MegaStorm Games

Ross McDade & Alasdair Hendry

title of game

Games Design Document

Table of Contents

[Game Overview 2](#_Toc495407590)

[Game Proposal 2](#_Toc495407591)

[Summary 2](#_Toc495407592)

[Aesthetics 2](#_Toc495407593)

[Influences 2](#_Toc495407594)

[Mood board 2](#_Toc495407595)

[Approaches 2](#_Toc495407596)

[Chosen Approach 2](#_Toc495407597)

[Rejected Approaches 2](#_Toc495407598)

[List of Resources 2](#_Toc495407599)

[Project Plan 2](#_Toc495407600)

[Milestones & Deliverables 2](#_Toc495407601)

[Technical Resources 2](#_Toc495407602)

[Design 3](#_Toc495407603)

[Artwork 3](#_Toc495407604)

[Sketches/Concept Art 3](#_Toc495407605)

[Storyboards 3](#_Toc495407606)

[Flowcharts 3](#_Toc495407607)

[Flowboards 3](#_Toc495407608)

[Implementation Documents 3](#_Toc495407609)

[Pseudocode 3](#_Toc495407610)

[Damage Pseudocode 3](#_Toc495407611)

[Points Pseudocode 4](#_Toc495407612)

[Player Pseudocode 4](#_Toc495407613)

[Enemy Pseudocode 5](#_Toc495407614)

[Friendly Cells Pseudocode 5](#_Toc495407615)

[Power-Ups/Weapons Pickups Pseudocode 5](#_Toc495407616)

[Use Case Diagrams 6](#_Toc495407617)

[Use Case Descriptions 6](#_Toc495407618)

[Test Plans 6](#_Toc495407619)

[Technical Testing 6](#_Toc495407620)

[White Box Testing 6](#_Toc495407621)

[User Testing 6](#_Toc495407622)

[Black Box Testing 6](#_Toc495407623)

# Game Overview

## Game Proposal

Platform for game, Genre of game and the overall theme/setting

## Summary

Short summary of game. Quickly mention story and gameplay.

## Aesthetics

Look of the game, realistic/cartoon whatever.

# Influences

Influences of game, such as art styles, gameplay and games that are similar to what we aim to create. Subheadings may include films/tv and games.

# Mood board

Images of similar art styles and games that inspired the game and will help give anyone reading the document a clear idea of what we want the game to look like.

# Approaches

## Chosen Approach

We decided to go with the Agile Software Development Life Cycle. It was agreed that due to the short amount of time we have to create the game this approach will work best for our team as it allows for lots of very quick iteration and is flexible when it comes to making changes to the project, this will work well as we have ideas for the project that can be implemented later if the time is available for us. It also fits very well with our test plans as we decided that we would test as we implement which Agile encourages. However, what we did choose to amend were the scrum meetings, instead we will merge them with our weekly minutes meetings and use that to address any and all problems while we are developing the project. Instead of releasing the project to any form of client or outside tester we will be handling the majority of it internally and will be critical of each other’s work in order to create the best project in the allotted time.

## Rejected Approaches

Talk about any approaches we discussed but ultimately were rejected and why they were rejected.

# List of Resources

Table of what will be needed for the game, sound effects, models, textures, etc.

# Project Plan

## Milestones & Deliverables

Talk about Gantt Chart and stuff. Link to Appendix document. Quickly list milestones and deliverables.

## Technical Resources

List of the software and hardware that will be required in order to produce the game. Talk about modelling software, game engine, hardware requirements.

# Design

## Artwork

### Sketches/Concept Art

Talk about what we wanted the game to look like. Link to actual sketches and drawings in appendix document

### Storyboards

Talk about storyboards of animations and such. Link to them in appendix document.

## Flowcharts

Insert flowcharts, talk about how each section will run through

## Flowboards

Eventually add in flowboards.

## UI Design Document

Create UI Design Document.

