



Første del – til offentliggørelse på DADIUs hjemmeside.

Introduction

In Flipside the player takes on the identity of a mental patient with extreme mood changes who is planning his escape from the insane asylum in which he is being held. The imaginary escape through the landscape surrounding the insane asylum will bring him to safety on the ship from the people who want to take him back to the asylum. The escape takes place in a world put together from cardboard pieces and jumping jacks decorated with magazine cuttings, stickers and other. Those objects come alive in his mind as he fantasizes about the obstacles he will face and how to defeat them. As the main character is a hospitalized mental patient nothing in the game is “normal” – every experience is an extreme and caricatured perspective on what a “real” experience could have been like.

After only a few brainstorm sessions the core concept specified a platform game in a universe consisting of flat 2D objects placed in 3D space. Using a flat universe blended nicely together with the idea of working on a main character that had two sides to his personality. On one side he would be super happy, almost deliriously so, whereas on the other he would be angry and boundary frantic. Allowing the player to flip between those two states of the game world has remained the backbone of the concept throughout the development process.

During the concept development it became clear the game would target children aged 6-12. Testing a late prototype of the game with this target audience turned out to be very interesting, as the subjects thought the caricature was new and interesting compared to the orcs, racing cars and spaceships they are used to.

What went well and what did not

One of the main challenges in this project has been to define the game concept. From the beginning we had a core idea from which we developed the game concept. The time needed for developing the core idea into a concept took more than the pre-production.

Spending long time developing a concept is not all bad. First and foremost it gave us the time needed to develop a concept and fine-tune it as much as possible and thereby creating a concept we all like and could be proud of. By not choosing the first and most obvious concept and trying to develop a solid concept, we have had an open project with the possibility of changing the concept and artefacts accordingly to new ideas and changes to the concept. Being open have created a pleasant environment where the team have had the opportunity to come with suggestion to artefacts and the concept

Because of the long time needed for developing the concept we have sometimes felt we were under pressure to show progress in the concept. This pressure might have had the effect that we pushed ahead without thinking of the consequences. Often it resulted in trying to develop the concept without reflecting over its usefulness. The project had more focus on performance than reflection which sometimes needed for quick decision making sometimes resulted in features being added on the fly and sometimes thrown out to the team before the needed paper sketches and reflection about the use were in place. This made the team feel they sometimes worked with tasks would not be part of the final project making them frustrated and feel like they wasted their time.



Conclusion

Did we fail because we spent much time in developing a concept? Surely the final product suggests otherwise. We might started out slowly but it also forced us early on to really think about what we wanted and the time we spent only made our concept that much better. Of course things could have been different, but in reality we should be proud of the end product and the few problems we encountered. From our perspective the long time needed for developing the concept most of all helped us realise what we needed and what we wanted. Asking today if we would change things, we probably would but we are not sure it would make a better concept or even a better process.



Post Mortem – Flipside

By

Tim Nielsen, Producer

Thomas Pilgaard, Game designer

Jan Rahbek, Director

The product:

In Flipside the player takes on the identity of a mental patient with extreme mood changes who is planning his escape from the insane asylum in which he is being held. The imaginary escape through the landscape surrounding the insane asylum will bring him to safety on the ship from the people who want to take him back to the asylum. The escape takes place in a world put together from cardboard pieces and jumping jacks decorated with magazine cuttings, stickers and other. Those objects come alive in his mind as he fantasizes about the obstacles he will face and how to defeat them. While jumping in trees and climbing cliffs to reach the ship the main character has to flip between his two different states of mind to avoid cute bunnies and honey bees or defeat their counterparts; evil nurses and stinging wasps. Flipside is an action platform game that takes place in a 3D universe made from flat objects. Throughout the game the player will be able to flip the game world between two modes in which different abilities must be employed to overcome the obstacles faced by the player. As the main character is a hospitalized mental patient nothing in the game is “normal” – every experience is an extreme and caricatured perspective on what a “real” experience could have been like.

