

Ascendance

Final Year Project

By Michael Pledge, James Lott and Joe Toomey

Contents

Abstract.....	4
Introduction	4
Research.....	5
Method	13
Content, Structure and Functionality	23
Look and Feel, Interface design	26
Usability, Audience Testing and Results	30
Conclusion.....	35
Project Management	36
References	38
Appendices.....	40

Abstract

For this project we aimed to create a 3D game which would be of high quality and have a high entertainment value. Our motivation for creating this was to create something that we all have a large knowledge about and also to advance our knowledge in programming and modelling. The three main areas of work were environment design, character design and JavaScript programming.

The team felt that the Final Year Project provided an excellent opportunity to create a team based project like this. Once the project was done the team planned to use their creation as part of their portfolios.

Introduction

The team had various reasons for choosing this brief. They wanted to create a 3D role playing game (RPG²) that would appeal to a wide target audience; be enjoyable to play and would match what the current independent market in scale and functionality. As the team are enthusiastic gamers their selves, they hoped that this combined with the knowledge and experience gained from the course would help them create a game that would appeal to other fellow gamers; that would be aesthetically pleasing and easy to navigate.

They wanted to develop skills and consolidate the knowledge gained so far on the course in programming, 3D animation and graphics. When embarking on this project; they were aware that an endeavour of this scale had not been completed by undergraduates in the past. However they wanted to challenge themselves with an ambitious project to fulfil their aims both personally and academically.

There was not a 'problem' that needed to be solved with this project; instead of taking on one of the briefs suggested by the Department of Electronics the team decided to create a project that would prepare them for the industry's rigours; rather than provide a solution to a pre-existing problem.

One of the main aspects the team wanted to explore was the final year project as a wider community effort; a project that their friends and peers could contribute to. They hoped that by making the project something that other people could feel part of it would be a social endeavour, as well as academic. Only with the input of others were the team able to

make an RPG on the scale they achieved; for example help with the beta testing and the voice acting.

All three team members had individual skills they wanted to use and improve on; as well as what they achieved as a collective. The format of the brief enabled them to use these skills. Michael wanted to expand upon his knowledge of programming and the gain the experience of creating a large, in-depth product. The challenge for James was to create an extensive environment suitable for a games engine in one particular style. Joe was looking to use his organic 3D skills to create a variety of different characters suitable for the game. With these individual aims in mind; they believed that by using them collectively they would achieve the product they wanted.

The team have been interested in working in the games industry throughout the course. They knew that the games industry is difficult to get into, and they would need a vehicle to showcase their skills if they were ever to make this ambition a reality. In the current economic climate and with thousands of students graduating every year interested in the industry, they realised that they needed to demonstrate practical knowledge and skills. By having a physical product to present to prospective employers, they believed this would give them an advantage in the jobs market.

While the team did not have a problem to offer a solution to; they used the project to provide themselves with practical proof of their skills and abilities to show employers, as well as to offer something new to the independent games market. They were interested in experimenting with the project as a social enterprise, and by treating it as such they were able to get a wide range of voices, as well as effective and useful feedback. The original proposal is included in the 'Appendices' and the team felt they have on the whole fulfilled its objectives.

Research

With the recent popularity in video games, you find that more and more genres of games are being brought out. The game industry is constantly trying to create new and unique games that will impress and encapsulate the player. The most common way is by creating hybrid genres that use features from another genre, or is just a previous genre extended in a

new way. Using RPG's⁵ as an example, you will find that it has firstly has been split up into cultural differences with the choice of Western (WRPG), Japanese (JRPG⁶⁺⁷) and Eastern (ERPG). These three types of RPG came into existence due to the differences in the development between cultures, having been created for different cultures they work towards different target audiences and have different outlooks on what a game should be like. However despite these being created for a certain culture, all three are successful worldwide.

The three culturally different genres are then split up into many hybrid genres including: Action role-playing games, Role-playing shooters, Adventure role-playing games, Dungeon crawlers, Massively multiplayer online role-playing games, Tactical role-playing games, Turn-based role-playing games and more. FPS, RTS and others are also split into just as many types. Having a clear focus on what type of game you want to make from the outset is vital, as picking features from across the board can end up with a disastrous outcome. This also allows you to research into where previous games of that genre have succeeded and failed. Before the team started researching examples and gathering ideas of what they wanted to do, they had to first discuss and choose a target audience to aim and design the game for. Upon discussion they decided to aim the game for people who have had experience in gaming, be it RPG or not from the age range 14- 25 and play on the PC. The main reason the team picked experienced gamers rather than novice gamers is firstly because it would be a more realistic target in the industry and secondly, as their university class consisted of many gamers, they could keep in close contact and maintain communication with a similar target audience all throughout the project.

With the target audience decided, the team had to decide what genre the game would be. As they didn't have the time or manpower to create multiple open world environments, they decided to look at creating a JRPG by revolving the game around a linear story with focus on character development and side quests. With this in mind the team set out to discover the background behind JRPGS, looking at their pros and cons. Firstly, it is important to have a thorough understanding of what a JRPG is, as quoted by the n4g.com website; 'A JRPG is a video game genre made in Japan where you take the role of a predetermine character or characters in a heavily story driven experience.'



Final Fantasy XIII-3

http://www.play-mag.co.uk/wp-content/uploads/2009/12/final_fantasy_xiii_livefeed.jpg

Although generally they are created in Japan, any game that follows the style also falls under this genre. JRPGs were created in the 1970s; however they did not become popular until 1980 with the titles 'Wizardry', 'Ultima', 'Dragon Quest' and 'Final Fantasy'. The combat system and graphical style do not affect the cultural genre, there have been as many famous turn-based as there has been real time JRPGs with a few JRPG shooters as well. JRPGs also do not take some of the elements from traditional RPGs; in the traditional form you create your hero and create a story for them and play throughout their story, playing the game the way you want. JRPGs take away a lot of the freedom that you get from RPGs, placing you in the role of a predefined character with a full history, lack of customization and in most cases a linear story with little room to take control and make it your own.

A few polling sites were looked at, including 'www.misterpoll.com⁹', 'www.uk.gamespot.com¹⁰', and 'www.gamasutra.com⁸'. They discovered that 'Final Fantasy 7²⁴', 'Zelda Ocarina of time¹⁺²⁵', and 'Chrono Trigger²⁶' came up as popular favourite JRPGs on multiple occasions. As these games are very old and from the Playstation and N64 era, they also searched for modern JRPGs with 'Valkyria Chronicles' and 'Lost Odyssey' coming up as recent successes, both being in the next generation era. Having looked at the obvious similarities, first there are three turn based battle systems and two real time; they are all based in fantasy worlds with two combining swords and guns and two including impossible technology, four of them include magic. The five games are in the style of anime and they all have a clear enemy and plot to follow.



