# Titor's Equilibrium Project
# Formal Game Proposal - Sketch

Marino Alge
Gioacchino Noris
Alessandro Rigazzi

March 24, 2007

## Contents

# 1 Game Description

## 1.1 Introduction

In SS06 we worked, as group, to the winning project of the class Physically Based Simulation. The project was called Chuck Box (in honor to Chuck Norris) and was a real time 3D rigid body simulation, where the user could use some ”Force Power” (inspired to both ”Star Wars” and ”Cell Factor”) to move around simple objects (cube, spheres, ...), everything inside a simple box arena. Despite the simplicity of what we had as ”gameplay”, ChuckBox was real fun, and quite addictive. We are therefore interested on pushing that concept further, by creating a real game

Our game is called ”Titor’s Equilibrium”. The first word refers to John Titor, a person(s?) that in 2000/2001 was quite active in the internet time related communities, claiming to be a time traveler from the year 2036. He was supposedly sent back to 1975 to retrieve an IBM 5100 computer which he claimed was needed to ”debug” various legacy computer programs in 2036. Besides the fun of the thing, we got inspired to the concept of time handling and we planned to put it into the game (slow motions and, if possible, time reversing). The second word refers to the physical concept of equilibrium, which we, as Titor, are going to break :)

## 1.2 General Description

The game takes place inside a virtual reality runned by an ancient artifact (ideally a IBM 5100 ). The virtual reality entails:

- An arena, composed by:
  - infinite mass objects (boundary and obstacles)
  - finite mass object (rigid bodies)
  - special volumes (force fields, inaccessible areas, ... )
- Ghosts (Human and Computer guided players)

The players are supposed to be ghosts that can infest rigid bodies. Ghosts will be able to cast physically related powers to alter the dynamic of the simulation.
There are two main factors that determines the game possible situations:

- Time (Running/Stopped)
- Ghost Status (Free/Bounded)

When time will run, the simulation of the rigid bodies will take place. The arena structure will not be affected by the collisions. The force fields involved will be gravity, plus specific force field meant to make the arena more interesting. When time is stopped, the simulation will simply freeze, while the game will still be running. Some actions, such rotate the camera, will be always feasible. Other, such power casting, will require the time to run to be accessible.

### 1.2.1 Arenas

The arenas will be mainly composed by a single area, with a defined boundary, and some obstacles. The arena will also have some special volumes which will affect the physical simulation and other gameplay aspects.

TODO: Define special volumes in details.

### 1.2.2 Rigid Bodies

Mainly cubes and spheres (cylinders?), with defined sizes. They will interact during the physical simulation. They will be available as host to the ghosts.

### 1.2.3 Ghosts

A ghost has two main status: free and bounded. **Free** means that it is not attached to any rigid body. In this status the ghost is free to fly around the arena. Visually speaking it is no more than a will-o-the-wisp, a light particle(s). Moreover, being free, the ghost is not allowed to cast any physical power.

Once a ghost enters into a rigid body, it becomes **bounded**. In this status the ghost will gain characteristics and abilities. The characteristic will be:

- Bound Strength
- Energy

Two bars will show the amount of each quantity. When a ghost infest a rigid body, the bound strength bar is fully replenished. During the game the damage inflicted by the other players will lower the strength, and if the ”Bound Strength” goes to zero the ghost is kicked out the rigid body and goes back to the free state. The only way to recover bound strength is to quit the current object and look for an other, or in the case of blue objects, use a special ability.

The energy is used to cast powers. There is an amount of initial energy granted to the ghost when it first enters into a rigid body. Every time the ghost cast a power a certain amount of energy is drained. When the energy is spent, no more powers

can be casted. There are three ways to recover the energy. One is to wait enough time so that the energy is recovered (small regeneration rate). One is to quit the current object and look for an other (energy goes to initial quantity). The last one is to get damages (high regeneration rate). This method is the only one that can brings the energy to a higher level than the initial one (energy charge override). Therefore there is an implicit healt/energy trade-off.

*Chino: L'idea é che a livello tattico ogni tanto puo' convenire beccare danni, in modo da aumentare l'energia e poter castare qualcosa di potente.*

### 1.2.4 Powers

There will be two types of powers: Online vs. Offline. **Online** powers can be casted while the simulation is running. They are immediate and relatively simple, and will require less resources. **Offline** powers will be available only when the time is stopped. They will consume more energy, but also have more interesting effects.

In general online powers will make use of gravity fields, will deal damage to the enemies in various form (provoking collisions, through energy beams, ...), or will used to reflect damage, recover health, or do other gameplay stuff. They will be casted during the simulation actions, when the game is in first person view. See corresponding section for details in camera view.

Offline power will make extensive use of physical options. Here is the list provided by the physical engine.

- Joints
  - Springs
  - Attractors
  - Rigid bars
  - Ropes
- Fields
  - Gravity
    - Point
    - Directional
  - Magnetic (Torque)

### 1.2.5 Time Handling

Time will affect both the physical simulation and powers availability. To stop the time, the player will need to have a certain amount of energy. Once the time is stopped, offline powers can be casted. Those powers will require a planning (selection of the target, eventual parameter specification, ...) and will also require the player to pass some skill test (mainly based on controller possibilities, such fast pushing, or stick aim precision, ...). Once the offline power is set, time start running again.