The idea:

The development of the core concept for Flipside was begun 1-2 weeks before the actual production was initiated. At first the game director and game designer came up with some of the basic concepts with the project manager keeping track of what was going on. Shortly before the production began the lead visual designer was brought in to discuss the visual universe. From then on the remaining leads and eventually the whole team were more or less involved in the process of developing the concept by making comments and suggestions.

Early on in the development process it was fairly clear that the game would explore ideas different from what is commonly seen in commercially viable games. This affected what sources of inspiration could be used and had a major impact on all parts of the finished game.

After a few brainstorm sessions the core concept specified a platform game in a universe consisting of flat 2D objects placed in 3D space. Using a flat universe blended nicely together with the idea of working on a main character that had two sides to his personality. On one side he would be super happy, almost deliriously so, whereas on the other he would be angry and boundary frantic.

Allowing the player to flip between those two states of the game world has remained the backbone of the concept throughout the development process. Initially the game was set to let the player explore the inside of an insane asylum as well as an outdoor setting during the escape. But it soon became clear that we did not have resources to produce two distinctive environments. The outdoor environment was chosen as it had the most potential to use the 2D objects in the game world to create the vistas and impression of depth that had been envisioned.



From the beginning of the concept development it was clear that the game would not be serious or suggest realism in any way. Instead all energy was aimed at creating a caricature that would be enjoyable to our target audience.

Target group:

At first the target group was estimated based on the designers' nostalgic feelings towards the platform genre as well as thoughts about a main character with a very violent and somewhat perverted personality. As such adult people were targeted at first. As the concept developed further it became clear that the game would do much better if targeted at children aged 6-12. Testing late prototypes of the game with this target group turned out to be very interesting. The children taking part in the test turned out to be more than capable of understanding the caricature and thought it was a pretty funny idea compared to the orcs, racing cars and spaceships they are used to.

All textual interfaces in the game were produced in English to ensure that the game will be accessible to as many people as possible. Contrary to this all speaks were recorded in Danish. This was done to emphasize the origin of the game. Fortunately an opportunity came up to hire an extremely talented voice actor whose passion and commitment has made the main character come alive. We are confident that this will make him come alive to people who do not understand Danish as well.

What went right, what went wrong

Developing a game concept is not something that just happens, it requires creativity and ideas. From the start we set out with a core idea to make a flat world with two sides. The player should be able to switch sides as it suits her. Going from here to a fully developed concept that would be developed into a game has maybe been the single most difficult and influential task throughout the entire project.

The first week of the project was spent in the search of a viable concept - we developed more than one. But no one really seemed to capture the essence of the core idea. Hours and days were spent and at the end of the weekend of the first week we had a concept ready for implementation although it can be thought of as unproven as we did not do any early play testing like paper prototyping.

Because of the long time needed for developing the concept we have sometimes felt we were under pressure to show progress in the concept. This pressure might have had the effect that we pushed ahead without thinking of the consequences. Often it resulted in trying to develop the concept or concept related artifacts without reflecting over these and their usefulness. The project had more focus on performance than reflection (Thomas) which sometimes forced people to work differently than they normally would do. Breaking peoples work pattern might make them uncertain in their abilities and accidentally have had an impact on their work. Because of the pressure in developing the concept the director and game designer felt they had more pressuring matter to attend to than actually communicate decisions made regarding the concept (Thomas) or answer related questions. Leaving the team in suspension may have had a negative impact on people leaving them in an unsteady state of mind regarding the game. The need for quick decision making sometimes resulted in features being added on the fly and sometimes thrown out to people before the needed paper sketches and reflection about the use in context of a level were in place (programmers). This made



the team feel they sometimes worked with tasks and assignments which would not be part of the final project making them frustrated and feel like they wasted their time (programmers).

