



Post Mortem

Introduction

This is a post-mortem for "Sheep!", a game created by a group of three students - dubbing themselves "holzkuh" - during the course "Game Programming" at Technische Universität Berlin, Germany, in Summer Term 2008. The idea of the course was to initially create multiple prototypes in a process of rapid prototyping, whereas at least one prototype was initially requested from each participant - and to finalize one of the prototypes to a full game within the last 7 weeks of the course.

This game had to be able to be executed on an Xbox360, though it also works on Windows computers.

The prototype we chose was "Sheep!", a 2D-sprite-based, rather experimental game.

The basic idea of "Sheep!" is that you need to prevent sheep from fleeing into their freedom, by controlling two separate characters - dog and shepherd - at once, one with each control-stick and respective shoulder triggers.

It should also be possible for beginners to win without crushing their brain upon controlling both characters at once, but to break the highscore you will need to master the control of both characters and minimize fence material and time you need to fence in all your sheep.

What went right



Rapid Prototyping

Before we choose to make "Sheep!", we created seven prototypes, which allowed us to decide which of our ideas were best for further development.

"Sheep!" was a good choice, from various points of view.

The unusual game concept, as well as the funny setting united several of our radically different expectations for the game, increasing motivation in some parts of the group.



Coding

Overall one can say that we didn't really code very clean.

At the beginning of development, we build up a Model-View-Controller-System, and stuck to it throughout the course (more or less).



Still, it allowed us, despite our unclean coding to work parallel without having the difficulty of unnecessary commenting or code clean-ups.



Artwork

We'd say we fulfilled our goal to reach a unified art style pretty well.

The world of "Sheep!" is depicted in a humorous way, yet it is recognizable what's going on.

This strongly influenced the 'fun' various testers experienced.

This became visible especially after a massive graphic-update for the game's backgrounds for Iteration 2 - a motivational boost for the entire team.



Game Testers

Us programming this game led to many of our family and friends getting interested in videogames, and applying for testing.

Even though opinions differed, it was a great help for balancing the level's difficulty, as we ourselves were no longer able to objectively judge it due to our own excessive testing sessions.

The game was supposed to be beginner-friendly after all.



Telecommunication

Efficient teamwork, especially in game programming, requires a lot of communication.

During development, we really got to appreciate the use of Skype. It allowed us to be all on our own, when we needed time for ourselves, yet, connected all actively working team members.

We were also able to help out each other on a visual basis, as we had access to an excellent remote tool, which enabled us even to react to problems and errors occurring only on one of our computers.

What went wrong



Sub-optimal role assignment

When we started this project we were asked in which part of game-programming we were into most. According to this we were shuffled to groups which fit best to our personal preference. However, some of our team members now realized that they were much more into other roles in the team, and found their ambitions and ideas wasted, due to our lack of time. Of course this resulted in an immense demotivation.



Uneven Workload

When we started this course, we were told that this project was going to be a massive.

But we had NO idea. The actual workload was simply -beyond- our imagination.

From our undergraduate studies we had not enough experience in long-term-programming projects with self-determined time management. We absolutely underestimated the process of debugging and finalization, which led to a waste of valuable time at the beginning of production.

While in the beginning, we worked roughly 20 hours per week on the course, this changed radically up to 60 hours and more at the end of the term!





Developing a Game ain't easy

Programming a game in a single term requires an extreme ability to withstand stress and pressure, going FAR beyond the boundaries of a semester. Good experience in the required field (especially in software-programming) present an enormous help, as they extremely increase your working speed.

Without experience from previous projects it is ...not adviseable to try programming a full game.



Time Shortage

Due to the limitations of our game being our semester-project, a lot of good ideas didn't make it into the final game. So didn't the balancing of the separate game-aspects (e.g. the highscore).

This might be the result of our own disability to estimate the optimal aim-complexity of our game, but seven weeks are a really short time to design, implement and test a single player game in three iterations.



Communicate with your Group

It will be practically impossible to develop a game without taking a few timeouts, a few days off for regeneration, once in a while.

However it still might make sense to inform the group about this. Even if your doing nothing, this information may be worth gold. For the group it can be extremely demotivating not to know if everyone's working on the project.

Conclusion

The „Final“ of Sheep! is a nice playable game, with a tutorial that explains the game's rules. Some things are really not balanced. Especially the levels and the scoring system would have needed much more time – time we needed to get the base mechanics working.

Making this game caused very different reactions inside our group. Some of our members hold, that this was mere torture, while others think that it was the best experience they made at the university. In the end we can say that we are proud of the game we made, and that we all learned a lot - trying not to freak out at all times.

Specifications

Name: Sheep!

Developer: holzkuh, Technische Universität Berlin, Germany

Number of developers: 3 students

Team members: Carlo Fürst, Benjamin Janeck, Helen Perkunder

Time of development: 7 weeks

Release date: 9 July 2008

Platform: Windows PC, XBOX 360

Used software: MS XNA, Visual Studio 2005 and XACT, Flash, Fireworks

Used hardware: Low-end to High-end PCs, XBOX 360 Conrollers, XBOX 360, Graphics tablet