Zelda: ocarina of time

<http://www.emulanium.com/images/n64roms/z/the-legend-of-zelda-ocarina-of-time-2.jpg>

After having a more in depth look, the team looked at some of the features that made the games so successful. Using Game spot quote as a main source of reviews they looked at the three old generation games, starting with 'Final Fantasy' which received a Games pot score of 9.5 and a user score of 9.6. It was highly rated due to its storyline and combat system; 'best aspect of 'Final Fantasy²⁴' VII is the plot that these peerless aesthetics help weave' and 'perfectly fusing typical RPG complexity and accessibility' reference needed. 'Zelda²⁵' received a Game spot score of 10 and a user score of 9.7 and was described as having 'perfect harmony between combat and puzzle solving'. 'Chrono Trigger²⁶' received a Game spot score of 8.5 and a user score of 9.3 while being described as having 'dense, rewarding RPG gameplay; a memorable and endearing cast of characters; a complex storyline; vibrant, expressive graphics; and one of the most cherished game soundtracks in history.' reference



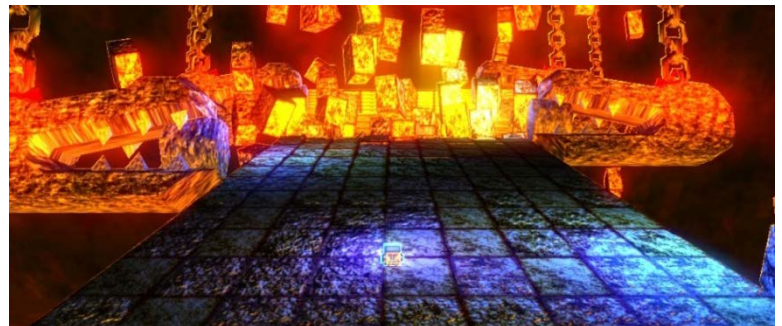
Final Fantasy 7

http://www.teamteabag.com/wp-content/uploads/2008/07/final_fantasy_vii-combat.jpg

Between these reviews it was apparent that storyline was one of the main features that made these games as popular as they are; as well as having a twist to the game, 'Final Fantasy' had a brand new dynamic battle system, 'Zelda' had the puzzle aspect and 'Chrono Trigger' had the time travel. It would thus appear that dynamic aspects such as these are factors which made them much more popular than the rest of the market.

The problem with looking at these games is that they have been developed by large multi-million pound companies, with a huge team behind them and access to the best resources and equipment of their time, not to mention a longer timescale. This meant that re-creating a game like that would be near impossible for the group, so they looked at some smaller projects created by students to see the kind of level they should be aiming at. Starting with their own course they soon found that no one had attempted to make a full 3D game to the depth that the team did; there had been a share of flash games and interactive programs which the team looked at but couldn't compare with their own project. The 3D projects made by past students have been high polygon animation making it difficult to compare

with as the models have much more complex shapes which the team would have to replace with clever texturing or lighting to get the same effect. Looking at projects from across the world¹², researching online for 3D games created by students in a similar position to the team



Igneous

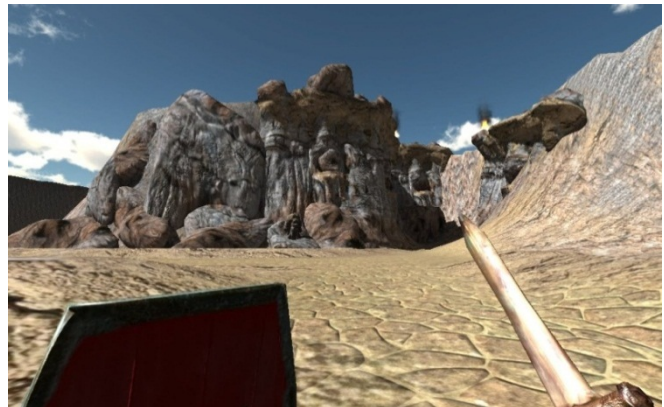
https://www.digipen.edu/fileadmin/website_data/gallery/gallery_upload/igneous_InGame01.jpg

returned much more promise. There was one project in particular that they liked called 'Igneous¹³', an adventure game where you must escape a raging volcano by navigating your way through various levels. This game had a great feel and had a good mixture of puzzle mixed within the game, making it very enjoyable to play.

A great source of projects made by small groups and individuals was the Unity website itself; an interesting short demo which Michael found was called 'HackN'Slash¹⁴'. HackN'Slash was a only a short game where you walk around a level chased by enemies made of rock, but it was made by a novice to Unity and was focused more on the artwork, this was mirrored in several other projects and gave us a glimpse of what can be achieved by new users who only

have experience in the 3D side of the software. They also looked at fully fledged projects made in Unity such as ‘One Night¹⁷’ and ‘Skull Crack¹⁹’ which are bigger and more managed projects being created by a team of experienced users and

although they lacked the knowledge this was the type of level the team aimed for.

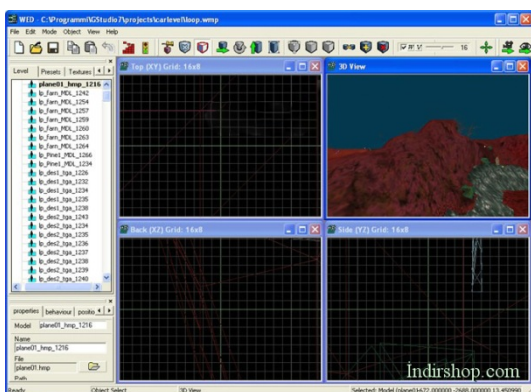


hacknslash

<http://www.preik.net/unity3d/hacknslash.zip>

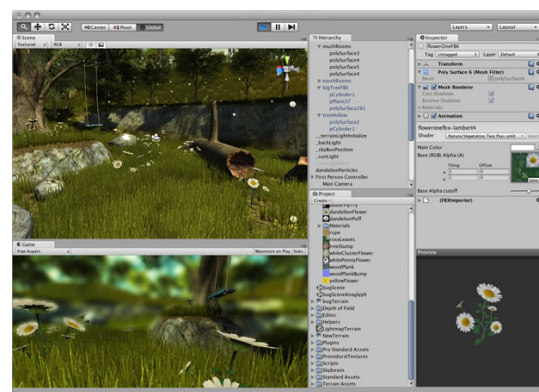
The projects that the team looked at really showed them what could be achieved in a more realistic manner; large production studios create games that the team would never have been able to match at their level and resources. Looking at developers in the same or a similar position to the team not only gave them more confidence into carrying out the project but also a realistic outlook on what the project would end up looking and playing like.

The game engine is the heart of the game; it supplies the functionality for the rendering, collision detection, lighting, physics, audio and networking so choosing the correct engine to fit their needs was very important as it would have been very difficult and time consuming to change later on. The considerations while choosing the engine were their abilities at mesh rendering, ease of use, plug in ability, support available (community and official), water shaders, languages supported and that it was free to use. With previous experience in 3D game engines, Michael had a good idea where to begin and had knowledge of what was available, quickly narrowing the choice to two main choices: ‘Game Studio A6’ and ‘Unity.’



Game Studio A6

<http://www.indirshop.com/resimler/8b543f1f26.jpg>



Unity

<http://blogs.unity3d.com/wp->

Both fitted the requirements, the differences being Michael had experience with 'Game Studio A6', while Unity had an easier to use interface and better integration between different software which made it the logical choice. The problem with Unity however was, like all independently created game engines out there were two versions, the free version and the Pro version. The cost of the Pro version was too much for the teams' budget, so they had to use the free version. This of course limited them from using several main features; the main being multiplayer, which didn't affect them but, it also disabled shadows, reflections and transparency. Although not ruining the game it did take away from the experience, so what they decided to do was create the game in the free version then use the 30 day trial of the pro version to export the game with the extra features.

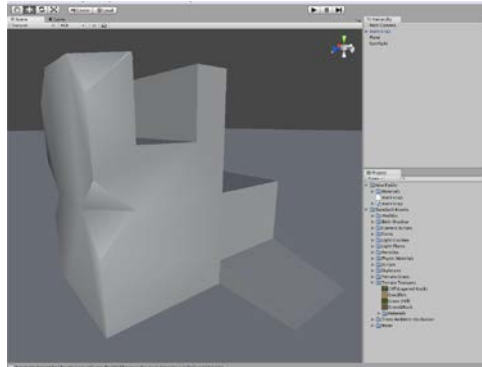
As the game engine was completely new software to them they took the time over the summer to run multiple tests to make sure it was suitable and would perform as they needed, and secondly Michael followed 2 tutorials and made several mini projects without guidance. While testing the program the team ran the engine through multiple situations which they believed would appear in the game. This allowed them to know what was going to work and how efficiently so carrying this out as soon as possible was critical.



