

**CGP2003-0708: Computer Games Production Group Project****Assessment 3****Group 5****Supervisor – Duncan Rowland*****Work Percentage Allocation Chart for Assessment 3***

<b>043315026</b>	<b>Toby Arnold</b>	<b>19%</b>
<b>06039966</b>	<b>Jason Gunn</b>	<b>19%</b>
<b>052429198</b>	<b>Stuart Horwood</b>	<b>15%</b>
<b>06045447</b>	<b>Kenny Gorton</b>	<b>16%</b>
<b>06032396</b>	<b>Mitchell Williams</b>	<b>16%</b>
<b>05004268</b>	<b>William Davies</b>	<b>15%</b>

**Introduction:**

The Purpose of this document is to demonstrate how, as a group, we intend to implement the requirements stated in our first deliverable, and provide an in-depth overview of the tasks involved. We will also discuss any changes we made to the aforementioned requirements and the consequences of those changes. We will include a Flow Chart depicting the role/outcomes of the user in our University Induction Mod Project. Any third party components vital to the composition of the final product will be pointed out and briefly discussed as well as a justification for their inclusion.

**DESIGN:****1. Changes:**

In our initial design, we had planned to include particular areas of the University for the "Player" to preview but not actually interact with. Areas such as the Architecture Building's foyer and the area leading to the Sports Hall. However because of the timeframe we decided as a group that this simply wouldn't be possible, at least not with consequence to the quality of the modification.

We also stated that the modification would utilise the same engine parameters as the commercial game Team Fortress 2. However, it is now evident that this will not be a possibility due to Valve not including the engine features in the source sdk base soon. That said, they may include it in the duration between this document and the proposed deadline. However we can't deal in "ifs" and "maybes", so we will have to put the work effort into duplicating the feel of the game at least in terms of theme and style.

**2. Design Overview:*****2.1 Game Mechanics***

This section of the report will talk about the mechanics of the game, how the player interacts with the engine, examples of quests etc.

***2.2 Quest Givers/NPCs***

In order for the player to progress through the deliverable, they will need to be set goals to achieve so that they understand what to do, and when to do it.

The main method for setting these goals (Or Quests) will be via various NPC's which are used to represent their real world counter parts.

We still need to attain permission to use the characters, and voices of key members of the university staff, however if we can get this then the overall experience for the player will be much more realistic, allowing them to recognise staff should they come to Lincoln University.

The key NPCs will be:

- Mark Doughty
- Kevin Jacques
- Reception Staff (Both Main, And Computing)

A player will interact with these NPCs by pressing the "Use" key which will be as default "E", the NPC will then respond to the user, either by giving them a quest, or useful information, or simply telling them to return later as it is not time for the NPC to be conversed with.

***2.3 Rewards***

In order to promote the players progress through the deliverable, they will be given rewards to mark progress.

When a player completes a quest, then they will be rewarded, for example the initial quest will be to report to the help desk in the Atrium, when they do this they will be awarded with a map, which will both show them they have completed the task, and help them with the mod.