## 1.3 Visual Aspects

### 1.3.1 Game look

The virtual reality components are supposed to look very high tech and very ”virtual”. While fix elements of the arena will be represented mostly by wire frames and energy grids, the rigid bodies will present smooth and clean shapes, few surface details, bright gray colors and, hopefully, nice visual appeal. Moreover, the rigid bodies will present some luminescent parts, whose color will affect the gameplay, as explained in the next section.

### 1.3.2 Colors

We are interested on give a certain meaning to the colors inside the game. See Figure 1.

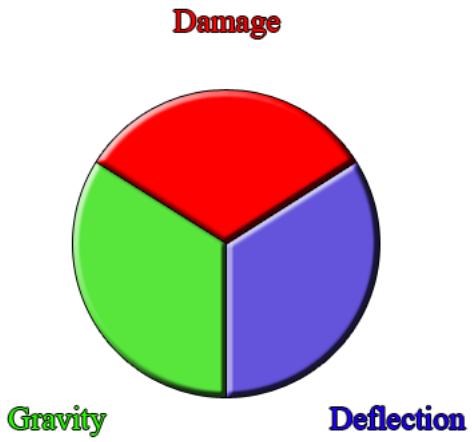


Figure 1: Colors define both powers and mission goals

*Chino: la scelta dei tre colori si giustifica sia per il fatto che concettualmente é facile avere in mente il paradigma: aggressivi + difensivi + alieni, sia perché il pad ha bottoni con questi tre colori. (Il quarto bottone, giallo-arancio possiamo usarlo per il tempo, definendo quindi tutto ci che ha a che fare col tempo di quel colore).*

The idea is that the availability of **powers** depends on the color of the rigid body that the ghost is infesting. when a ghost enters inside a rigid body, it gain new powers, when it leaves it will loose them. This will force the player to look for an other object to infest, depending on the power he or she need to cast.

Not only online powers, but also **mission goals** will be defined with respect to these three concept (damage, deflection and gravity). For instance a mission goal could be to do the maximum amount of damage in a certain time. An other example would be to push the enemy out of a ring, or simply to resist as much as possible. Knowing the goal, the players will try to get the right color.

Time will be represented by the color yellow-orange.

### 1.3.3 Special Effects

The game is supposed to express the physical simulation. Therefore possible eye candies will be welcome. It's hard to tell at the moment what will be implemented, but the idea would be to express object movements, as well as power effects. One aspect we are interested to mark is time bending. Some field of view manipulation, as well as motion blur and color enhancement will be considered while casting offline powers.

## 1.4 Interface

### 1.4.1 Camera Modes

- Free (Ego) View
- Scene Orbital
- Object Orbital

Free View is the one the player has while the animation is running. The eye is positioned at the player position, and the game controller will allow the player to look around freely.

Scene Orbital is the first view in the time stopped mode. It basically displays all the scene, allowing the player to circle around it, to better chose how to cast offline powers.

Object Orbital is then set when a particular offline power require to go closer to a specific object.

# 2 Development Schedule

## 2.1 Task List

| Task | Description | Who | Hrs | Actual |
|---|---|---|---|---|
| 1 | Brainstorming | All | 3 | 3 |
| 2 | Basic Physical Engine | Marino | 20 | ? |
| 3 | Modelling | Chino | 10 | ? |
| 4 | Graphical Engine | Chino | 10 | ? |
| 5 | Basic Game-play | Riga | 20 | ? |
| 6 | Testing | all | 5 | ? |
| 7 | Advanced Game-play | Riga | 10 | ? |
| 8 | Advanced Physics | Marino | 10 | ? |
| 9 | Advanced Graphics | Chino | 10 | ? |
| 10 | Testing | all | 5 | ? |

## 2.2 Task Description

### 2.2.1 Physical Engine

- Rigid Bodies definition (B)
- Collisions (B)
- LCP Solver (B)
- joint definition (A)

### 2.2.2 Modelling

- low poly meshes (B)
- uv mapping (B)
- high poly meshes (B)
- diffuse + normal + specular maps (B)
- improve quality of above steps (A)

#### 2.2.3 Graphical Engine

- Objects data structure (B)
- Camera definitions (B)
- Shader definition (A)

#### 2.2.4 Game Engine

- Logic definition (B)
- Structure definition (B)
- Binding with physics (B)
- Binding with graphics (B)
- Arena Editor (A)
- Tool Editor (A)

# 3 Assessment

We believe that this game has potential because of the fun of ChuckBox. People is not used to have the screen full of rigid bodies going everywhere, and jedi powers are indeed cool for everybody. Moreover we think that the combination of online and offline powers can really be fun. We would like to make the control easy and add an arcade score system, so that beginners can have fun and the geeks can try to reach new records.

Our team is well assembled. We have already worked together and we know each other pretty well. In terms os skill we are quite differentiate. Marino Alge has huge experience on c++ programming, and is focused more on the physics. He was the main coder for ChuckBox. Gioacchino Noris knows about modeling and texturing and is learning how to handle shaders. He's going to do the graphic related task, and visual design. Alessandro Rigazzi knows more about console video-games than anyone else, and has good math and programming skills. He will focus on gameplay implementation. Briefly said,we quite match the Tech-Art-Fun paradigm.

Besides that we all likes video-games, and we still don't believe that we have this opportunity.