5/6/2018

Connor Rowland – P16204361

LECTURER: Dr Simon Coupland

Blaster Design Report

Object Oriented Design and Development with C++

Contents

[Introduction: 1](#_Toc513394073)

[Controls: 1](#_Toc513394074)

[Design decisions: 1](#_Toc513394075)

[Test table: 1](#_Toc513394076)

# Introduction:

I created a 2 player, 2D worms-type, game. You fight another person, taking turns. You can either move, or attack. Your soldier has a health, lives, and a score.

# Controls:

Left mouse-click to move.

Right mouse-click to throw a bomb.

# Design decisions:

I started by planning what the game would need. Once I had thought about what the game will have I started to extract data from each object, so I could create a hierarchical design. This would allow me to create polymorphic data structures.

I decided to extract the most basic components of an object, into a class called GameObject. This has a (game) position, a size, and a pure virtual method that child classes must implement, it inherits the draw method from sf::drawable. My plan was to have every game object, in the “world”, inherit from game in game.

Another (abstract/interface) class called DynamicObject inherits from GameObject. Every object that can move, will inherit from this class, as it has all the attributes necessary to move an object. This forces any child classes to create an update method, and a draw method, as every dynamic object should be able to move, and be drawn.

I didn’t see a reason to add an abstract base class, for static objects, as they’d mostly have the properties of the GameObject.

I extracted the ability for object to be pixel perfect by having

# Test table:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test number: | Context (what am I doing) | Input: | Expected: | Actual: | Passed: | Comment: |
| 001 |  |  |  |  |  |  |
| 002 |  |  |  |  |  |  |
| 003 |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |