Process Document

The Process Document contains a chapter from each member of the team, where we've talked about our specific areas during development as well as what else we had on our minds. Of course these don't cover the entire development process, but they should give an idea of some of the phases and problems we've been through.

Contents

Development by Per Juul Ostergaard

Project Management

Project management was taken quite literally, as being a manager of a project, but not manager of a team. The team should be able to lead itself and everyone should be able to influence how the development should be run. This did not quite work out it has become obvious that a team on unknown ground needs someone to take charge to drive the project forward. Especially when everyone is focused on their own role in the team.

Concept Development

Our group had a good start developing our concept. Everyone agreed on the concept of game and was enthusiastic about it. Spending several hours on brainstorming and concept development really paid off during the rest of the development. Unfortunately the team lost some of the touch with the concept during the months of development. This should have been avoided with better documentation which could have kept focus on what we had decided, and how things should be done.

Scrum

Our Scrum development took off well with all the plans laid and documents written. Having a brush-up meeting about how Scrum worked seemed like a good idea, but the value of it seemed poor later on.

When team member were asked about their time estimates for completing their tasks, they often displayed had no qualified guess. This was not surprising in many cases, as most of our tools were still unknown to us at the time. These wildly inaccurate estimates lead our sprints to be inconsequential as we at no point had any real idea on how many tasks we would have completed by the end of the sprint. This decreased the value of the sprints in eyes of the team and a re-launch of using sprints failed at a later stage due to a (natural) lack of understanding what good it would do the team.

We held on to the scrum meetings a bit longer, but as the sprints themselves had been dropped and the team members were focusing on their own tasks, they became obsolete as well. During the next few weeks we tried to establish sending status mails once every couple of days to keep everyone updated, but as there rarely were more than one or two responses each time and the feedback was limited, this idea was also dropped.

What remained of the scrum method was our product backlog. We used it throughout the development to keep track of what everyone should be doing next and adding any new tasks we found necessary. The prioritization in the backlog was not the only thing to consider when handing out which tasks to complete. To keep motivation up team members would often work on tasks that they wanted to work on, though they might not have been as critical as other tasks.

As is mentioned several times in the book Agile Software Development with Scrum, scrum development is normally undertaken by a team of people who know how to solve their tasks. We did not reach this stage until

very late in the development. Part of the basis of a successful scrum team is its ability to be self-organizing. This implies that the team can be given a number of tasks and be expected to figure out who would be best suited to do what and which tasks need extra hands and so forth. This would probably work fine for a homogenous team of programmers, artists, designers, etc. but for a mixed team like ours, it was rarely possible for one member to take on other roles and complete their tasks.

Team

Starting out we had a good solid team. Everyone got the role on the team they wanted, and most of those had previous experience of some kind in their field.

Our programmer did not show up for the first several meetings and classes and was extremely difficult to get hold off. It was not until later that he started appearing and taking part in the development process. Unfortunately he spent all his time getting to know the tools before he disappeared again. This was very unfortunate for the development and the team. We had no other programmers and offers to other teams to trade programming for art was not successful. The team spent quite some time developing concepts, which would no longer be feasible because of the lack of programming expertise.

In the end we had to spend time cutting and rescaling a lot of our content to enable us to move on and try to create a meaningful representation of our game concept. This was pretty bad for the morale of the team. In the end other team members had to step in and try to implement some critical features through scripting, which caused other areas to suffer comparably. But this was an acceptable tradeoff, considering the circumstances.

Communication

As mentioned earlier our team started out by taking a lot of important decisions together, but as development kicked into gear this changed. Towards the end of development, one part of the team was taking all the decisions, while the rest was working on their assigned tasks. This might not necessarily be a bad thing, and certainly made decisions quicker, but it was harder to "share the vision" of the game and valuable input might never have made it into the game. The cause of this is very likely to be the end of our "official" meetings were we each had a chance to talk about what we felt was important about the game, the development, etc. During our work days, each team member was heavily focused on their own work which did not foster a knowledge-sharing environment. This makes it even clearer that we should have held on to our weekly meetings.