This is an example of the 3D project Michael made during the summer

They started with simple tests such as importing from Maya; they would then import it with different file extensions and see which would work best, the best file type was the '.fbx' file. After being happy with the importing they started manipulating the mesh in foreseeable and unforeseeable ways, they started with a basic mesh and started performing different actions on it, using simple tools such as extrude, manipulating edges, polygons and vertices before moving onto more advanced features such as Boolean, shatter and nCloth. The results came back mostly positive; the first thing that was realised was that they had to delete the history and keep the outliner empty,

otherwise strange artefacts would come through inside the game engine. The only other thing was importing dynamics, the team had already presumed this but the test confirmed it; luckily the software did supply some similar tools to nCloth.



Example of a mesh which has been manipulated by the extrude, chamfer, Boolean, merge vertices and cut faces tools.

When they had finished testing the mesh they started experimenting with textures, the engine uses shaders to create different effects such as reflection or transparency and these shaders can have animation coded in, so a lot can be done if you know how to code the shaders, however they just wanted to use the engines default texture shaders so they set about to make sure they could do everything they needed such as; covering file formats, transparency, normal maps, specular and water animation. Everything came back positive except that the basic version of Unity didn't support transparency, so for the majority of the game it had this disabled. Lastly the team checked animation support with a basic humanoid, they rigged and animated a model then imported it, they tested with and without 'ik' handles and weights, bone animation and transform animation; they didn't test too much in this area as everything they would be doing would be very basic.

The research for the game never truly ended, as sections of the games opened up or where the group felt to change the direction of the game it all required more research to be carried out but on more of the style and understanding how things work, an example being the crane in the docks something the group always planned to include however at the time of modelling it more research was carried out into the inner workings and how such a mechanism was created. However with the foundations researched this meant the group had the means and the understanding of how everything needed to fit together and a shared visualization of the end product slowly coming together. Having this visualisation meant that James and Joe could finish carrying out their research on

the modelling side with a clear direction in mind and understanding of the game genre they were designing for, as well as how they needed to model for the game engine to display correctly.

Method

From the research carried out and the general gaming experiences of the group, it was decided that the RPG (Role Playing Game) genre was the best one to encapsulate the vision of what the game should be like at the point of design. Setting the game in the past removed the necessity of having guns and other modern warfare equipment. The objective of this was to make the characters interact more, not just automatically reaching for weapons to eradicate their enemies in a mindless way. The decision to make the environment a broadly Tudor design was mutual; all of the team members found this style aesthetically pleasing and believed it would suit the RPG format.

The team decided on this idea through a series of meetings, and through communication on Google Wave; which proved a valuable resource throughout the project for sharing information and ideas between team members and also with the Team Supervisor, Bret Lu. The original design process was more of a team process than the later stages of development of the game. This was an inevitable change; the processes of coding and modelling the game are obviously more solitary than the period in which the design parameters were agreed.

In the first instance, the team tried collaborating on an internet forum set up specifically for this project. This was a useful tool as they could upload reference pictures and discuss matters like the storyline or any troubleshooting issues. However they decided after some weeks to use 'Google Wave' instead. Google Wave was chosen because it updates in real time and it is easy to upload and keep files inside of a 'wave.' This is useful because the team often worked individually and they could just update work online and Google Wave would notify the other team members.

As Michael was in charge of using the Unity programme, a game creation engine, he was in charge of looking at models created by Joe or James. They uploaded their models and texture maps to the wave and it would be Michael's job to check if the scale was correct and that the texture fitted in with the current models already created.

The team decided that Google Wave should be set as their homepage so that whenever they were on a computer they would be updated on the progress of the project. It also proved to be a great tool for editing storylines as new ideas came up for the narrative.

The original story was formulated during team meetings from imagination; this includes everything such as names of places and people as well as dialogue.

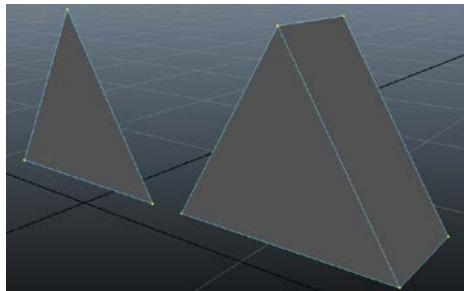
The team then set about coming up with short pieces of dialogue and asking their friends to audition for them on Facebook. To do this a Facebook group was set up and participants were then asked to read lines for whatever character they liked the sound of. Most participants tried to voice act as many roles as they could, making it easier for the team to decide who to cast.

During the summer of 2010 Michael practiced working with the game engine Unity, firstly following a tutorial where he had to create a simple platform game. The aim was for the player to explore the environment, collecting items whilst trying to avoid enemies which would attack the main character. Michael then moved onto creating a small environment which was more puzzle based; the aim of the game was to collect 3 crystals. To collect each crystal you had to solve a simple puzzle; this was to test Michael at his programming skills as he had to have several variables inside the code working together. These two projects as well as learning basic Javascript skills during his degree helped Michael gain the skills necessary to work on the final year project.

Buildings were created using Autodesk's Maya. Maya is a program capable of producing 3D scenes for use in video media, games and architecture. It has a customisable taskbar you can add any function to increase workflow for whoever is using it and for what purpose. The team chose Maya for the project as they all had experience using it and it has useful tools features for lowering poly count.

Making sure the poly count is low for buildings and models is vital for games; if poly count is too high the game will not run smoothly. Throughout the project the team had to be very careful not to allow buildings or models to have too high a poly count. Poly count is defined by how many vertices a model has; vertices being markers which define the corners of a model. For example, a simple 2D triangle has 3 corners; henceforth it has 3 vertices and a poly count of 3. A 3D triangle would have twice as many as it has 6 corners. The 3D Triangle would also have five faces rather than one but most game

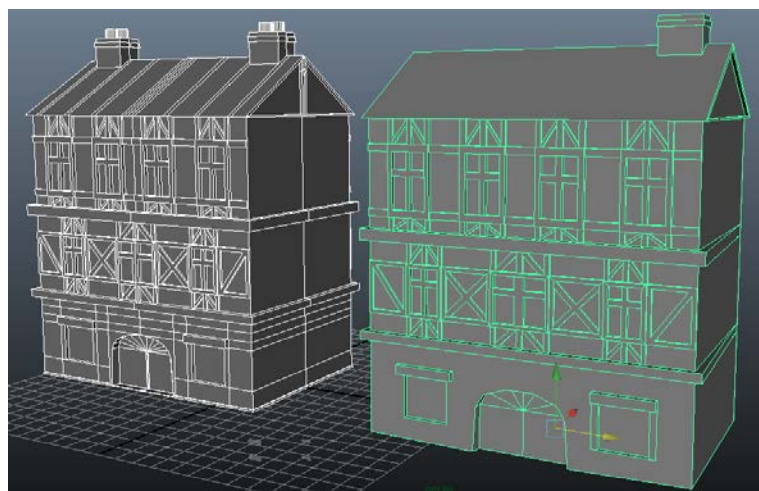
engines including Unity are affected only by poly count and not by how many faces it has to render.



An example of a 2D triangle and a 3D triangle – The yellow dots represent 'Vertices'

The team agreed that a poly count of up to 6000 - 8000 would be allowed for character models whereas buildings were only allowed a maximum of 2000 polys.