# Implementation Documents

## Pseudocode

We decided to create some Pseudocode for the main functions of the game in order to help better understand what skills would be required to achieve our vision. It will also help outside parties to understand the source code along with the comments that are contained within the code itself. Please see ‘Pseudocode’ section of the appendices document for pseudocode that was created when planning for implementing additional features but was removed.

### Damage Pseudocode

#### Player Gets Hit

If player is hit by enemy attack  
 corresponding value of damage is removed from health bar   
else if value of damage is greater than value of health bar  
 player dies

#### Player Dies

If player dies   
 game is over

If game is over  
 game over screen pops up  
 gives player option to use regeneration ability if purchased  
 allows player to restart

#### Enemy Gets Hit

If player hits enemy with attack  
 enemy takes damage

If damage value is greater than enemy health  
 enemy dies  
 player gets points

### Points Pseudocode

#### Level Completion Points

If player completes level  
 Certain amount of points are awarded

#### Bonus Points

If player gets combos  
 points are awarded based on multipliers

Points are awarded based on the average position of infection meter

### Player Pseudocode

#### Player Movement

If w key is pressed   
 player character moves up to top of screen/back of map

If a key is pressed  
 player character moves to left of screen

If s key is pressed  
 player character moves to bottom of screen/front of map

If d key is pressed  
 player character moves to right of screen

#### Player Combat

If right mouse button is clicked  
 minigun fires bullets

If left mouse button is clicked  
 special weapon is fired/pickup ability is used

If r key is pressed   
 minigun will reload

#### Player Direction

If mouse is moved left   
 player character will rotate to the left

If mouse is moved to the right  
 player character will rotate to the right

### Enemy Pseudocode

#### Enemy Type 1(Melee) Behaviour

If player is not in the vicinity of enemy   
 charge towards player’s position  
else if player is in vicinity  
 use melee attack to cause damage to player

#### Enemy Type 2 (Martyr) Behaviour

If player is not in the vicinity of enemy   
 slowly make way towards player’s position  
else if player is in vicinity  
 explode to cause damage to player

If friendly cells have spawned  
 slowly make way to friendly cells’ positions  
else if friendly cells are in enemy vicinity  
 explode to cause damage

#### Enemy Type 3 (Shooter) Behaviour

If player is not in vicinity of enemy  
 begin firing bullets at player’s position  
else if player is in vicinity  
 retreat away from player and begin firing at player’s position once far enough away

#### Enemy Type 4 (Brute) Behaviour

If player is not in vicinity of enemy  
 slowly make way to player’s position  
else if player is in vicinity  
 use ground pound attack to cause damage

### Friendly Cells Pseudocode

#### Friendly Cells Behaviour

If no enemies in vicinity of friendly cell  
 remain idle  
else if enemies in vicinity of player character  
 attack closest enemies to player  
else if enemies in vicinity of friendly cell  
 attack closest enemies

### Power-Ups/Weapons Pickups Pseudocode

#### Picking Up power-Ups/Weapons

If f key is pressed near power-up/weapon  
 player picks up and stores power-up/weapon

If player has power-up/weapon stored and f is pressed over new power-up  
 power-up/weapon will not be switched

## Use Case Diagrams

Talk about UCDs and link to Appendix Document for diagrams themselves.

## Use Case Descriptions

Talk about Use Case Descriptions and link to Appendix Document

# Test Plans

## Technical Testing

### White Box Testing

Methods for testing with access/knowledge of code.

#### Test As You Go

Describe how as components are added they shall be added and outcomes will be recorded in Dev Testing Table, with expected outcomes, actual outcomes, fixes, and how long it took to debug problems. Provide example in appendix document.

#### Safe/Extreme Testing

Test certain components within their limits and try to purposely break them. Provide example in appendix document.

## User Testing

### Black Box Testing

Create a test document and allow users to play game while trying to find any bugs but give no access/knowledge of code to them. Provide example in appendix document.