Core project document

Group name and number

Group 18 - Brant Game Design

Theme and interpretation

We chose the theme ‘Water is cool’. Water will be the main theme in our parkour-based game, there will be water obstacles, water weapons and water enemies.

Game idea

In order to win this game you must outrun the boogeyman on a water-based parkour. You have to traverse waterslides, fountains and water cannons all the while tormenting your adversary with water based weapons.

You have to make quick decisions in order to outrun the monster, however, on your path there are also small adversaries. All these enemies can be beat by using weapons you collect during the game, such as a water pistol or an ice sword.

If you manage to get to the end of the parkour and kill the monster you win the game.

Student names, emails and role assignment

Thies Blaakman, thiesblaakman@hotmail.com, game designer/world builder
Benjamin Aartsen, b.aartsen@student.tudelft.nl, lead artist
Ardijan Dzeloska, a.dzeloska@student.tudelft.nl, gameplay testing
Niels Beaufort, n.m.beaufort@student.tudelft.nl, lead programmer
Rick van Arkel, p.f.vanarkel@student.tudelft.nl, producer

Game features

Computer graphics

- **3D animated models (procedurally) ** (*)**
  We'll add enemies in the game, which will have to be generated with the map as well. We need to be sure they are loaded properly.
- **3D models ***
  Simple models without moving parts only require textures and placement, depending on whether the map is auto generated or not.
- **Procedurally generated textures ****
  We want the textures to be generated only when nearing the location (so not a pre generated map). This could be done by generating textures in a specific distance range.
- **Animated textures ****
  We'll have to make sure the textures remain logical while animations are being played.
- **Particle Systems ***
  Water-gun, boss particles, explosions. These are all basic features.
- **Audio effects (explosions, soundtracks etc.) ***
  We can download some free files and/or record some of our own, and implement these when something happens. We have done this before.

Artificial Intelligence

- **Dumb enemy ***
  Simple character who shoots or moves towards a player, not very hard to program.
- **Smart enemy ****
  More sophisticated mechanics and smarter moves, so harder to program.
- **Smart boss (neural network) **(**)**
  The boss will learn based on the player's behaviour. When the player uses a certain weapon a lot during the first battle, the boss will develop defenses against that weapon for the second battle.
  If we choose to develop a neural network to really simulate a learning process, we think there will be a lot of challenges. None of us have ever created a neural network. How to select and use the inputs is an example of a challenge.
- **Consciousness in the level ****
  We can implement different behaviour in the level based on the players performance, but we do have to implement some kind of learning in this.

Game Analytics

- **Highscore ***
  A scoreboard to show the end scores of each play of the game. Not too hard to create, but a challenge keeping everything sorted and aesthetically pleasing.
- **Data analysis: keep different scores next to highscore ****
  Next to the highscore we also want to keep track of other scores by analysing the data generated during the runs. It is quite hard to keep track of everything and show this in a nice view.
- **Create gamer accounts ***
  We can make a script that stores the data and makes the account.

- **Create server for storing data ****
  We can make an online server that is connected to the game. We just have to find what kind of server can be used.

Programming

- **Game Mechanics**
  - **Power ups ***
    W*e change the values of a few attributes in the script (speed, health etc.), so quite easy.*
  - **Local multiplayer (two different characters in the same level) ****
    We have to add another player with its own key binds, shouldn't be too complicated.
  - **Pre generated boss room with own view and tactics ****
    We make a new kind of game mechanic here, but not too complicated.
  - **Waterslides, water blasters, fountains***
    We mostly make an object that has an influence on the gameplay once the player interacts with them.
  - **Dynamic difficulty ****
    We'll have to buff the frequency of enemies and tricky obstacles once the game advances.
- **Game loop**
  - **FPS independent ***
    A small feature to keep the gameplay the same for each device you play it on. Necessary to have as a feature, but not t0o complicated.
- **Physics**
  - **Use unity's physics to simulate normal movement and collisions, water physics etc. ****
    Essential features to make the game realistic and playable. Quite a lot of features have to be taken into account, so it will be time consuming.