One feature of Maya which makes it unique is it generates both sides of a face inside the program, most other 3D software and all game engines only ever show one side. If a 2D model, say a flag, is inserted into the game engine it can be viewed from one side but on the other side the model would be completely invisible or disappear. So when models like flags were created they were made on with a 2D plane, they were then duplicated and mirrored and merged with the original, doubling their poly count but making them visible from whatever angle they were viewed at. This was one problem that the team faced but was easily overcome.



Example of a non-optimised house (left) next to a house optimised for a game engine (right) both houses are exactly the same shape and would look identical in-game, however the house on the left would be too complex for the game engine.

The method in which the buildings were created was to take one or several reference pictures and create a model distinctive enough to distinguish it from a building that was already made, but still in line with the style of the game. The reference pictures were

sketches drawn by James, Michael or Fiona Brittle, who had drawn a couple of landscape views of her idea of what the town centre would look like. The sketches were then replicated into a 3D model by James who tried to ensure the scale of windows and doors would be the same for all buildings. The poly count of these buildings would often go above 2000 during the modelling stage so it was James' role to remove any unnecessary vertices on the buildings. Removing the unnecessary vertices made the buildings easier to UV map.

UV mapping is the current way to texture a 3D model and was used throughout this project. In simple terms UV mapping is flattening a model into one 2D representation of the model, scaling all the faces of the model and placing them into a square so that they can then be placed into order or have a simple texture placed on top of them which will be in scale in comparison with other faces.

The UV map would then be exported to Adobe Photoshop which is a graphics editing program. Most faces of the UV map were separating into different areas of the map such as 'dark wood' 'light wood' 'plaster' etc. to help make the texturing job easier. Many textures were obtained before the start of the project by downloading large 'packs' and by searching on specific sites such as cgtextures.com, they were also obtained during the process of texturing. Several of the textures were also obtained by James on his camera whilst on holiday in Greece and in Canterbury as there are many Tudor styled buildings there.

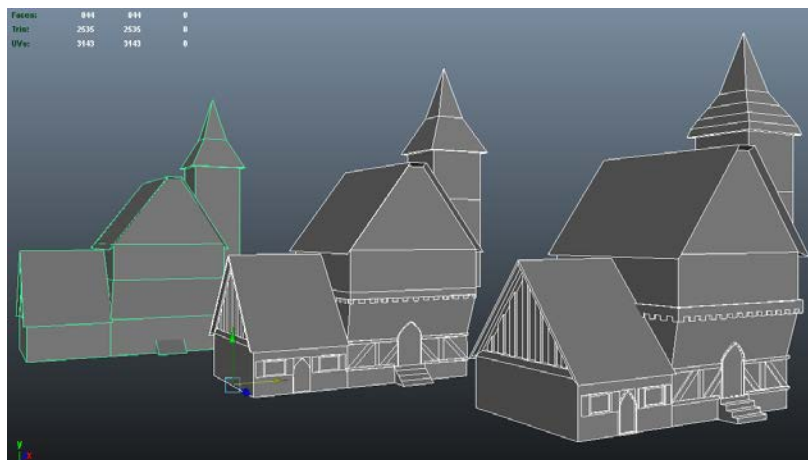
Once the textures had been set up for the buildings they would then be added back in Maya and it would be for James to decide whether the texture looked correct on the building. Often the texture map would be edited and corrected several times before it was finished. James would then upload the model to Google Wave where Michael or James can then add the building into the game using Michael's computer. The building then needed to be placed inside the scene and be bump mapped. Bump mapping is a technique that uses light to create bumps in a surface without adding any vertices. Finally a collider would be added to the building so that the character models could not walk through them.

When the village was near completion it was noticed that the game had started to 'lag' or run slowly. Michael solution to the problem was to create a 'Level of detail' (LOD)

system, which would swap high poly buildings for low poly buildings when the player was far away and medium poly buildings for when the player is closer

The way the LOD system works is by replacing buildings that are far away with simpler objects; this means the computer doesn't have to render many complex objects allowing the game to run more smoothly. As the player moves away from a building, the building itself is replaced by a simpler object which is similar to the original. Because the building is far away it is not noticeable that there is less detail on it.

Unfortunately implementing a level of detail system into the game required James to revisit every single building already created so he could then make two similar buildings with fewer polys. He started first by making the medium poly buildings by erasing unneeded vertices and by flattening wooden beams, windows and doors and merging nearby vertices together. On average this lowered the poly count by about 60%. James chose this method because it meant that the original texture map could still be used, not only did this save time for creation and implementation it also helped the game run smoother as it only had to generate one texture for both buildings.



Example of a low, medium and high poly building. The building on the far right has twice as many vertices than the middle building and ten times more than the building on the far left.

The low poly buildings took more time to create, they basically had to be a shell of the original building but still look similar. Michael decided a good poly limit for these buildings would be 80 vertices, 25 times smaller than the original limit. James decided to use the method of print-screening all the buildings and then using the shot of the building as the texture. He trialled this out on several buildings and it was very effective so this was the method used for creating the low poly buildings.

One problem with the LOD was that all the lower poly buildings had to be very similar to the original or it would be noticeable that they were changing as the player approached them. The most noticeable building in the game where this happens is the church. As the user approaches it seems to gain an extra chunk of wall and as the player moves away it loses this again. Fortunately this is the only building where it is easily noticeable that the LOD system is in place.

As mentioned before, the game engine can only see one side of each face on an object. So when the inside areas of the game were created the faces of the ceiling, walls and floors had to be reversed. To begin with it was trial and error, James created the models, Michael then imported them into the game to see if it looked correct and if not he would tell James what faces to reverse to make the model correct and then reimport.

Backwards faces also sometimes appeared on other buildings and had to be rectified in the same way as described above. It was not until the team were approaching the conclusion of the project when they realised that Maya had an option to view models in exactly the same way as the game engines see it.

Joe's models were made using a different method. As Joe was creating character models he was making what is called 'organic models.' Organic modelling is any object that is not man-made such as trees, animals, humans etc. Additionally to using Maya and Photoshop he also used the programs Zbrush and Autodesk Mudbox. Joe was advised to study anatomy to help with concepts and understand how muscles were built up. He referred to two books to improve in this regard, 'Figure Drawing for All It's Worth' (1943) by Andrew Loomis and 'Constructive Anatomy' by George Bridgman.

He studied and learnt different practices about edge-looping which is of great importance when modelling characters, especially if the models are going to have lip-syncing done. Most of the main characters had appropriate and proper edge-loops as expressions and lip-syncing; this was planned from the beginning of the project. However when it became apparent that the characters weren't going to have lip-syncing done, the lack of this requirement allowed the resistance members to be created far quicker than the previous characters.

The two difficult areas when creating the characters were the noses and eye sockets. The mouth initially proved most difficult at first but through practice became the easiest

out of the three. This was an area Joe had to focus on and had to refine a lot at the start of the project. It was hard not to speed up the process when working on very familiar areas of modelling; whenever this did happen usually it resulted in having to reload a previous incremental save. Maya's default incremental save feature proved reliable, it saved different file names individually when large sections of a character were completed so that Joe could continue work without fear of losing all of his work.

Texturing for Joe was done usually with a mixture of Mudbox and Photoshop, the majority of which was done with Mudbox's projection tool which allows highly detailed imagery to be painted onto models rapidly and usually with minimal artefacts occurring. Most of the images used were pictures of real fabrics, metals and people. Photoshop was used to touch up certain areas and remove any small artefacts that didn't show up in Mudbox.

Reference pictures probably helped most in regards to getting proportion and flow of the typology to a satisfying point; working from photo's always proved far more productive than working from a drawing. Whenever a problem area was proving too difficult to create in Maya usually the best way to keep working was to try creating it in Zbrush or Mudbox and then refining it later in Maya.