Milestones

We had a number of milestones and we missed just about every one of them. We kept being far too optimistic about the speed of developing our game, which, together with our programming troubles, led to even heavier slashing of content. This had the most serious impact on our QA process as each time we took a day longer to reach a milestone, that day was taken from our QA.

QA

Due to the type of game we created, with the focus on story and atmosphere, it was not really possible to playtest the game before the very end. This led to a number of things we did not have time to correct before finishing the game, as well as to little time to balance the levels and polish the atmosphere of the game.

Game Design and Programming by David Nordahl Larsen

Development

We started out with a strong concept, that had pretty much been decided (though not the specifics) before the group was formed. From there we did several brainstorming sessions to decide on the precise game we wanted. The initial concept was to make a survival horror game, where the player would have to fight against different enemies, kind of in the footsteps of Half-Life 2. The initial idea was to have two levels, the village and the castle, with the village being a very interactive place for the player to explore.

We initially didn't have a clear idea of the story, except that it included a mysterious artifact, a village and a castle, with a man who lost his son. It wasn't until we decided on the background for the Blackstone Pendant that the whole story starting taking shape, with the crafting of the pendant, the attack on the village and the kidnapping of the player's son. Something we also discussed in the beginning that regrettably we dropped was the addition of voice acting. With the turn our game took, it would've added a lot to the atmosphere, if done properly. After getting the story settled, we decided on the addition of a third level, which is something that added a lot to the final story told in the slice of the game.

After getting to work with the Unreal Engine we started realizing what our limits were. That combined with our missing programmer made it necessary to cut away a sizeable chunk of content that we had planned (as can be seen in the Design Document). Some of the things we were close to including but had to cut away includes additional elements on the HUD, having more enemies for the player to encounter, the Journal, granting the player a torch to light his way and improved dialogue trees.

Problems

During the development we did encounter some problems, some more serious than others. Our lack of a Programmer required someone else to step into that role. Having prior experience with web pages and some modifications in Neverwinter Night and World of Warcraft I was our best bet, although it wasn't optimal. We were also told it was quite a daunting task for beginner programmer.

The lack of documentation for the specific version of the engine was problematic as well, especially on the programming side. Although it was possible to find tutorials, most of them were outdated. The few months that the new version of the engine had been out weren't sufficient for the fan base to learn and develop enough documentation. That made it very hard to work with the programming beyond the most basic things. On the bright side then a cheer went up every time something worked as we wanted it to. Another annoyance regarding the engine (although it is very easy to use for level building) was the fact that it is built up around multiplayer games, while we were working on a single player level.

Having two talented artists we ended up running into problems on the art front as well. The models for the NPCs and enemies ended up being harder to implement in the engine than we anticipated, so we had to

abandon some of them while reducing the quality of the others. In the end we relied on one of Unreal Tournament's own models for the end boss fight which worked out okay. Not being able to make it work in the way we wanted it in the program code, we ended up having to give the end boss a modern day weapon to simulate him using magic against the player.

One of our most "interesting" problems occurred when we after many attempts as well as failures at increasing the size of the player character. In relation to the Village the character was only about half the size of everything else. It wasn't something we worried about in the beginning as we assumed it would be easy to change the player's height. IN the end we ended up with a working dirty hack. We added an invisible volume around the whole village, raising the player into the air, making him appear higher. It didn't negatively affect gameplay and ended up giving us the desired effect.

Improvements

If we could get one thing for our next game, it would have to be a proficient programmer, or maybe two. It would have been great to been able to add a way for the player to fight as well as avoid combat encounters by the use of stealth, although adding both might enlarge the scope of the project too much.

