GameDesign Document
Surreal Gravity

Group 2

November 28, 2014

Contents

1 Introduction

This document is a description of the game from a consumer point of view. It provides a detailed description of what we aim our game to become, without covering technical details. All elements mentioned are subject to change.

2 Target Audience

The target audience are PC Gamers who like fast paced FPS games like Quake, Unreal Tournament, Shootmania Storm, and Tribes Ascend. We also try to make it fun for casual people but we definitely want the game to be hard to master. But we hope the uniqueness of the game will make it accessible for people that are not that into F

3 Platform & Controls

The platform is PC (keyboard and mouse). As of now the controls involve rather standard fps control:

- WASD to move around in the horizontal plane
- space to jump up
- the mouse movement for aiming your weapon and looking around.
- left mouse button to fire a bullet that can kill other players
- right mouse button to fire your gravity changing gun

The game is designed for multiplayer gameplay using LAN.

4 Story, characters and setting of the game

We did not establish a background story yet. The background story will not be a core component of the game. Depending on how much time we have we can implement a tutorial level with extra background, and maybe build a sci-fi arena setting around our game. The characters in the game are the players and as of now just competitors.

5 Artificial Intelligence

Artificial intelligence in our game is mainly going to play a secondary role. It will not have a direct impact on the actual gameplay, instead it will serve to make the game's level itself more fun and interactive.

The main artificial intelligence component in the game will be little robots. During a game a certain amount of these robots will be spawned. These robots fly around the map and destroy the blocks that make up the level.

The first action a robot performs is select a block using a particular algorithm. This selection will not be completely random because you do not want to ruin the gameplay by destroying blocks. The solution to this is to make the robot only select blocks that are not part of a path in any of the 3 directions (x,y and z). If in any of the 3 directions the block is connected to two other blocks, then this block should not be selected for destruction. This will ensure that paths will remain in the level and therefore the ability to run around the level fluently will not be altered significantly by the robots.

The second action a robot performs is finding a path to the selected block from its current position. This path finding will be done with a Dijkstra algorithm.

The third action the robot should perform is actually following the path that has been found. This will involve animations for the movement.

Once the robot found its way to a block, there has to be a final animation that destroys the block. After the block has been successfully destroyed, the robot goes through the entire procedure again.

If players encounter a robot they are able to shoot it. This does not favor the player in any way, it's just a fun interaction between the player and the environment. The robot will respawn after some time if this does happen.

6 Level and environment design

The levels are made of procedurally generated blocks. The procedural generator defines a grid and fills up this grid with blocks. It chooses a number of "spawn" locations from where it will begin building islands of blocks. If there are enough blocks to be spawned, these islands will eventually meet to give a rise to a single coherent level. There are options in the algorithm to give it a preferred direction, which makes the islands more likely to grow in a particular direction. This function could be useful for example when there are 2 teams on either side of the level that must fight each other. It is then possible to make the level more likely to grow in the direction between these teams, giving rise to procedurally generated paths between the teams.

Currently there is no variation in the blocks but we plan on still implementing this in the future. We currently have a custom skysphere that gives a somewhat sci-fi feel to the game. We also plan on adding elements in the theme of the robots. For example spawn locations. The ideas of this subject are still in early development.

7 Gameplay and mechanics

The WASD keys handle the standard forward, backward, left and right movement. Space handles jumping. Currently one can jump when in collision with a building block. This means you can wall

jump! We like this feature because it gives a bigger feeling of freedom when navigating through the level.

The big feature of this game is the ability to change your own gravity. This is done with the right mouse button. This produces a rail that can hit a building block. The inward normal of the collision surface determines the direction of the gravity. This means that if you are standing in front of a block and shoot it, the direction of gravity will be towards this block, not away from it (note: this does not mean the block is a point source of gravity, just that the direction will be parallel to the inward normal of the face of the cube that was hit).

We think that this feature has great potential to make for some fresh new gameplay experiences. Because every player controls his own gravity, the wall of the first player can be the floor of the second player.

Finally there is the left mouse button. This fires a projectile in forward direction. The player has to use the projectile to hit opposing players. The bullet has no gravity component. The bullet can also be used to destroy ”employees”, but we currently do not plan on that being a goal.

8 Art

The art of the game involves multiple aspects. We mainly focus on blender models with skeletal animations. We use these primarily for the players and the ”employee” robots.
List of our blender models:

- Character
- Bullet
- Flag
- Robot
- Weapon

Because we have a futuristic sci-fi setting these models are made accordingly. The bullet is a sphere with 2 rings rotating around the core and will have a light effect attached. One ’hand’ of the robot will be a drill to drill out blocks and its other hand will be a plunger to pick up these blocks. The weapon is made with holes in it to make it more cool and futuristic looking that fits the bullet. We have made a relatively simple character and we will let some characteristics be customized. There is a flag which is used for the ”capture the flag” game mode. Textures are not done yet and will soon be added. We will also give the blocks that make up the level different textures so there is some variation.

9 Sound and Music

We use sounds for feedback in the game like any FPS would. So we have different sound effects for the player, when: walking, jumping, shooting a killshot, shooting a gravity rail and getting hit. We aim for sci-fi and futuristic sounds that are not too mechanic. The music in the game (while playing)

has to encourage the player and therefore adrenaline-boosting. Of course, it also has to fit in the theme of the game and therefore it has to give the player a feeling that can be summarized with the keywords: sci-fi, robots, aliens, futuristic. Our main inspiration sources are the song ”On the run” by Pink Floyd and Dubstep songs by Skrillex. We are planning to make another soundtrack for the menu as well, but this is of less importance than in game sounds as of now.

10 User Interface, Game Controls

We will provide the user with a variety of options in the menu section. The player will be able to modify its character and store it on a webserver. The host will have some additional options for the game settings (e.g. gravity strength). We have the possibility to make controls customizable, but this is no priority whatsoever. The credits will be available from the start menu.