# Second Assignement
# Design specification

April 15, 2007

# 1 Gameplay description enhancement

## 1.1 Death Match

### 1.1.1 Goal

Kill all the other participants (ghosts).

### 1.1.2 Match Description

At the beginning of the match, all the ghosts are spawned at specific location inside the arena. They will be in free mode, with a certain quantity of energy. The time will start running, and the player will be free to fly and look for a rigid body they are interested in. The player are supposed to attack each other, and when the bound strength is consumed, due to damage, the ghost is kicked out. The match will end either because a certain amount of time is passed (say 5 minutes) or because all the ghost except one have been killed. A ghost dies when it's in the free state and its energy goes to zero.

## 1.2 Castle

### 1.2.1 Goal

The player either defend or attack a certain structure. The structure is composed by some rigid bodies, which are contained in a specific area. The whole structure has an health amount. Destroy the castle basically means to make a portion of the structure get out of the structure area (disassemble the structure) or simply to damage enough the structure health.

### 1.2.2 Match Description

The attacker(s) spawns at one corner of the arena, at the opposite of the defender(s) and the castle. All the ghost spawns free, with a certain amount of energy. If a ghost dies, it is automatically re-spawned in its initial location. The match ends either after some time or if the castle is destroyed. The attacker wins if the castle is destroyed, otherwise it looses.

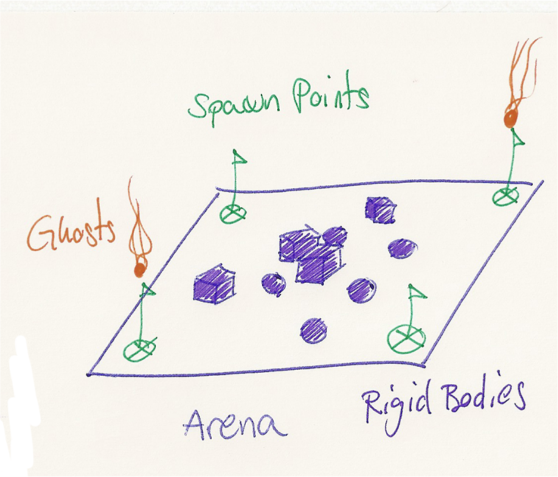

Figure 1: Deathmatch arena

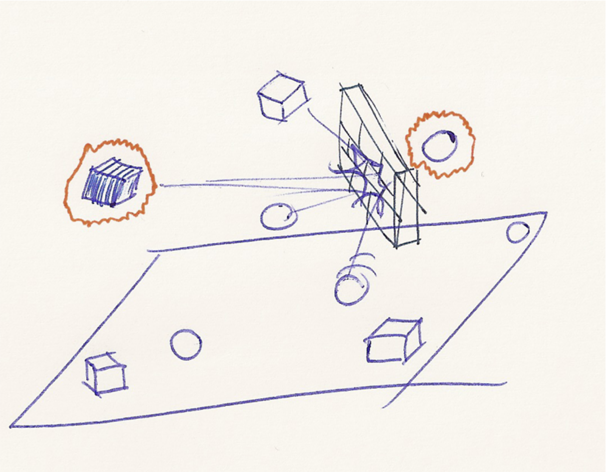

Figure 2: Attack and defense powers

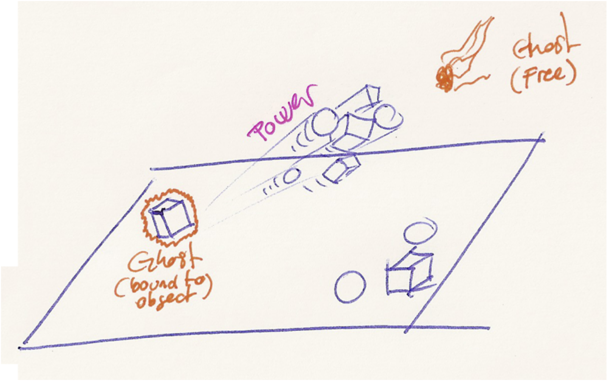

Figure 3: Fatality

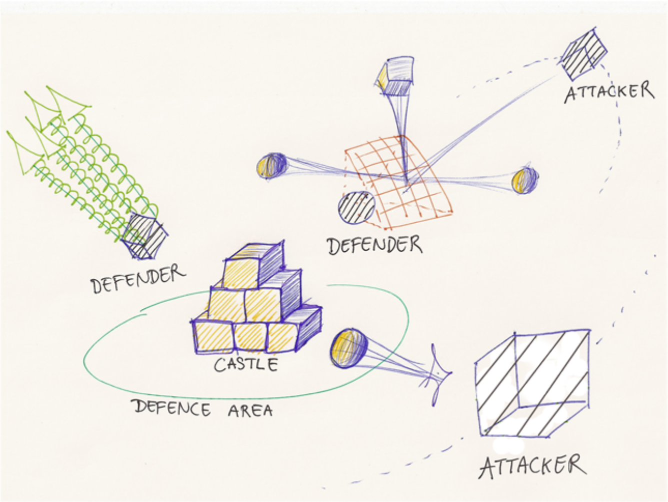

Figure 4: Castle Game Mode

## 2 Layers

We currently have worked on layer one (minimal proposal). The graphical engine is ready, the game-play code is almost there, and the physics translation from ChuckBox to C# is in it's way. The current rigid bodies do not have detailed texture, and the player interface need to be improved. We plan to be at a good point to the end of the week.

# 3 First feedback report

In this first phase, we basically had to cope with the XNA framework and the project bureaucracy: both caused more problems than what we thought.

XNA Framework: The XNA framework, even though lots of tutorials are provided, is not that easy-to-use one would expect. Especially if we want to go further than simple 2D spaceships games, we have to build many things from scratch, or -at least- is what it looks like. Math libraries are enough for what we have to do, but the graphics ones are hard to read, therefore to understand what is already there, and what we need to implement on our own, we feel a little lost, we hope to get through.

Project bureaucracy: What we really did not expect, was the large amount of documents we have to write, before and while we are developing the game. Being busy with so many reports, proposals and presentations makes it hard to start programming and testing our ideas. We feel like we have to do the work of a fifty people team, having four or five roles each one (e.g. write the proposal, develop the graphics engine, improve shaders, draw models and textures, or prepare the presentation, describe the goals, draw a basic GUI and think of the controller, etc.). Although we know that all this bureaucracy is needed in a big project, or even in a smaller one, to coordinate the work, we think that there should be people, for example, doing mostly that and less (or no) coding, maybe the chief of the project, the software engineers and so on; since instead we all want to really do something practical like coding or drawing, we are really slowed down by all the writing stuff.