Another thing to improve would be the process itself. Communications-wise we started out strong while it went down towards the end. Production wise we started out on the light side, although we had many ideas, and it went up towards the end. I'd take a guess and say that it was because of our rising proficiency with the engine as well as a bigger sense of accomplishment as we saw out ideas come to life. But mixing these two to have a middle ground the entire duration would be optimal.

Having tried to work with the editor, it would've been nice to be able to do it all over again, as the result would end up in a better state. In the end it ended up as a bit more of an adventure game, rather than a survival horror one, but our resources taken into account we're pretty pleased about the end product.

Level Design by Niels-Henrik Pesonen

Development Stages

Initially the game was planned to have 2 different stages: The living village level and the castle level. The living village was estimated to need more demand out of the 2 levels as the original plan was to really breathe life into it in the form of NPCs going about doing their daily routines while the player would seek out hits on the whereabouts of his lost son.

As work progressed on the village level we slowly became more and more aware of our programmer not having the heart for this project, so about a month into the project as we had still not seen one line of code the plans for the village level were changed from a working living town, into a more deserted, traumatized town.

When the village level reached a stage where covering the BSP and meshes with materials (1 week overtime from the original planning), the puzzles for the castle level was brought up again for evaluation and a paper prototype of the level was drawn up. After discussing the final positions of the puzzles, we started work on the castle level in much the same fashion as the village level by first digging out the BSP of the level and thereafter implementing the puzzles as well as static meshes/monsters.

During the creation of the village level we had talked a bit about the possibilities of adding a third level to the game: the village level in the land of the dead. This had been pushed back as we were rather concerned at one point with meeting our deadlines without a programmer, but as we pressed on with our work it became apparent that we had the time to create at least a scraped version of the “ghost village”. Originally the ghost village was to be populated by evil ghosts who would attack the player and thus forcing him to seek refuge in the church and thus unfold the story of the Blackstone further, but with the lack of a dedicated programmer and mounting AI issues we decided to leave the ghost village mob less in order to still be able to use the level and thus give the player an idea of how one of the Blackstone functioned by letting the player explore the village freely.

Problems

Apart from the planning issues described in the chapter above, most level design issues were derived from the learning process, examples of which could be issues with the different puzzles not working as intended, lighting shining through BSP and of course parts of the level such as the river disappearing from time to time. Most of these problems were overcome by looking at some of the many tutorials littered across the internet.

If we look at the some of the puzzles, the door gauntlet of the castle level could serve as a good example as when it originally was created the doors simply closed far too slow with the original setup in kismet. This was fixed by changing the delay between each door matinee to instead all start running when the initial trigger is activated.

Another reoccurring issue regarding puzzles was objects such as lights, sounds and so forth would start before being triggered by the puzzle they were a part of. This was repaired with a “go around” fix that would lit all lights when the level starts and then turn off the selected lights so that they wouldn’t be re lit before they were triggered by the puzzle they were attached to.

One problem that occurred on the village level between some builds were one of the more intriguing ones, sometimes when the level was built the river would simply disappear from the level. We never managed to find the reason for this and decided to just leave the river till the final build since recreating it wasn’t a hard task.

Production Tasks

All in all most of the production tasks upheld the time they were assigned. In a few cases we had to extend the timeframe slightly in order to accommodate for changes wanted by the designers, as well as a few issues regarding the puzzles bypassed production times as we as mentioned before were all learning the programs while making the game.

One of the few “larger” things in the game we ended up dropping was the torch that the player was supposed to obtain on the first level to use with some of the puzzles in the castle level. The reason we dropped it was simply that we didn’t have the programmer to do it and for the rest of us to spend time creating it would take away precious time from the other tasks we were all assigned to.

Generally the level design part of the game wasn’t fraught with problems, UnrealEd while being slightly annoying to work with at times (coloring terrain kills wrists) isn’t a difficult editor to work with, and the many tutorials found on the internet were able to answer most questions about building different things in the levels.