The long time needed for developing a concept also took its tolls on the plans for the project. It is difficult to plan the unknown and therefore the plans were not developed in full until after a viable concept was in place. Planning requires detailed knowledge of both the artifacts in question but also the people performing the work and their expert knowledge. In this case the concept was rather fluent up till late in the project. This calls for some serious planning sometimes requiring people to plan and estimate in more than one way to acquire a sensible and sustainable plan. Unfortunately this sometimes made people feel that some of the estimating was a waste because it wasn't directly reflected by the plans (programmers). The highly detailed planning also had another problem one department would have some features as top priority while another department would have others (Audio).

But the fluent concept also made it difficult to plan and afterwards to follow up, because things may have changed, making people feel that there were not any follow ups or less than expected (Thomas). Because of the state of the concept it also required a lot of work to plan non-functional requirements of the game, like ingame menu, mostly because we did not really know what to put there or if it would be changed. Basically we prioritized functionality before aesthetics. But this was frustrating for the programmers because they needed this information (programmers), especially because we in the end of the project actually had the time to develop most of the non-functionalities.

The pre-production or the time needed for developing the concept took too long (Audio) or simply we did not manage to produce the sufficient artifacts in the time we spend. The result was that it took too long to be precise in what was needed for this game and we could not develop a level until rather late in the project.

Spending long time developing a concept is not all bad. First and foremost it gave us the time needed to develop a concept and fine-tune it as much as possible and thereby creating a concept we all like and could be proud of (programmers, artists, audio). In spending the time needed we did not limit ourselves to the main purpose of Source but tried to think outside the box, and we think we have come up with a great and fun concept which has been challenging for most of us. By not choosing the first and most obvious concept and trying to develop a solid concept, we have had an open project with the possibility of changing the concept and artefacts accordingly to new ideas and changes to the concept. Being open have not only created a pleasant environment where people have had the opportunity to come with suggestion to artefacts and the concept, but it have also helped the creative forces of the project to keep coming with ideas (Thomas). By giving us the opportunity to come with suggestions to the concept and artefacts we have potentially opened up for large discussions, but we have chosen to let each department have the last say in any discussion, thereby giving them power over their own subjects but still being open for new ideas. This have helped the departments not feeling they have had to come up with everything but they have a team to back them up, while everyone have had the feeling of opportunity to influence every aspect of the game. This has worked with great success in this project (programmers, artists, audio).



Because of the lengthy time spent on the concept, few details or loose details were given to each department. This has given the departments the opportunity to explore the opportunities of each aspect of the concept without limiting ourselves (Audio).

The concept developed could in many ways be considered bigger than what most teams would be able to implement in a month. To counter this problem we did extensive planning. One thing was to cut the concept into use cases and detailed information for each use case (Jan). By producing use cases we had the opportunity to prioritize the functionality of the game and the information needed to set up pipelines, delegate assignments and estimating the needed resources for the game to be completed. By dividing the concept into use cases also showed that we did not have all the time in the world to complete everything and we chose to split the use cases into Need, Want and Nice. Need is what had to be in the game, Want is what we liked in the game and Nice, well what it would be nice to have. This gave us a very strong way of prioritizing because each use case had an estimated number of days it took to finish and we knew how many days we had left. By doing this we really did cut into the core of the game and succeeded in prioritizing the most important parts of the game (programmers).

Because of a loosely defined concept we also had to come up with a way to communicate. One way we went was to use a small meeting each day, where we shortly would describe what we had done and what we would do for the rest of the day. We also had the opportunity to ask questions related to other departments and discuss a few subjects. These meetings gave us an overview over the entire project helping us to understand the current state of the project and where we were headed (programmers, Thomas)

Even though the long time needed for concept development worked out well in many ways, does not mean it can not be better. One of the things we were not very good at in the start of the project was to create visuals of the concept, making it difficult to talk about the concept. By producing visuals earlier we may have been able to create a better communication surrounding the concept (Jan). Furthermore we were very focused on the universe making it very difficult to come up with a suitable gameplay and ending up giving ourselves a hard time (Artists). It is worth mentioning that not everyone feels this way and some felt it was a strength, that we had a universe to centre our development around. Another way to specify the concept in more details early on in the project might have been to play around more with the concept by producing paper prototypes, visuals and trying to write down rules. Instead we tried to develop the entire concept in written form.