Sometimes this proved to be far quicker than creating it solely in Maya. Other times the imported geometry needed a vast amount of work done. This wasn't always discouraging; it helped break up the work at times. There were errors encountered usually when importing back and forth, a common was models not mirroring as expected after being imported. Usually this was solved by simply reimporting.

Mudbox and ZBrush are similar programs made by rival companies. Like Maya they display a 3D model which you can manipulate. However they differ in the way the model is manipulated, ZBrush and Mudbox display the model as if it's made out of clay and allow you to pull, grab and twist the structure in any way you wish. These programs also allow you to paint textures straight onto models which make texturing easier than the technique used by James on his buildings. UV mapping a character model is a harder than making a UV map for a building as a character model has lots more faces and rarely in an easy shape.

Joe mostly used his graphics tablet with these programs as it allowed him to have a more a more tactile input into the models. After he had finished modelling and using Photoshop he would then edit the model and upload screenshots of the models to Google Wave to receive feedback. Towards the end of the project the buildings were finished on schedule so James then went on to create a few character models for villagers, including the Tavern Keeper.

All of Michael's programming was done within the Unity engine using the engines own debugger. He initially chose to use the C# programming language as he had more experience with it. However after his first few tests and upon researching how to make the game he opted to use the programming language 'Javascript.' Javascript was recommended by many professionals on Unity forums because it was less likely to require debugging, ran smoother and is easier to manage on large scale projects such as this. Whenever he was unsure which direction to take with a problem he would often use the Unity forums to learn to overcome any difficulties he faced.

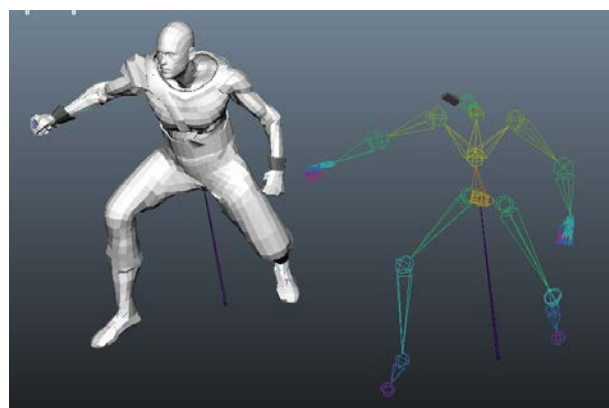
Michael also worked on environmental animation, such as the fountain, the sea and small signs such as the swinging pub sign outside the taverns as well as animations for characters spells such as the fireball and ice spell. He also worked on placing sounds around the environments, such as water noises and birdsong where appropriate. Lighting was also done by Michael, such as fire lamps in inside areas and sun lighting for outside as well as creating a night scene to help develop the story and add intricacy.

Finally the team needed to work on animation of characters. The team were advised by Dr David Byers-Brown that the simplest way to animate all the characters in the game would be to use a motion capture suit. The team agreed on all the actions they needed to record and James agreed to wear the motion capture suit. Michael took the lead role in leading James and pointing out anything he thought needed correction, whilst David, who had worked with motion capture data many times, made sure the computer was taking in the data correctly. When the team were satisfied with the amount of actions they had recorded they then had to go through the data and remove any imperfections. As over 15 minutes of data was captured this was a time consuming process. The data then had to be cut up into each individual action and added to the appropriate character.



James in the motion capture suit

Joe's models also had to be rigged before the motion capture data would work. Michael took on the role of rigging the characters as he had the most experience at doing it. 'Rigging' is similar to adding a skeleton to a model which then controls the movement of the character. Applying a rig to a model is a tricky and time consuming process as it is very easy to pick up unwanted vertices on the skeleton and the model will look strange as one part of its body will be misshaped. The other problem is applying the skeleton to the motion capture data without the animations looking very unnatural due to the model bending its body in an inhuman way. This job took Michael several weeks to complete and the team came close to abandoning the motion capture data in favour of animating everything by hand. Fortunately Michael did manage to get the data working convincingly.



Rigged character with exoskeleton

Throughout the project the team also voice recorded over 30 people (including themselves) so that their voices could be used inside the game as a villager. From the recordings submitted through the Facebook group, Michael and James discussed who

should take what role. The team began work on the script based around the storyline they had already come up with. They asked the chosen voice actors to come in and read some scripts so that the team could have talking characters in their prototype version of the game. Each line of dialogue had to be cut up into individually named MP3's and then individually added into the code so that the line would be read out at the right time.

During the prototype presentation David picked up on the fact that the father, Belzar, did not have a mature enough voice. The team conceded that this was true, as the chosen voice actor was 20 and voicing the role of a middle-aged man. James decided it would be best to ask David himself to voice the father instead as he knew David was a keen voice actor.

Michael and James continued to write scripts for the main characters as well as lines for individual villagers. Each villager had the minimum of four lines to say; two to shout whenever you were near them and two to say when you approached them. At first the team used friends and family to voice these roles although it quickly became apparent that to get enough diversity in voices they would have to start asking a larger variety of people to voice a character. The team then started asking people in the Department of Electronics and friends of friends. Again each individual line had to be cut up into individual MP3's and noise adjusted so they were not too loud or too quiet.

For ambient sounds such as bird song, footsteps and wind, Michael and James took a camera outside and tried to record as many appropriate sounds as possible. The footage was then uploaded to a computer and converted into a digital video; the sound was then 'ripped' from the video and cut up into individual MP3's. Finally the music for the game was bought under a student licence from 'www.audionetwork.com' although the team did at one point consider asking students studying a music course to help with the music too.



The team also ran two beta sessions in which participants were asked to play through the game once before answering a series of questions. The feedback was then analysed by Michael and James before fixing any problems picked up before continuing the work they were previously doing. This is mentioned in detail in the 'Results' section.

Content, Structure and Functionality

As the team started planning the story they all knew it would be very ambitious, one they were likely not to complete telling. In fact they were right, they never finished the full story but that wasn't what they set out to do, they created the story to be cut up into sections, as you progress through sections of the game you go to more areas and meet new characters, this building block style of structure meant they could add the story in section by section as the assets were completed, it also meant that if time ran short they could remove sections of the story without affecting the game play. This structure meant James and Joe could focus on getting the initial town and characters modelled and Michael focus on gameplay thus achieving a game build, which is what they set out to do and then move onto the next section and add the story bit by bit, even though they could have submitted the game in this primitive form, they wanted to push as far as possible and create an impressive game with a sufficient amount of story to tell with it.

You explore the game through your character 'Alrik' and the story is displayed to you like a book, achieved by talking to NPCs and watching game scenes which follow predefined instructions. Talking to the NPCs is the easiest way to tell the story from the developer's point of view, as they are easy and quick to set up but make the game repetitive and boring. Game scenes take a lot of time to set up but create a much more interesting experience for the player and breaks up the mundane process of running around and talking to hundreds of NPCs, so the team had to choose carefully so they didn't spend too much time creating interesting scenes which could be told by an NPC, this type of decision had to be made in multiple scenarios throughout the course of the game.



Alrik talking to an NPC

Another area it affected were side quests; adding simple quests was quick but very boring such as collecting packages and killing x amount of enemies so the team tried to add more exciting quests such as collecting supplies but then being ambushed by the enemy, just to mix things up a bit and break up the mundane tasks.

The features they choose to implement came from research based in successful RPGs and what the team felt important to add; they split up the features of the game into manageable sections with the main ones being social, explorative, combat and system. The social features affected how the player communicated with the world and how he received information back; an important aspect as they didn't want the player to be bored while gaining data on what to do, like more modern games such as 'Mass Effect' and 'Dragon Age'. They decided to voice act all the character's dialogue, this brought life into the characters, making it easier to process the information and keeping the player immersed in the world. This feature in particular received great feedback from both beta testing sessions. They also added multiple dialogue lines per villager and about 3 in 4 characters have shout lines which trigger whenever you are nearby, the function makes the character shout out one of their lines at random, bringing the town alive and adding more atmosphere. Lastly a rare feature is smart dialogue which changes according to what stage of the game you are at and what decisions have been made, an example of this is if you kill the mother rat then the tavern keepers lines will differ then if you hadn't.