Future Improvements

The only thing that might be able to be improved regarding the level design is the planning of the levels, so that some of the extra work in changing things in the level could be avoided. If you were able to place the village and the castle next to each other and compare the amount of extra work on the 2 levels you would see that the castle had far less re building of it as it was far better planned than the village level.

Communication in the team was never really an issue as instant messaging systems made questions and troubleshooting across the different departments easy.

Art by Rositsa Dineva Dineva

Style

To suit the game genre, horror survival, the visual look of the game is chosen to be in dark, medieval style.

The game starts with an introduction that gives the player the pre-story of the game. The introduction is made in the same way as the medieval books– with naive, simple drawings, a lot of decorative elements and big, richly ornate letter in the beginning of the text. The font is Omnia LTStd, which is a medieval inspired font, easy to read on screen.

There are two different stages of the game play. The first one is in the village and the second is in the castle. In order to create an accurate village a research on medieval houses, life style and structure was done. As a result of it, all the houses are built out of wood and stone, they are not very tall and have wooden or lawn roofs. The atmosphere is dark and hostile.

Following the "tradition" of the dark, horror genre a Gothic style is chosen for the castle. The player cannot see the castle from the outside. There are gothic architectural elements for the interior: narrow Gothic windows, furniture and statues. In order to achieve a scarier and more mystical environment in the castle the surrounding is made dark, partly ruined, there are odd looking creatures and puzzle elements that the player has to solve, in order to continue the game.

Dimensions and Point of View

Dimension strongly affects the user's experience and especially the sense of environment that the game provides. Dimension and point of perception make together the game environment rough form.

"Blackstone" is a horror-survival three-dimensional game.

The audiovisual style of "Blackstone" is Photorealism (Gran Stylissimo: The Audiovisual Elements and Styles in Computer and Video Games Järvinen Aki 'June 2002 Computer Games and Digital Cultures Conference Proceedings').

The characters, the environment and the objects in the game are in accordance with the chosen style. To create photorealistic objects is a difficult and time-consuming task. The process of creating the objects was more difficult because of the poor familiarity with the software and its possibilities.

The PoV (point of view) is first person view, and the player cannot change the perception point. All objects in the game are made in accurate size and proportion, so that the experience of the user is more realistic and vivid.

Character Development

All the characters are three-dimensional, with accurate proportion and outlook. The outfit of the characters is also truthful. Their clothes are similar to those of the people in the medieval ages. The clothes are in neutral colors and natural materials, such as leather and wool. Due to the shortage of time and limited potential with the new software the number of characters and figures in the village had to be reduced.

The creatures that the player has to overcome in the castle are imaginary beings, like zombies, undead, etc. They are frightening and creepy, and complete the overall dark and scary feeling in the castle.

Blackstone

The Blackstone is a conglomerate of different elements that the player should collect and put together in order to complete the game. There were different ideas during the development process of the Blackstone.

The first idea was for the Blackstone to be a combination of geometrical objects that together make a 3D object. It was to be a heart shaped stone, whose destruction kills the owner of the castle and completes the game.

For various reasons, such as better aesthetics and recognition, the Blackstone was decided to be pendant alike. The pendant is also made in Gothic style, with interlaced ornaments on the metal surface.

Art by Vorobjova Jekaterina

My field of responsibility is art and visual appearance of the game. As Rositsa and I are the artists in our group, we shared our areas. I was mostly in charge of the visual appearance and modeling of the game's characters and some of the other objects like furniture in the village, library in the castle etc.

We used 3dmax for drawing and modeling is 3dmax. It has strong modeling capabilities, and flexible plug-in architecture. It is widely used by video game developers, TV commercial studios and architectural visualization studios.

It was a completely new program for me and I spent a lot of time exploring tutorials to understand how to make the objects needed for the game. I started with simple objects, such as sofas or chairs, using standard primitives. This is a basic method, when modeling is performed by using simple items, such as boxes, spheres, cones, cylinders and other predefined objects. After this, they can be polished with "Modify" function. These were my first steps with this complicated, but exciting software.

