

System design document for Proximity

Version: 1.0

Date: 30/04 2015

Author: Linda Evaldsson, Simon Gislén, Johan Swanberg, Hanna Römer

This version overrides all previous versions.

1. Introduction

Introduction

1.1. Design goals

The design must be loosely coupled but the main focus is to have an advanced modular functionality. It must be easy to expand the program and add new components and functions. In addition, the design needs to be testable, that is the classes needs to be isolated and encapsulated.

1.2. 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-options)
 - 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. System design

2.1. Overview

In this section we explain the overall design choices.

2.2. Software decomposition

2.2.1. General

Package diagram. For each package an UML class diagram in appendix

2.2.2. Decomposition into subsystems

2.2.3. Layering

2.2.4. Dependency analysis

2.3. Concurrency issues

2.4. Persistent data management

2.5. Access control and security

2.6. Boundary conditions

3. References

APPENDIX