The explorative features included the functionality that affected movement and the scenery. This ranged from the very basic animation on the buildings such as the waving signs and smoke from the chimney; which helped at adding motion into a static world, adding more realism, to the larger features such as the chests scattered around the world. The chests contain helpful items to aid in your adventure but more importantly they give the player an aim, a purpose to explore the town. The placement of the chests were carefully chosen and the city built with the locations in mind, the positions are in obvious but discreet places, so that the player will walk past them multiple times but unless they are observant or search carefully they won't immediately find them. The main reason they did it like this is so the player will only find a few of the chests while playing through the story; but when they return later on in the game they might find the chest, making areas more re-playable and less boring. The 'Fable' series also uses this

technique and it works very efficiently. They also had features such as secret items, secondary objectives in quests, plenty of paths and alleys so the player can choose the way they want to go, all these just help break up the game and make the traveling less boring.

With the combat features they wanted a diverse combat system which the player could drastically alter the way they want to play, so developing the features which allowed this was very important. They stuck with the generic levelling up you see in the majority of RPGs and when you level from gaining experience you gain statistic points in the three main areas of combat strength, magic and agility and you also gain a skill point which is used to level up and gain new skills from the skill tree. Again the skills are divided into the three main areas of combat and can be combined into hybrid skills, depending on how you distribute these skills affect how much your statistic points are increased by each level. The statistics are also affected by weapons, helms and armours which you buy throughout the game. The combat is an in-depth system which uses multiple features for the purpose of one outcome, a character that the player has built to the battle style they prefer.



Alrik using Ice magic

The system features covered saving and loading, area changing, options and all the other code that affected the back-end of the game; features that you couldn't so much see but made the game tick, a big one being the quest handler function, that as its name suggests handles all the games quests and calls them when needed. As well as keeping it all under its own class making it easy to update, everything else comes under the Gameplay features such as enemies, shops, items that change the appearance of the character and more. All of these features have been added for one purpose, creating a

better experience for the player. The only features that don't alter the game play are James' and Joe's turntables which are viewable from the main menu; these were added to demonstrate their favourite models without having to play the game.

Look and Feel, Interface design

It was important that the look and feel of 'Ascendance' matched throughout the game, with models, characters and graphics. It was also important that this style portrayed the games' genre and immersed the player in the artificial world. They began by researching the time period they were basing the game on, which was the late medieval period. Although it was based around this time, it was a fantasy world meaning they could change it as they saw fit. Collecting reference images of the time period gave the team a good idea of the architecture and the type of buildings that would be used, but before they could draw up designs or begin modelling they needed to look into a style for the game. The majority of this research was based around games on the market.

The team were all enthusiasts of the RPG genre whether they were Japanese, Western or Eastern RPGs and they all had an idea of how they expected and wanted the game to look. Michael was inspired by the 'Final Fantasy' series, James the 'Zelda' series and Joe more the 'Elder Scrolls' series, each one different to the last, both in style and game play. Agreeing on a realistic version of the JRPG style and gameplay, they compiled a list of examples including games, films and animations. Some of the most useful and referred to were 'Fable 2' and 'Fable 3', 'Kingdom hearts', 'Lineage 2' and 'Final Fantasy', each game contributed to the final product in their own unique way. 'Bowerstone

Marketplace' from 'Fable 2' in particular bears a great resemblance towards 'Ascendance' purely because in the teams' opinion they had created near perfect 3D representations of the same architecture the team wanted to make. 'Fable 2' was also used a lot because of its beautiful

scenery, with beautiful back drops and interesting environments. 'Final Fantasy' also had



Bowerstone Market Place

<http://images1.fanpop.com/images/photos/2100000/Fable-2-screenshot-Bowerstone-outside-the-Cow-Corset-fable-2185953-1280-720.jpg>

aesthetically pleasing environments especially in 'Final Fantasy 13', 'Lineage 2' was used more for the character design, although 'Final Fantasy' also had great character designs they felt 'Lineage 2' had more interesting designs especially with the armour varieties and the intricate details of the characters.



Character Design from Lineage 2
<http://www.gameogre.com/lineage2c.jpg>

We also looked at a lot of games that weren't used towards the project; these games just didn't match the style the team was looking for or didn't match the time period. Examples are 'Star Ocean', a great RPG but its set in the future with a very clean bright style, 'Lost Odyssey' one of the favoured games between the group however its fantasy based world did not match their own and 'Dragon Quest' a classic RPG but its cell-shaded style again didn't match their own. The three examples were all great games and they did look at features of each however for the art and design they didn't suit the purpose.



Dragon Quest VIII
<http://images.slashdot.org/articles/05/12/quest2.jpg>

Games weren't the only sources of artwork the team drew upon. They also looked into movies and artists, and with the artists in particular the work was used as reference on frequent occasions. 'Google Images' was the first place they tried when looking up reference images and it produced some fantastic results, however they were always one of images. It was when the team started looking into other websites such as 'Morgue File' and 'deviantArt', sites where artists come together to share their work. While browsing through the art they discovered that certain artists were popping up on frequent occasions; two in particular were 'SvenArt' and 'Cuculus' both of whom had galleries full of art that matched the teams medieval time period and style, they were

both referenced on multiple occasions and had buildings based upon their work. Movies were the least referenced source as they did not give enough focus to the buildings; they did however show off some good atmospheric effects.



Digital art created by SvenArt

http://fc04.deviantart.net/fs6/i/2005/033/9/0/Ankh_Morpork_by_svenart.jpg

Having decided on a look for the game they had to make the environments feel alive and realistic. This was achieved by giving the areas an atmosphere and certain feel as the player explored. Each area had its own unique story and feel and this was based on the teams' story line which gave details on events happening in each area. These events are what would affect the atmosphere; the story itself has a few dark twists and they tried to create the town so it could encapsulate both a happy explorative mood, but also a darker sinister side. This was mainly done with the transition between day and night ,

but they took this into consideration while designing the scene by creating small alley ways that twist through the town and creating the streets close together with over hanging buildings, so that if they needed they could use this to alter the mood. The docks are kept out of the main story and used to host side quests more than anything; which don't carry much emotion with them except



Town map design, showing the plans for alleys

maybe one or two, so they kept it happy with a beautiful view from the top, a nice open-plan area with plenty of people keeping you entertained with a clean environment and plenty of ambient sound to create the atmosphere. The castle they wanted to make very dark and sinister, achieved by large sharp jagged rocks and a dark colour scheme

creating an unsettling feeling for the player. Music was a key part of creating atmosphere and matching the correct track to each area was an important task which had the ability to create or break an area.



Castle surrounded by large jagged rocks and a dark colour scheme

For the graphical user interface Michael started with researching what functionality previous games have used and what specific menus they had in common whilst at the same time applying their own needs to it. Apart from the heads up display including health and magic bars, hotkey icons and menu links they also needed to create various menus with important functionality such as options, inventory and statistics. They also created help-screens and dialogue boxes. After knowing what they needed they started researching game menus specifically in the RPG genre, a few well designed menus the team found were the 'Diablo 2' menu, a nice rustic design which flowed and blended well with each, there's a lot going on but you easily and quickly learn where everything is placed. 'Dead space' had a more modern outlook on the menu design actually making it appear as if it's floating in 3D space in front of the player.