The list objects needed for the game has been decided together with all members of the group. There was a very successful brainstorming session in the end of which, we came up with a list of objects and characters, which has been modified in process either by adding or deleting entries.

Apart from level modeling, one of the most important aspects was characters. Modeling characters is a huge and entertaining, although time consuming process. But as we had to be completed it in a short time, we did our best helping each other. We started by creating the body of the characters. Then skin was applied.

The next development stage, and a whole different story is texture making. To make textures and materials for all the objects and characters of the game, we used 2D graphics - mostly Adobe Photoshop. We had previous experience with this program and drawing in 2D, so there were no problems creating the textures we wanted. There is a function in 3dmax of texture application on the object, which our teacher's assistants have shown us. To apply the skin on the character is much more difficult task. But we've done this as well, although it took us a lot of time to complete. There was a problem to "unwrap" the human body (particularly face), and even teacher's assistant could not help us in solving this problem. In the end, we found another way to make it properly, and as you can see, the result is embodied in three human characters in the village level of the game.

The first character we created was Innkeeper in the Village. He is the first person you meet there. And since we are in the medieval, our characters were created accordingly. We searched for the inspiration in exploring the life, culture and traditions of those times.

The people in game have custom made clothing and accessories. One of the most difficult parts of character creation is the clothing. You cannot just draw it the way you draw a chair or a cup. In order to create clothes, we had to become real tailors ourselves.

The reason is that to produce the elements of clothing in 3D max, it is necessary to create a dress-pattern first, and then, connect seams together in a particular sequence (just like in the real tailoring). It took us a long time to make the right patterns and size of clothes, because we were not particularly experienced tailors. But, nevertheless, we managed to succeed. The most problematic part of the cloth creation, was making a hat and a hood because of its difference from the usual parts of clothing. One major difference is the material, the other - location and binding. First, it could not be attached to the head at all, but rather flew away into nowhere. But after multiple attempts, we succeed in this as well. Where we did not succeed was shoemaking for the same reason as with the hat.

Other two characters in the village are Blacksmith and Caretaker. Technically, it was easier to create them because we were already familiar with the process step by step. The rest was our inspiration and imagination. Nevertheless, we still ran into some difficulties applying skin to the faces because the texture we wanted went in a wrong way when we tried to put it on in 3D max. Attaching skin to the face in “Unreal Tournament 3 Editor” went even worse. After several fruitless attempts we decided to leave the idea of applying those textures and drew new ones.

Speaking about the hostile characters – we have two kinds: zombies and ghosts. It was not very difficult to create a zombie. It was rather simple to model the body and it has almost no clothes on. We used the texture drawn in the Adobe Photoshop (bloody bones) to skin him. (Unfortunately in the final version of the game we decided not to use zombies) We’ve had difficulties creating a ghost, since it was not supposed to have a body (just a skull), but we still needed to create some kind of clothing to wrap around the nonexistent body. The problem was to attach the clothes exactly to the skull. We did this, but for some technical reason, we could not open the saved file later on. In the end, we’ve found another option to create ghost with clothing by combining skull and human body with clothing on.

In general, during the producing process we had to cut a lot of initial ideas. The reason for that was overestimation of our strengths, as well as loss of our only programmer. For some reason, he has not been attending classes and meetings, ignored our calls and emails. The part which suffered the most was the animation of the characters, AI of the hostiles (attacking the protagonist in the castle level). We also had to simplify the dialog system.

As to suggestions for the future – what could be improved?

- The group MUST have at least 2 programmers.
- More extensive work on character animation.
- Put more NPCs and decoration to the levels (more friendly characters in the village, animals such as cats, dogs, horses, donkeys etc. that every village has; more hostiles and addition of the dialog system in the “castle” level, create the boss with appropriate AI, different hostiles with different behaviors, more puzzles.
- Rigorous testing during the different stages of game production.