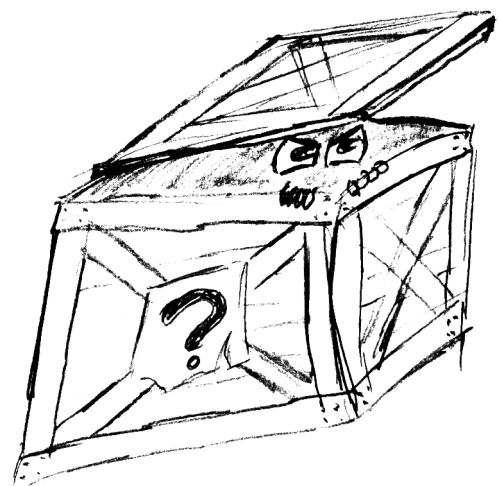
### 2.2 Tasks

#### 2.2.1 Functional Minimum

The functional minimum should be the prototype of the game. It should contain all elements required for a proof of concept:

- Standard map with some simple static entities
- Simple dynamic entities like boxes and one bomb type, without big visual effects
- One playable character, approximated by a simple box
- Multiple dimensions
- Working physics for the maps and the entities

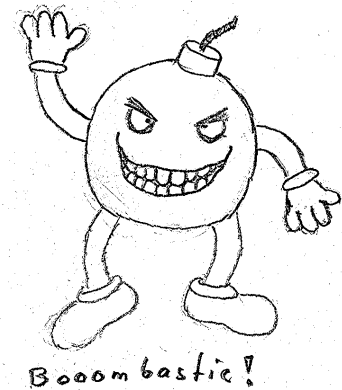
The map editor should be able to load and display meshes and to assign to each mesh an effect, a material and textures.



### 2.2.2 Low Target

The low target is a game which could be released, but without visual satisfaction and many entity types:

- A simple menu, with options and map selection
- One loadable map for multiple players
- An additional weapon
- Power-ups for life and ammunition
- Spawn point controllers for players, power-ups and boxes
- One character model loaded from dcc tool, several instances by color variation
- Animated character for forward / backward and sidestep motions
- Sound for all events

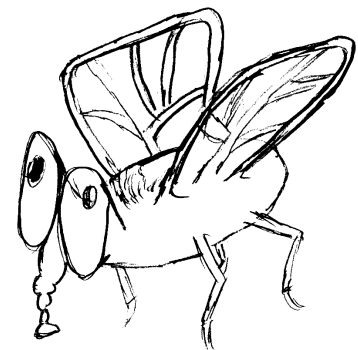


One can add the existing spawn point controllers to the map using the map editor.

### 2.2.3 Desirable Target

The game should now look good and could be released as it is. It contains:

- Improved puzzle maps
- Controllers for switches, elevators and doors
- An explosion effect
- Background music
- A mechanical non-moving enemy
- Several character models
- Realistic character locomotion generation
- More character actions: Jumping, Shooting a Bomb...
- Basic AI for NPCs
- Intelligent bomb which follows characters
- Improved character rendering: Shadow Mapping, Levels of Detail

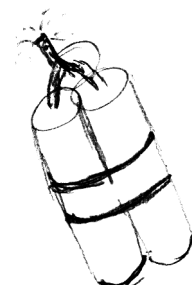


The map editor is extended so that the new controllers can be added.

### 2.2.4 High Target

High target is to have a visually ambitious game with a whole variety of different entities:

- A target tracking rocket and a doomsday bomb
- Keys (realized as power-ups) used to open certain doors
- More enemies which are more intelligent
- Characters have a better collision handling and corresponding animation
- More maps, especially puzzle maps
- Different shader techniques



### 2.2.5 Extras

As extras we plan those things which would be nice, but probably out of scope:

- AI for computer controlled players
- The game is playable over the network

## 2.3 Schedule

The term (x/y) stands for x hours expected, in y hours realized. Easter break is marked grey.

Date	Course Items	Peter	Christian	Nicola
24.02.		- FGP (3/-) - Sketches(2/-) - Mockup Scene (5/-)	- FGP (-/8)	- Design (-/4) - Skeleton (-/4) - Physics (-/16) - FGP (-/8)
03.03.	Formal Game Proposal	- Simple Map(10/-) - Meshload (5/-)		- Character Controller (20/-)
10.03.	Mutual Project Critiques	- Project Critiques (1/-) - Simple Mapeditor (20/-)	- Project Critiques (1/-)	- Project Critiques (1/-) - Dimensions (15/-) - Box Controller (4/-)
17.03.	Game Prototype, Functional Minimum	- Map-Modeling (5/-) - Mapeditor (15/-)	- Character modeling (12/-)	- Bomb Controller (20/-)
24.03.		- Textures (5/-) - Mapeditor (15/-)	- Animated character (12/-)	- Menu (20/-)
31.03.		- Textures (5/-) - DM-Map (15/-)	- Animation clip production (20/-)	- Multi Player (20/-)
07.04.	Interim Report, Low Target	- Sound Effects (10/-) - Props-Modelling(10/-)	- Realistic locomotion generation (20/-)	- Power-Ups (10/-) - Weapon (5/-) - Interim Report (5/-)
14.04.		- Enemies (20/-)	- Several character models (16/-)	- Switch (4/-) - Elevator (8/-) - Door (8/-)
21.04.		- Mapping (10/-) - Decorate (10/-)	- Character actions (20/-)	- Puzzle controllers (20/-)
29.04.		- Mapping (15/-) - Music (5/-)	- Improved character rendering (10/-)	- Effects (20/-)
05.05.	Alpha Release, Desirable Target	- AR-Report (5/-) - Balancing (10/-)	- Basic AI (10/-) - Intelligent bomb (6/-)	- Mech. Enemy (10/-) - AR-Report (5/-) - Adjustments (5/-)
12.05.	Playtesting	- Playtesting (10/-) - Model Cleanup (10/-)	- Better AI (10/-)	- Rocket (10/-) - Playtesting (10/-)
19.05.	Debriefing	- Shaders (10/-) - Finishing (10/-)	- Better Character Collision handling (12/-)	- Doomsday Bomb (10/-) - Shaders (10/-)
26.05.	Presentation, High Target	- Presentation (15/-) - Finishing (5/-)	- Presentation (15/-)	- Presentation (20/-)
29.05.09	Conclusion, Video	- Video (10/-) - Conclusion (5/-)	- Video (10/-) - Conclusion (5/-)	- Video (10/-) - Conclusion (5/-)



### 3. Assessment

Our game introduces a unique twist to the jump'n'run genre and will enforce a team oriented gameplay. As games are always more enjoyed when played together, the required cooperative parts will be fun on and off the screen requiring the players to actively communicate with each other. The jump'n'run parts are known to be fun and are proven to be successful, as long as the difficulty is right. Puzzles are very popular on a wide range of platforms and are generally enjoyed, but especially by a more mature audience. This allows the game to be played and enjoyed across age boundaries.

The different gameplay aspects attract a wider range of players. Both people liking puzzles and people which play shooters or action adventures should have fun with our game.

The introduced novel game twist combined with the funny atmosphere of a toon world should delight people not only for few minutes. Especially if we care to implement an easy character control.

