Technical Design Document (TDD)

**Shooter on rails**

Team Name

Version: 2

Created: 9/3/2015

Last Updated: 9/13/2015

**Table of Contents**

Executive Summary 3

Project 3

Technical 3

Programs 3

Market Release 3

Time to Completion 3

Estimated Cost of Completion 4

Hardware and Software 5

2D Software 5

Sound Software 5

Programming Software 5

Development Plan 6

Milestones 6

Project Goals 6

Features 6

File Formats 8

2D 8

3D 8

Audio 8

Scripts 8

Scenes 9

Other 9

Asset List 10

Audio 10

Executive Summary

**Project**

The purpose of this project is to produce tools suitable for use in a rail shooter. The tools should be able to receive movement, time, and special effect waypoints over a given timeframe, read in with each individual waypoint and translate it to play. The code should also be able to read in user generated content from a text file and parse it into waypoints for the tools to implement.

**Technical**

Programs

The following programs will be required for the creation of Around the World:

|  |  |  |
| --- | --- | --- |
| Program | Purpose | Cost |
| Unity3d | Game Development Environment | $1500 |
| Audacity | Sound Creation | Free |
| Microsoft Visual Studio | Debugging Environment | Free |
| G.I.M.P | 2D Image Manipulation | Free |
| Blender | 3D Graphics and Animation | Free |

Market Release

Shooter on rails will be an in house tool with no release.

Time to Completion

The total estimated completion time for Shooter on rails is two weeks. That includes time for coding and bug testing.

9-13-2015--There have been hiccups and time budgeting problems for each individual so a more stable form might take more time to complete.

estimated Cost of Completion

|  |  |
| --- | --- |
| Unity3d | $1500 |
| 60 Hours Salary | $1,620.00 |
| **Total** | **$2120.00** |

Hardware and Software

**Programming Software**

|  |  |  |
| --- | --- | --- |
| Software Name | Description | Cost |
| Microsoft Visual Studio | Used to debug and create code | Free |
| MonoDevelop | Used to create prototype code | Free with Unity3D |
| Unity3D | Development Environ-ment. Used to release to multiple platforms. | $1500 |

Development Plan

**Milestones**

|  |  |
| --- | --- |
| Date | Milestone |
| 9/4/2015 | TDD Complete |
| 9/7/2015 | Prototyping Complete |
| 9/8/2015 | Engine Completed |
| 9/11/2015 | Tool Features Created and Editor Work Started |
| 9/13/2015 | Software Created and Entirely Bug Free |
| 9/14/2015 | Complete and Submitted to Professor Fisher |

**Project Goals**

Features

Movement, Camera, and Special effect waypoints will have separate arrays and time frames so they compile, run and can be modified easier. This makes the inclusion of UGC files a bit more complicated, however.

There will be a minimum of four different movement types among the waypoints, including a straight line, a Bezier curve, or others.

The camera will include at least two different types of movement, including a free movement setting determined by the player and made possible by code acquired from author asteins at the Unity3D wiki, and a fixed location stare.

The camera will also be able to have several effects added to it. The camera should be able to shake, fade, and add a splatter effect to it.

There will also be extra code to read in User Generated Content that is written into a text file. It will be translated and added into the individual statements.

File Formats

**Scripts**

|  |  |  |
| --- | --- | --- |
| Naming Convention | Description | Format |
| FilenameEditor | Editor script for an object script | .cs |
| Filename | A script for an object | .cs |

**Scenes**

|  |  |  |
| --- | --- | --- |
| Naming Convention | Description | Format |
| SceneName | A scene in Unity3D | .scene |

**Other**

|  |  |  |
| --- | --- | --- |
| Naming Convention | Description | Format |
| TDD\_Finished | The completed Technical Design Document | .pdf |

Asset List

Scenes

Test scenes.

Unity Primitives

General capsule with a character controller

extra cubes to see movement, possibly use for transform locking.

Scripts

-For Camera

FacingEngine.cs

FacingWaypoint.cs

FacingWaypointForcedLocation.cs

FacingWaypointFreeMovement.cs

CameraMovement.cs - by asteins licensed under GPL 3.0 at <http://wiki.unity3d.com/index.php/SmoothMouseLook>

-For Movement

NodeMovementPath – holds path data in node objects

NodeMovementPathFollower – can read paths from path objects and move accordingly

NodeMovementPathEditor – for better display of node data

-For Special Effects

CameraFadeScript.cs

CameraShakeScript.cs

Accountability

Victor

* Team Lead
* Git repo management
* Camera facing

Gipson

* IO
* Movement node
* logging

Darrick

* Camera effects
* gizmos