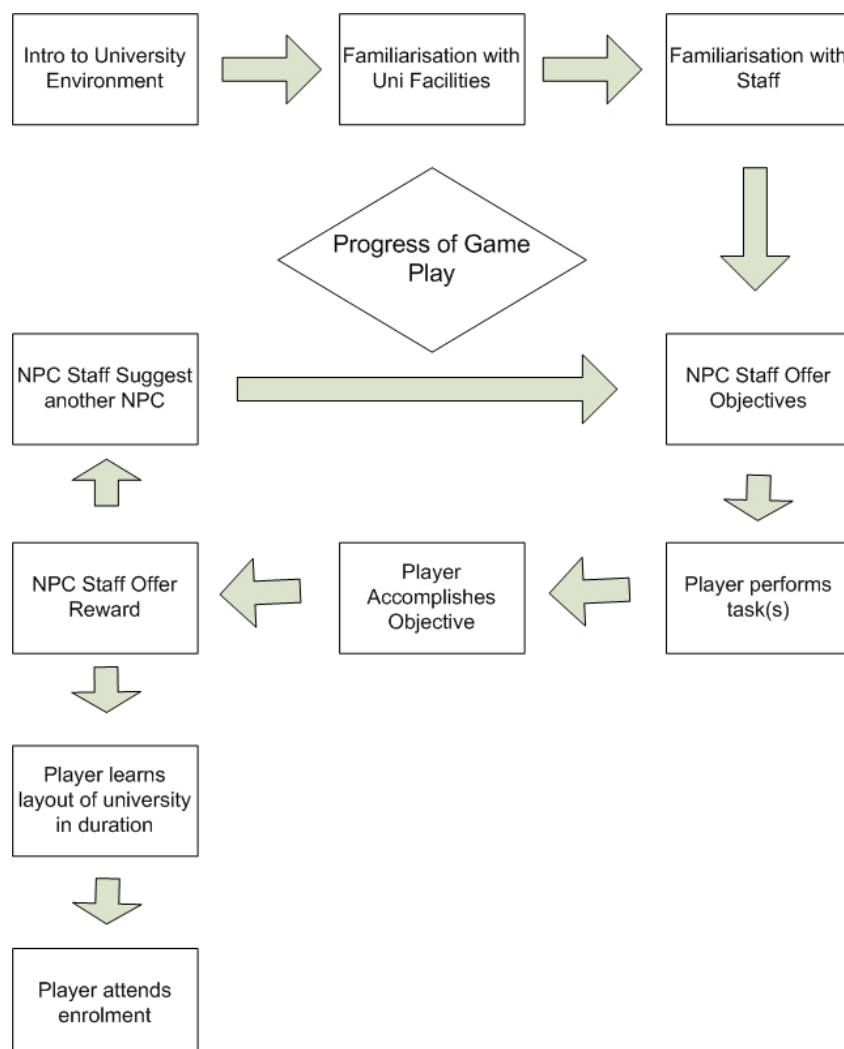
#### 2.4 Navigation

We will be recreating sections of Lincoln University, to a realistic scale so that the user can recognise them both virtually and in real life. Because the intended user will not have seen the campus before we will have to aid them on their navigation around.

In order to do this we will implement a custom map, which they can look at, during any time, the map will display their current location, goals etc.

They will also have an on screen arrow, pointing the way to the next location, so that they simply have to walk from A to B.

A better description of the processes the player will go through in terms of game play and recurrence can be shown in this flow diagram:

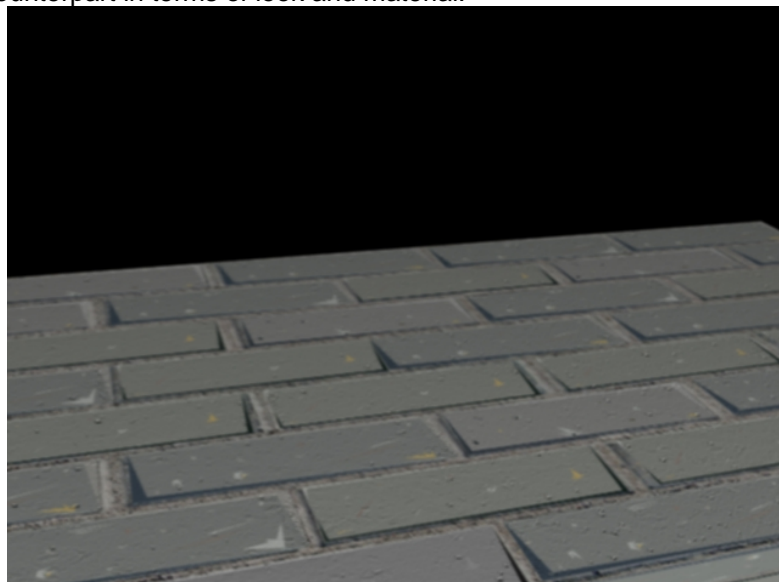


### 3. Design Detail:

As per our initial requirements, we wanted to represent the university in a largely inspired TF2 style theme. To this end, all of our assets will need to be presented as such. Below are some concepts of character models for the main NPC's and characters that we will eventually include within the levels:



We also decided to start getting into the habit of texturing in a cartoony style like TF2, although quite vibrant and sticking to a particular colour palette, the textures should represent their actual counterpart in terms of look and material.



Even though our actual maps are to be of the University Campus, we still require an overview of the schematics with a reference of where the entities within the levels will be. Entities such as NPCs, props, anything that has a solid yet abstract effect on the overall gameplay will be the only ones noted, as dynamic entities will not be shown in the event they are changed and such. These are attached at the end of the document.

Map Creation Examples (Atrium, **MHAC**, Quad, Sports Hall)



#### **4. Third Party Components:**

##### ***Valve Hammer Editor***

This software will be used to create a large portion of our project, it will be used as the main software for development, allowing us to import custom models and textures, as well as create the playable final project.