Not all centres around the concept but also around the people involved in the project. Dadiu had placed us at the Danish Design School, where we had one room for the entire production team. This could have caused us problems, but because everyone is open and friendly it helped in creating a great environment where everyone was welcome. Even the Audio designers had a space in the same room, sometimes making noise, but it helped us all understand their work and the rest had an opportunity to hear the sound made (Audio).

A tool which has shown to be most useful is SVN. Not only did it help versioning our files but we also had the opportunity to have a compiled version of our game on SVN the entire time. This meant that everyone on the project team at any given time could download the current state of the game and try it out, and come with feedback. By having this ability open to everyone we had a lot of internal testing, more important we had a feeling of progress in our development and we could see it. This really helped in keeping our spirit high at all times.



Animators first came into the production after a week making them not feel entirely part of the team even though we tried. We considerer this a poor decision because animators have knowledge about character and story development, which could have been put to good use the first week. Instead the animators left outside any decision making because most of it, is in place at their arrival.

Methodology

When managing a project many different methods and tools are available. In order to finish the product on time it is my belief that the project manager must take in to consideration what kind of project he is working on. The most suitable tools for the task should be determined according to this. The project manager might also use methods and tools known to her from previous projects, but doing so by default may result in disaster.

First and foremost we as a team have no or very little knowledge regarding Source and producing games with the tools available as part of Source. This of course makes it very difficult to do estimation and risk assessment for the game development process. The method used should reflect this uncertainty and try to counter any problems. Unified Process (UP) is a development method for developing large it systems over a long period of time. UP contain a vast number of tools for most imaginable cases. But what makes UP really interesting is that it is iterative and risk-driven at its core. This essentially means UP is a number of small projects developing small parts of the overall systems beginning with the part that has highest risk of failure. Risk in this case is not related to project risks but implementation and system risks. For this particular project this means that our concept will be divided into use cases which will then be ranked by risk. Risk in our project is considered from a combination of game relevance and implementation difficulty. Game relevance meaning how important we expect it to be to the finished game, and implementation is the expected difficulty implementing the use case.

Another method which was considered to be of interest to this project is Xtreme Programming (XP). One of the most important aspects of XP is the close relationship with the customer and the developer. The customer's direct influence on the project is important to the development process. To ensure that the customer have insight to the progress of the project another part of XP is frequent small releases throughout the project. In our project we were frequently visited by a consultation group that could be considered our customer to some extent because they would provide feedback on the current state of the project. This required a visual impression of the game and its state which is part of the reason why we made small releases each day. The advantages of this approach is that the every small release is available for the team to test, use and provide feedback while making it much easier to get an impression of what progress has been made by the team as a whole.

Because we lacked knowledge of Source we used prototyping for putting working models together quickly. In addition to that we released a prototype each week with all the features and functionality produced up until then put together, even if parts of it were not working perfectly. This was done to get a general idea of what the game would be like and for use in internal play testing.

Knowing whether our game is fun and playable is no easy task which is why I planned several tests sessions. The first session was tested on a prototype only just holding together (it crashed every time the player died). It was aimed at gameplay testing and studying whether the players thought the



game was any fun. The second session was with a game that only needed polish and balancing. This time around we let 8 children aged of 10 – 12 test the game to get some feedback from our target audience. As hoped we found out what they thought of the game and got some clues to how it could be balanced further.

