Requirements and Analysis Document for Proximity

Date: 26/3 2015

Author: Group 11

This version overrides all previous versions

1. Introduction

1.1. Purpose of application

The purpose of this game is to entertain people who have time to pass and want to relax for a bit. We want to target people who enjoy strategy games, planning and virtual massacres.

1.2. General characteristics of application

Proximity is a deviation of a normal tower-defence strategy game detailing the story of a war between complex and simple geometric shapes. Compared to other tower-defence games this game focuses more on direct interaction, enabled by a spell system.

1.3. Scope of application

One playable level with finished artwork, enemies, working towers and an end game scenario

1.4. Objectives and success criteria of the project

- A gamefield (map)
- A placeable tower that costs resources and shoots enemies
- Approaching enemies that kills the player
- The player gets resources when towers kill enemies
- Different characters with different passive abilities
- Waves of approaching enemies as score system

1.5. Definitions, acronyms and abbreviations

- **TD**:Tower defence
- **Experience**: Points gained from killing enemies
- **Experience curve:** The experience required to gain a level is exponential
- **Unlockables**: Gaining a level unlocks corresponding towers and faction spells.
- **Level:** The level of the player.
- **Level-up:** Reaching the next level. Levels are gained by having enough experience.
- **Wave:** The enemies come in clusters, the current wave is the cluster the player is currently facing

- **Faction:** A type of spell-set that the player can chose from, different factions offer different passive abilities and active abilities (and possibly tower-optinons)

    - Fractal
    - Sphere
    - Cube
    - Stars
    - Crystal
    - Gems
- **Player:** The person who plays the game
- **Tower:** A block that the user can place on the screen. The player places towers. Most towers fire projectiles at enemies.
- **Projectile:** Some towers can shoots projectiles that hit the enemies, these projectiles cause damage to enemies it hits.
- **Enemy:** Creeps that follow a certain path in the game and tries to kill the player. These can be of different types and strengths.
    - Circle
    - Triangle
    - Square
    - Pentagon
    - Trapezoid
- **Devolve:** When an enemy dies, it “reincarnates” as a smaller enemy type, if it’s the smallest type of enemy the enemy gets destroyed
- **Resource:** “Money” that the player can use to purchase towers, different towers require different resources.
    - Line: A type of resource
    - Point: A type of resource
    - Polygon: A more valuable type of resource
- **Upgrade:** The player can upgrade towers to make them better
- **Spell:** The player gets one or more spells from the faction and this spell can be used to affect the game in the players favor
- **Passive Spell:** The player gets a passive spell from the faction, which is affecting the player without interaction
- **Enemy drop (loot):** The resources an enemy drops when it gets destroyed or devolves.

2. Requirements

In this section we specify all requirements

2.1. Functional requirements

2.2. Non-functional requirements

Possible NA (not applicable).

2.2.1. Usability

2.2.2. Reliability

2.2.3. Performance

2.2.4. Supportability

2.2.5. Implementation

2.2.6. Packaging and installation

2.2.7. Legal

2.3. Application models

2.3.1. Use case model

UML and a list of UC names (text for all in appendix)

2.3.2 Use cases priority

A list

2.3.3 Domain model

UML, possible some text.

2.3.4 User interface

Text to motivate a picture.

2.4 References

APPENDIX
GUI
Domain model
Use case texts