Year 2 Project Technical Document

Introduction

“Scarlet” follows the story of an orphaned girl, taken in and raised by pirates. After her escape she travels the galaxy looking for treasure and spare parts to make her ideal ship.

Platforms

Our main platform is directed towards PC, with controller support. We have not ruled out console ports at this stage. 12/8/15

Platform Choice and Technical Specifications

The main platform we are targeting is PC, minimal specs include.

OS: Windows 7, 8

Processor: Intel(R) Core(TM) i7-3770 CPU @ 3.40GHz 3.90 GHz

Installed memory (RAM): 16.0 GB

Graphics: DX9.0c Compatible 256MB VRAM, NVIDIA Geforce 7600 Series / ATI x1600 Series

System type: 64-bit Operating System

DirectX®: 9.0c

Hard Drive: 3 GB HD space

Sound: DirectX 9.0c Compatible

Code

Source Control and build process

The source control implemented and monitored by Jack Carolan (Lead Programmer) is “Git Hub” as well as google drive for relevant files.

Using source tree, the development group will submit “Main” files relevant to the project (character geo, level floor plan etc.). Other files “non-essential” to the development of the game are to be submitted onto google drive. Etiquette regarding structure or submission in separate folders for each part of the team in development (programmers, designers, and artist) are outlined in a text file called “Etiquette” within each section. The main build of the game will run out of a forked repository integrated into Git-Hub, and updating through Epic Games main repository. This is to be maintained and updated by the Lead programmer.

The rules for submitting and editing are as follows.
All use of common files, as defined by each part of the development team, is to be announced when worked on, and announced when work is completed. This is to minimize duel editing.

At the start of the working day, every team member working that day must pull the repository from source tree. At the end of the working day, each member must push any changes to Git-Hub. This is to minimize the amount of pull and push request to make keeping track of them easier.

If a branch is made, it must be communicated and approved by either one or both Lead programmer/Lead Designer.

If a merge is made this must also be communicated and approved.

Google drive is free game go nuts.

How often do you build? Where do builds go? How do you make an automated build? What happens when a build fails? Do you have a dedicated build computer?

Builds are to be handled by only the Lead Programmer/Designer, builds will be staggered and made whenever necessary there is no real guideline for when. However, builds are not to be kept on Git-Hub and only kept on google drive for a minimal time, if builds are to be documented it will be the Lead Programmers responsibility to collect them in a folder on a designated desktop.

Coding Standards

X represents incorrect usage.
Member variables:

Variables existing within an object are to be labeled with “m_” and camel casing is to be used. E.g. m_Health, m_ChargingBladeBar.

All member variables are to be grouped appropriately in public or private as necessary, and must be GROUPED according to type to reduce cache misses.

E.g.
Float x
Char x
Float x
Int x

Int
Float
Float
Char

Temp variables:
Temp variables are to only be used in functions unless debugging/implementing mechanics, they are to be labled “temp” + “relevant name to what the temp represents” lower case temp with caps for resulting words with a number if there is more than 1 temp value (the first being 1, second being 2 and so on) of the same kind.

E.g.
Temp x
tempNum
tempCharacter
tempCharacterGun.
tempCharacterGun1.
tempCharacterGun2.

Functions:
“f_” is to be at the start of every function, with camel casing resulting afterwards. Functions belong after variables have been stated.

There are no language features or limitations in terms of coding at this time. 12/8/15

Software

Epic Games launcher (and required files).
Source Tree.
Google drive.

The lead programmer will update other necessary files and guide developers through installation in person if necessary.

Assets

Asset Formats

As of this time, 12/8/15 no file restrictions are imposed on the artist with and exclusion that all meshes must be ".fbx" files. This is to allow the artist as much freedom as they can, they will also amongst themselves settle on these kind of restrictions.

Folder and Project Structure and file naming conventions

As stated above all "Main files" are to be stored in Git-Hub within their allocated folders, "non essential" files are to be stored on google drive where its free game how you store your files.

Final assets are stored on Git-Hub in allocated folders labeled "Final".

All in work files follow the first listed rule, and follow the etiquette set by each group within development.

Asset List

Sewer:

Pipe

Dirty water

Ruble

Cobwebs

Ladders

Sci fi sewer light

Sci fi pipe

Control panel

etc

City:

Street lights

Bins

Stalls

Signs

Ladders(Fire Escape)

Clock Tower:

Bench

Rubble

Candle stand

Bells

Rope Piles

Torch

Characters:

Illuna Rose: Main character

3x General pirates

Sewer Hulk: 1st boss

Sewer Remanent: 2nd boss

Timeline

There is no set timeline for code submission or any set out plan of goals. Programmers are adopting a more agile method.

There are however task that need to be completed and the time allocated to each task.

Build conversions (converting from 4.7 to 4.8) 1 – 3 hours.

Code Cleanup 1 – 2 hours.

Applying animations 1 - 2 hours.

Development of Mechanics 1 – 4 hours.

AI Creation and Implication 7 hours.

De bugging/cleanup 1 – 2 hours.

Refining 1 – 2 hours.

Programmers are to communicate when working on the same file.