The development had been scheduled to last one month with no possibility for an extension. Due to this limitation we needed to prioritize all use cases. In order to do this we used a homemade system called Need/Want/Nice (NWN). The Need category represented use cases that was essential to the game, Want represented use cases that was thought to improve the game significantly and Nice represented use cases we would like to implement, but had the least impact on the realization of the game concept. The basic idea was stolen from another project manager, but what I did was to estimate each use case and then place them into NWN. We had a certain number of days until the game had to be finished which reflected the number of day available in the Need category. After prioritizing the use cases it was just a question of making the puzzle come together as planned.

Methods are a way of doing things, but they are no more than thoughts if not put down on paper. To do this a series of tools was applied. Developing an overall plan might be a simple task but a detailed plan is a different story. In our case the plans were developed with help of the leads as they possess expert knowledge in their area of expertise. The plan was entered into MS Project which is a tool that helps project managers keep track of their project. I mostly used a Gantt chart as I don't see any reason for micro management, and given the leads from each departments deadlines should enable them to plan the work themselves, effectively enabling me to retain the large picture.

Furthermore MS Excel was used to easily generate prioritized lists. SVN was used for easy version control as well as storage for all releases of the game. Walls in our room were used for visualizing plans, NWN and important information in an effort to enable all team members to get an overview. Not all succeed as planned, but this is probably related to the fact that many walls were not accessible and therefore difficult for the team to get close enough to get a good look.

From the beginning of the project the game designer and director co-operated in developing the core idea. From here on the development took on a more open approach, letting the whole team provide ideas and suggestions for improvement. For most part of the project we worked this way. Everyone could suggest improvements but in the end it was the lead of the particular area who made the final decision. This worked very well in the sense that everybody felt more like they had more of an influence on the game and probably helped keeping us focus on the project. Only on rare occasions did I have to cut off a discussion because the team knew the leads had the final say.

From the moment tasks and ideas were identified and if they were small changes the lead was responsible for documentation of the changes. If larger changes were identified I was consulted. I would look at the time and resources needed for the task and if necessary the task would be prioritized with respect other known tasks. This worked well because people felt empowered and I did not always have to manage every aspect of the project, but it also made it very difficult at times to know what people were working on and how the completion of each task progressed.

Communication about decisions was often verbal either directly to the person who needed the information or in daily meetings where we shared information regarding the work we were doing and new decision. Large decision like in-game functionality was written down on paper, but much



information was kept verbal. The daily meeting worked rather well and kept people up to speed. The documents on the other hand did not seem to be read by more than a few people. Even though people have been good at communicating verbally, it still remains one of the few things I would really like to change in the future.

If I have to point to things that did not work, I would like to emphasize that I have tried several methods of documentation to get the work finished on time. At first I tried to introduce a week plan for each department but only few people actually used it. Also I did not have enough accessible wall surface to put them up. Secondly I tried to put up a more general plan on the wall (the Gantt diagram was too complex), but that also failed. Either the information wasn't detailed enough or the wall that was available was too difficult to get to for people to bother.

Conclusion

Did we fail because we spent much time in developing a concept? Surely the final product suggests otherwise. We might started out slowly but it also forced us early on to really think about what we wanted and the time we spent only made our concept that much better. Of course things could have been different, but in reality we should be proud of the end product and the few problems we encountered. From our perspective the long time needed for developing the concept most of all helped us realise what we needed and what we wanted. Asking today if we would change things, we probably would but we are not sure it would make a better concept or even a better process.

With the problems (minor) we had everyone still kept the spirit high and we succeeded in producing a game we all thinks is worth the effort and sure is something we can be proud of.



DET DANSKE AKADEMI FOR DIGITAL INTERAKTIV UNDERHOLDNING

Cadle, James & Donald Yeates (2004), "**Project management for information systems**", fourth edition, Prentice Hall
Sommerville, Ian (2004), "**Software engineering**", seventh edition
Larman, Craig (2002), "**Applying UML and patterns**", Prentice Hall

<http://www.extremeprogramming.org/>