AGP Group Project – *Digital Desperado*

- MAIN DOCUMENT

- V3

- 17 MARCH 2015

CONTENTS

*Pg – [Section Num] [Section Title]*

*2* – 1 Project Abstract

*2* – 2 Goals

*3* – 3 Roles and General Duties

*3* – 4 Time-frame & Deadlines

*4* – 5 Communication

*4* – 6 Source Control Project Folder Guidelines

*4* – 7 Task Handling

*4* – 8 Work Already Made

*5 –* 9 Testing

*5 –* 10 Side Application

*5 –* 11 Coding Guidelines

1 Project Abstract

This project is an assignment for our course, with the aim of making a game to BUGS's criteria. The final deadline is 1st May 2015, and as of writing the game will be a re-make of the one Peter/Alex/Tim worked on for the OpenGL FPS game last year. The game in super-short is a First-Person Speedrun Target Gallery shooter.

2 Goals

- Make a polished First-Person Shooter Unity Windows Standalone game to the criteria below.

- At the deadline deliver a Zip file containing:

* The development folder containing our final project + should be able to compile without errors
* A release folder with just what is needed to run the game
* A promotion folder with anything used to promote the game (e.g. Trailer)
* A readme file with gameplay guide, any bugs + workarounds + Accreditation

- Have it tick EVERY point in the BUGS submission criteria, which are ([www.bugsnet.org](http://www.bugsnet.org/)):

* Game should be bug-free
* Should perform to all platform specifications + expectations
* Should be already hosted or downloadable from a third party source
* Should be presented on a fully filled out new games form as supplied including link to the relevant server/provider
* Should include these components/features:
  1. Loading Sequence
  2. Main Menu, Play Button(s)
  3. Options Menu, appropriate to format
  4. Quit Option for PC games
  5. Fully working, and easy to understand UI
  6. Credits
  7. Coherent Scoring/Progress system of some nature
  8. Full Menu System as required per platform
  9. Level intro/outro (Score screen counts)
  10. Any level structures fully finished and completed
  11. Smooth Multiplayer set-up and performance if used
  12. Full game loop that does not collapse at any point
  13. Proper AI (if used)
  14. Full sound
  15. Proper graphics
  16. Decent, smooth, Controls that work without error
  17. Checkpoint save system for longer/story-based games
  18. No dead ends in structure or level designs
  19. No need for console commands at any point
  20. Links to download appropriate hosting site that work reliably

3 Roles and General Duties

This section covers who SHOULD be doing what, which we should aim to keep to, but can allow cross-over if help is necessary, this section also lists who is relatively experienced in what field, if help is needed for certain aspects in your scripts. It is assumed everyone can program in C#.

**Peter**

- Project Manager, 2D Artwork, Audio finding & Production, Menus

**Alex**

- XML, Side Application

**Tim**

- Audio finding & Production, Player Controls

**Russ**

- Targets, Player Shooting

4 Time-frame & Deadlines

Overall we have up to the 1st May to complete this task, but this section details deadlines group-made and agreed upon leading up to that deadline:

1. Alpha
   * 13 April
   * Basic gameplay in a pre-made Unity Scene, Menus, Basic Audio, we can load the minimum for a level from XML, and can save data (Scores or a random integer)
2. Beta
   * xx xxx
   * All gameplay features in, More detailed XML levels, Options menu, Most if not all Art/Audio assets complete
3. 1d0
   * xx xxx
   * All features in, levels in, at this point just bug-squashing and final touches to the game
4. Final Deadline
   * 01 May
   * See Goals for what is required

5 Communication

For this project, we'll be using Skype/Facebook to communicate on a regular basis

6 Source Control Project Folder Guidelines

In order to reduce the likelihood of colliding files, the SVN project folder will be structured as follows:

- **Documentation**, Where documents like this will be kept, including any accreditation notes

- **Design Pages**, where the Inkscape design pages + versions will be kept

- **UnityPackages**, where exported UnityPackages should be exported

- **Builds**, where any merge builds will be kept + older versions

- **Merge Project**, where the primary build will be kept away from our sub-folders

- **Personal Folders**, where each of us will be given a folder to do our work in, including personal builds, and Unity projects, so as to avoid overriding one-anothers work by accident.

- **Promotion**, where any trailers, art assets are kept, pretty much anything to be put onto the web goes in here

7 Task Handling

In regards to handling tasks, we will use a Google Spreadsheet, so as to have a central file, which all can edit, and updates as it is being edited, to avoid having alternate versions of the list, as well as avoid forgetting to commit everyone’s list in SVN, the link to the file below:

<https://docs.google.com/spreadsheets/d/1zs0sTmEjFPRboYCtdpapDb-peBxPZjQQf1y_sPJuNYk/edit?usp=sharing>

8 Work already Made

This section details what work has been completed prior to the project being started, as well as detailing from whom it came from, if we need to ask how it works.

Peter (*Attempted DigDes Re-make*)

- UI, Menus

- XML format ideas

- Fonts

- Basic Textures

Peter/Russ (*Deadly Penguin Horde*)

- Transitioning Menu

- Music Player Script

- Effects Script

Alex/Tim (*Phokos Prison Riot*)

- Options Menu

- Local Save Data

9 Testing

Part of the submission's rules is to be bug-free, hence we will be using a variety of testing methods, focusing on play-testing:

As this game is being developed with speedrunning in mind, it could be prudent to send what we have made to speed-runners online who if we are lucky will respond with usable feedback.

Regarding development, a side application will be developed that will parse XML files, with the intention of parsing for the level format to then generate the level as it would in the Game itself – letting the user then move around in free-space to check for issues, as well as discover any issues without having to load the game EXE and find it to test.

Alongside this, we should attempt to build a set of friends/acquaintances who will agree to test for us, in order to get more views on components such as the tutorial, or general controls.

10 Side Application

As stated above, this application will be part of our testing strategy, the goal being to have a quick and efficient tool to parse, and test any level XML files created, allowing the team to test before implementing into the game, as well as have a tool to export with the game itself to allow players to make their own levels more easily as well.

As of now it only has these points, but Alex will be defining the application more explicitly over the coming Easter break:

* It will be a Unity Application (To test under the same code as the game)
* It will be Windows standalone as the game itself
* It will have a basic GUI to allow re-loading an XML file if it has been edited

11 Coding Guidelines

To keep things at a general consistency, scripts must have this structure at the minimum:

**- Programmer** – Note down that you are the one who has written the script, if someone else needs help on how it works

**- Script Summary** – Provide a small explanation of how the script works and what it does

- **Script** – It is imperative that in the script you have written, there is a script that does stuff, and not a novella

- **Comments** – Comment your work so that if you are unreachable for any reason, we can progress until you are reachable (If you die then we'll contact you through your chosen deity/Random Number God for aid)