This keeps the player immersed in the game where most menus break this as you have to navigate through the menus and the game stops playing, 'Fable 3' also took this approach with their minimal menu design replacing it with in game objects. After conducting a broad search they narrowed down their results quickly as they agreed on a style. The style decided on was a very worn old effect, a texture that people associate with olden times, mixed with a brown and dark orange colour scheme with a very dirty feel to it making it in style with the models and feel of the game. This is a very

traditional style and uses textures such as old worn tomes and paper, the style is commonly found within medieval RPGs and suited the style and age of their own game. After studying a few designs they liked such as the 'Dragon Age' menu which was an ideal reference for what they wanted and needed both in style and functionality they started making a few mock-up versions of their own in Photoshop, starting with a few basic designs testing colours and shapes. After choosing the design they liked they started cutting it up ready to be imported in the game; however the design agreed upon was later on replaced for a new better design.



Example of Dead space 2's 3d world GUI

<http://www.thebuzzmedia.com/wp-content/uploads/2008/11/dead-space-holographic-hud.jpg>

Usability, Audience Testing and Results

During the making of Ascendance the team asked for constant feedback from those around them. It was decided that they should also run 'beta sessions' to obtain more constructive criticism in a more formal way. The team decided that the best way to obtain feedback from their game was to run sessions where users would be invited to play a prototype version of the game and for them give feedback using a questionnaire.

The team decided that to test people in the age demographic of between eighteen- twenty five year olds as they would be their target market and would be easiest to get a large group together at the University. Michael was the member who would benefit most from the feedback as he was designing the interface and programming the gameplay. Together the team drew up a comprehensive list of questions on gameplay, and questions for feedback on the repetition of models.

The first beta session was eight weeks before the project was due to be finished; giving them time to make any major alterations to the game. In the first session the team managed to obtain results from twelve different people aged between nineteen and twenty three with an average age of twenty one. The team printed off questionnaires and handed them to each participant before setting the game up on the computer the participant was using. The testers were asked not to look at the questions before either completing the game. Once the participant had finished the game they were given a pen and would have a quick chance to ask any questions if necessary. The teams did ask their supervisors Bret Lu and David Byers Brown to attend both events but unfortunately neither could attend either session. Once all the forms were filled in the numbers were added up and the feedback assessed.

The game was well received in the first beta session; however user navigation was an issue; the testers continually got lost and couldn't find their starting point or continue their quests due to this. To help resolve this issue the team decided to create 'pop-up' windows to aid players understand the quests and to help with general orientation. In the second beta session these windows helped the majority of testers understand what they had to do and where they had to go next. Possible improvements would have been to add guide arrows to show where the player needed to approach to continue the story. Michael also added blockades in certain parts of the game so that the user couldn't get lost or go the wrong way.

The testers noted in the first session that the game started to drop frame rate in certain areas. The team looked into these areas to see if the issue could be resolved. The usual problem was that there were too many lights or lanterns in an area. To resolve these problems the team used a technique called 'baking' where the computer analyses where the light should be and records it, rather than rendering it in-game.

The testers also mention that in the first beta test the basement was too dark to be able to see around and attack the rats inside it. Brighter lights were added and then 'baked' into the game. In terms of taste, the visualisation and repetition of set assets in the town were commented on favourably. Exactly the same problem happened with the castle level in the second test; again the team added more lights and baked the scene.



Basement area

The beta test sessions proved invaluable and helped not only discover bugs but also encouraged the team to improve upon the games in ways they had not thought. For example a change in controls was implemented after the first beta took place to address some of the feedback.

Originally the games' camera moved in whatever direction the mouse was moved and this led to players getting the camera moved constantly in the wrong direction and then finding it hard to negotiate the main character around the map. This was subsequently changed for the second beta test and most testers commented favourably on the new controls. Michael also added hotkeys to each menu in the game so that players can type a certain letter and it would bring up its corresponding menu.

Gaps between buildings and walls were sometimes found by the testers; this allowed them to get to un-designed parts of the map which were not meant to be accessed. They were either stuck inside a ring of buildings or would have the choice to go back or walk off the side of the whole map. Whenever gaps were found the team asked the beta testers to note down where these were and also provide a screenshot. Additionally to gaps, some buildings had not been set a 'collider' which stops the game characters walking through them. This meant that testers could walk straight through these buildings and be able to access parts of the map they were not meant to. Again, the team asked for locations and screenshots if the testers found a building without a collider.



One of the beta players stuck in an area the player was not meant to be able to access

Although the team had music ready for the game, when the game was exported out for use on other computers unfortunately the background music no longer played. This was because of a slight error in the programming. This meant that the question about the music on the first beta form had to be disregarded.

The majority of the characters' voices worked fine. The biggest problem was that the volume of these voices differed quite noticeably which made the game feel more unrealistic. To improve this James and Michael went through the sound clips again and reduced or increased volume where appropriate.

Another problem with the voices is that they sometimes overlapped each other making them both inaudible. To prevent this Michael set up some code inside the game to stop this from happening. As well as this sometimes characters would say lines they were set for other characters, this was down to the way the programming had been set up and was fixed. The radius in which characters could talk to you was shortened so they would be less likely to talk over each other anyway.

During dialogue subtitles would appear at the bottom of the screen. This worked perfectly throughout the two beta sessions. However when it appeared on screen the main character would continue his walking animation on the spot. This again was rectified with a quick change to the coding to make the character stand still when spoken to. This also stopped the player being able to control the main character from walking off whilst in chat. Also inside these subtitles the text could be highlighted by the mouse, we disabled this for the second test.

When the testers had finished the final quest in the game there was no more dialogue to be said by the Tavern-keeper and by Belzar, the father. Because of this, when the player

went to talk to them it would freeze the game. To get around this an 'IF' statement was added to the voice programming to stop the game getting into an infinite loop and freezing.

Character movement for the villagers of the game was set by markers created as 'placeholders' so it was easy to see the line of movement of these characters. The placeholders were basic rectangles placed along paths and it was forgotten to take these out before the beta tests. Many testers asked what they were for. During the first test, because of edits to the map, they sometimes walked into buildings or each other and got stuck. To rectify this, their paths were remapped and code was put in place so that they turned around and followed a different path if they collided with anything.

Another problem with interaction was that some of the treasure chests dotted around the map refused to open. The player still gained the item but the animation for the chest did not play. This was fixed for the second beta testing session. Also during the first quest, when your return from the basement back to the tavern, the placeholder that transports you back to the pub was lower than the floor and that made the player fall through the floor.



A treasure chest with no working animation

Style was very important to get right whilst developing the game and was given high marks by the majority of testers.

The team asked the tester to comment on the repetition of buildings and the general consensus was that although they did see that some buildings were repeated they did not affect the overall gameplay or have a negative effect on how they thought the game looked; although they did suggest adding more unique buildings for future demos. In the second test there were many more buildings as there more areas to explore and the

village had increased in size by 60%. Fortunately the repetition of the buildings was still commented on favourably.

Other features that were improved upon between the two tests due to user feedback were floating items and architecture were lowered so that they looked more realistic and the white 'tic-tac' men were changed for realistic looking characters to keep in style of the game. The 'tic-tac' men were set in originally for Michael to program the game before he had been given any proper character models by James or Joe.

For the second beta session, we added the ability to look at a map of the area you are in, testers liked the rustic look of the maps and found them moderately useful, however they pointed out that the maps would be much more useful if they had a marker showing whereabouts the player was on the map. Unfortunately there was not enough time to implement such a feature.

Conclusion

The project was well received based on the feedback from the beta forms and the team considers it to be a success. The aim was to create to a 3D RPG and for it to serve as a valuable addition to each team member's portfolio. The skills used to create the project are relevant to the games industry but could also be linked to other areas of the multimedia industry. The game could be infinitely expanded; with more time and resources. The most important changes would be improved graphics, pacing of the storyline, new animations and new assets.