##### ***Autodesk 3D Studio Max (and .SMD Importers/Exporters)***

This software will be used to create several custom models, to use in our final project. This software is widely used in the game industry, due to the scope it gives to modellers. Therefore we will use this to our advantage to create more realistic components for our project, e.g. monitors, doors etc.

##### ***Adobe Photoshop***

This software allows us to create custom textures for use in our final project, these will be used in both 3D Studio Max, and Hammer. We will use the software to create the desired “Cartoony” look of our final project, by editing textures that we import from sources such as digital cameras.

##### ***FlashFTP/SmartFTP***

This software allows us to upload, and download files to a server, allowing the whole group access to files, updates, documentation etc.

##### ***Visual Studio 03/05***

This software enables us to edit, and compile code to implement custom functions to our project, in order to attain a more in depth modification for our final artefact.

##### ***GUI StudioMDL 2.\****

This allows us to edit the Valve Texture Format files, as well as compile 3d models for ingame use in a complete package, as opposed to command line methods. Also contains the VTF Shell Extension.

#### **IMPLEMENTATION:**

##### **5. Assigned Task Justification:**

*Included at the end of this document is a Gantt Chart* describing how our group’s work is divided out over the duration of the unit, so we can accurately visualise who is “slacking” and who is doing what they ask. This way, in the event of problems with members we can look at the chart which in itself a living document, and then be able to report back to whomever to make amends.

The tasks were divided up with two things in mind firstly the abilities of the individuals at hand and secondly the amount of work that needed to be completed. The group members whose competencies lied the closest to certain tasks were assigned to those tasks, and then these tasks were divided up where necessary due to time constraints For example: the two most competent mappers were assigned two separate areas each so that the modellers and texture artists would have spaces in which to place their objects as soon as possible.

Some tasks, such as testing, which require little skill and instead rely on the brunt of numbers to be properly accomplished were assigned to the group as a whole. In the case of testing this means that the statistical chance of uncovering errors is significantly increased against having only one or two dedicated testers.

##### ***Allocated Positions:***

**Ken M Gorton – Audio Technician**  
**Toby Arnold – Asset Director / Artefact Creation / Coding**  
**Jason Gunn – Asset Creation / Coding**  
**Mitch Williams – 2D Artist / Asset Creation**  
**Stuart Horwood – Art Director / Level Design**  
**William Davies – Documentarian/Script**

**6. Configuration Management Plan:**

To create our mod, we will use the source SDK. Creating a mod within the source SDK will automatically create two folders where our game files will go. When the whole group is happy with the artefacts and code that we have created, they will all go into these folders.

However during the creation of the artefacts and coding, the files will need to be stored in one place making management of the files very easy and sufficient. This is because; firstly to allow the entire group to view what has been created, allowing them to give their opinions and approval. Secondly, each member will be able to backup their files in one place.

We have chosen to store all our files on the web. We will assign web hosting accounts to each member of the group which will be located on one server. This will allow each member to upload their files, and for each member to look at each other's files. This way of storing our data will be secure and safe as the members in the group will be the only ones who can edit and view it. This will also give each member the ability to access all the files from anywhere.

An internet forum will also be set up in conjunction with the web server. This will allow the group to show other members what they have already done, and also express their opinions on other group members work.

The main reason we have chosen this way to manage our files is because of the circumstances we are in. As it's a group project, members will spend more time apart doing their assigned work separately than they will do the work together. The way we have chosen to manage our files will allow the members to view everything that everyone has created from anywhere and talk about it.

**7. Quality Plan:**

We will gain confidence that each part of our system works as designed by rigorously testing the mod up to the alpha version. Once the alpha mod is complete we will start working out any bugs or making changes to parts that may not look right, or those that are not properly scaled.

As our mod is based around the university campus, we must make sure that we make it geographically correct by getting university floor plans as well as our own photographs. Once work has been completed and a beta version is released each member of the group will then individually go through all of the files and code for our mod to look for errors or parts that may need to be changed, also each member will individually play the game and intentionally look for any bugs or glitches and take down notes on how the bug happened and where in the mod, so this can then be sorted.

We will also introduce 3<sup>rd</sup> party members to play through and test our mod, we will let them give there own feedback on how we can improve our mod. We can then continue work to make a final release which will also be tested. If we continually test our mod then we can be sure that each part of the system works as we originally intentioned, which in turn helps us make a quality modification of the Half Life 2 engine.

There will also be a chart that we will produce that enables us to identify key milestones in the development and decide if the work produced is complimentary to a standard we want. The Gantt chart will help in this respect so that we can identify whose job it was/is to do the aforementioned task and follow up on it.