In the games industry it is common for different team members to have to have multiple skills in different aspects of projects. There were always elements that could be worked on; whenever something was finished there was always another task waiting. Having difficult parts of the project to work on was a good way to break up the workload. Alterations of the games' assets was fairly common throughout, this required a lot of patience for everyone involved. Compromises were often made so that more important assets could be worked on first.

There are plenty of areas of the game that can be improved; and the project allowed each member to motivate each other and give thoughts on what worked well. The

enjoyment of creating content and then seeing it in the final application was also a great motivation tool. The more complete the game, the more encouraged the team became to push themselves.

Games companies will aim to employ one person that is multi-talented. Working in a small group is quite common in the games industry. Creating a role playing game with three people is a good way to experience the sort of deadlines that people working in the game industry have to expect. One of the most important lessons learnt was one of management and organisation.

Important lessons learnt were to respect each of the team's views on design and the story. Allowing Michael to have the final word so discussions progressed and issues were resolved was crucial to the project's success. This was the largest amount of programming he has ever done on one project and he feels vastly improved in this area thanks to the project. James felt that the most important aspect he learnt was being able to create poly buildings efficiently and quickly. He hadn't done this before the project had begun. Joe feels more confident creating almost every part of the human body in Maya dealing with the limitations that a game can have such as poly count and size of textures.

The game was exciting to create and will be a valuable piece of work to demonstrate the skills that each of the team have learnt at University.

Project Management

As Michael had the original idea of creating a 3D game the team decided he should be project manager. He was keen on this idea as he wanted to practice being in a managerial role as this would be a good experience that would help in the workplace. Michael decided that he would play to each team member's strengths so allocated James the job of environmental design, Joe the job of character design and himself the job of programmer, with all other roles spread between them. To even out the amount of work each team member performed targets were set out for each team member as well as a timescale for the work they had to do. If a team member felt they had too much work or they did not understand how to do any work they were encouraged to let the other

members know. Similarly if a team member was thought to not be pulling their weight they would be questioned on why they had not shown any work or why they had fallen behind schedule.

At the halfway point of the project despite regular attempts by Michael, the team had barely seen any work from Joe since his original sketches drawn at the beginning of the project. Joe also seemed to have trouble turning up to meetings on time, sometimes not at all and proved to be very hard to get in contact with when he was late or work had been set and not completed. Fortunately at the end of the project Joe's work was to a high standard and worked well within the game.

However Michael and James worked together very well, Michael's own tasks were self-regulated but he would review any buildings or sketches James made and would often add comments on how to improve the work. They also worked together well on recording dialogue and finding volunteers for the game and cutting the recorded voices up ready to be put into the game.

If the team were to repeat the final year project, they would have a better organisational structure in place; with a more definitive timeline of what work could be expected of each team member. More regular deadlines, but with smaller expectations of what would be produced would have given the team a more structured approach to their project. They would also have perhaps asked for guidance from their supervisor about this at an earlier stage; to prevent teamwork problems occurring later on.

References

Figures for game sales, polls, opinions and definitions

1. Legend of Zelda Sales Figures –

http://vgsales.wikia.com/wiki/The_Legend_of_Zelda

2. Definition of a RPG –

http://en.wikipedia.org/wiki/Role-playing_video_game

3. The week's top-selling games ranked by unit sales –

<http://www.vgchartz.com/weekly.php>

4. The Telegraph - Video Games Section –

<http://www.telegraph.co.uk/technology/video-games/>

5. The history of RPG's –

http://en.wikipedia.org/wiki/History_of_role-playing_video_games

6. Definition of a JRPG –

http://en.wikipedia.org/wiki/Japanese_role-playing_game

7. A guide to JRPG's –

<http://n4g.com/user/blogpost/ultimablackmage/516594>

8. Guide to the 20 best JRPG's -

http://www.gamasutra.com/view/feature/3581/a_japanese_rpg_primer_the_.php

9. A worldwide poll about gamer preferences on RPG's -

<http://www.misterpoll.com/polls/240946/results>

10. Poll on gamers favourite RPG -

<http://uk.gamespot.com/forums/topic/26819868>

11. Forum on best RPG's per generation -

<http://www.escapistmagazine.com/forums/read/9.244513-Name-the-best-JRPG-of-each-generation>

12. Selection of games developed in Unity -

<https://www.digipen.edu/?id=1170>

13. Igneous -

<https://www.digipen.edu/?id=1170&proj=8713>

Unity Examples

14. Hack n Slash -

<http://www.preik.net/unity3d/hacknslash.zip>

15. Project Freedom -

<http://holodeckrose.hku.nl/wybren/freedom/freedom.html>

16. Fantasy World -

http://www.youtube.com/watch?v=svH6_fKULvw&feature=related

17. One Night -

<http://www.youtube.com/watch?v=njUc2gR4ieY>

18. Oblivion Concept -

<http://www.youtube.com/watch?v=l49lhKLSQ&feature=related>

19. Skull Crack -

<http://www.youtube.com/watch?v=KBOYW5V5UtU&feature=related>

What makes the games so good?

20. Index of the greatest games of all time -

<http://uk.gamespot.com/gamespot/features/all/greatestgames/index.html>

21. Final Fantasy 7 Review -

<http://uk.gamespot.com/features/6155700/index.html>

22. Zelda: Ocarina of Time Review -

<http://uk.gamespot.com/gamespot/features/all/greatestgames/p-4.html>

23. Chrono Trigger Review -

<http://uk.gamespot.com/features/6147770/index.html>

24. Final Fantasy 7 on GameSpot -

<http://uk.gamespot.com/ps/rpg/finalfantasy7/index.html?tag=result%3Btitle%3B0>

25. Zelda: Ocarina of Time on GameSpot -

<http://uk.gamespot.com/n64/adventure/legendofzeldaoot/index.html?tag=result%3Btitle%3B1>

26. Chrono Trigger on GameSpot -

<http://uk.gamespot.com/ds/rpg/chronotriggerds/index.html?tag=result%3Btitle%3B0>

Bibliography

James M. Van Verth, *Essential Mathematics for Games and Interactive Applications: A Programmer's Guide*, (2008).

http://www.amazon.co.uk/Essential-Mathematics-Games-Interactive-Applications/dp/0123742978/ref=sr_1_1?ie=UTF8&s=books&qid=1305231980&sr=1-1

Michelle Menard, *Game Development with Unity*, (2011).

http://www.amazon.co.uk/Game-Development-Unity-Michelle-Menard/dp/1435456580/ref=sr_1_1?ie=UTF8&qid=1305232003&sr=8-1

Will Goldstone, *Unity Game Development Essentials*, (2009).

http://www.amazon.co.uk/Unity-Game-Development-Essentials-Goldstone/dp/184719818X/ref=sr_1_4?ie=UTF8&qid=1305232003&sr=8-4

Michael Ingrassia, *Maya for Games – Modelling and Texturing Techniques with Maya and Mudbox*, (2009).

<http://www.amazon.co.uk/Maya-Games-Modeling-Texturing-Techniques/dp/0240810643>

George Brant Bridgman, *Constructive Anatomy*, (1943).

<http://www.archive.org/details/constructiveanat00briduoft>

William Andrew Loomis, *Figure Drawing for All It's Worth*, (1943).

http://www.amazon.co.uk/Figure-Drawing-All-its-Worth/dp/0857680986/ref=sr_1_fkmr0_2?ie=UTF8&qid=1305232388&sr=8-2-fkmr0

Word Count: 11,871

Appendices

Available on request (includes Consent forms, Beta forms, Purchase receipts, etc.)