SwitchBoy Post-Mortem

Project Overview

For information on the game please refer to the design document. For this document the current chapter will contain more insight into the team and project structure.

The team was comprised of 3 people, which are in different stages in their computer science education. But none of them had previous experience in computer game development nor in the design of computer games.

A hierarchy within the team was not defined. But each team member had an area of responsibility, which was based on predefined components of the game:

- Marcel Nagele: HUD, Sound, Music, GameState
- Manuel Warum: Level, Level objects, PowerUps, Camera
- Gerhard Moser: Actor/Player, CollisionDetection, Controls, Speed/ScoreTracking

Each team member worked on his components (in a self contained manner) and took care of any possible migration issues. All the components and content (images, fonts, sound) were encapsulated in a Visual Studio (VS) Project. This VS Project was exchanged and updated when necessary via email.

Several project meetings were held periodically to make important decisions, create or adapt the project schedule and track project progress. Some minor decisions were sometimes also made via email.

What went right

In this chapter we will talk about which decisions, we made, had a positive effect on the project and why.

Decision 1

We decided to use XNA 3.0 and Visual Studio for development. This Framework and IDE offers a very good way of modularizing the game into components and also ways to let the game components communicate through a service structure. Furthermore it already has a built-in game-loop that is automatically calling each registered component to reach a default frame rate. Besides that XNA also incorporates for example Vector classes for positioning and movement of game objects.

Decision 2

From the start we created a class diagram with all game components the game would be comprised of. That allowed each of us not only to get an idea of how each game component was connected but also already find missing components very early before implementation started (e.g.: we almost forgot about the collision detection component). And with this diagram and the known components we were able to create the final packages each of us would have to tackle.

What went wrong

Now that we covered the positive experiences made, we'll also disclose the bad things we did. Murphy's Law always strikes without mercy.

Mistake 1
We didn't plan the project well enough concerning scheduling. That means, there were no strict deadlines or real milestones set (or they were not kept if defined). This resulted in delays of the delivery of several components, which sometimes slowed down the development of another connected component as well.

Mistake 2
Changing method signatures and not adding comments is a good way to confuse team members. Most parts of the code were lacking proper documentation, but at least some things were self-explanatory enough to not cause bigger problems.

Mistake 3
Planning to employ the endless powers of pixel and vertex shaders is always a good thing - or it would have been, if anyone knew DirectX's HLSL (High-Level Shader Language). A lot of wasted hours later the game development went on, but without the power of shaders.

Conclusion & Closing

Gerhard: "Personally, I loved to design and implement my first game ever. I already wanted to for quite some time but never got to it. I'm a bit disappointed that we couldn't get all the features of the initial game design implemented though, because it would have been a great experience for the player. But I'm also very happy we finally got a good game on the way anyhow! It was a nice experience, and it already showed how stressful the days close to a game's release can be. On top of it all, it's wonderful to see the player character get moving and come to life as well as all the rest of the game environment around it to make it a complete package."
Manuel: "FYI I am a SPY."
Marcel: Well, the project was easier to implement with XNA than I thought and much easier than it would have been with Java. End of the story: It was a nice project and we also had our fun moments ;-).

Project brief
Programming language: C#, ~~HLSL~~, LevelDescriptionLanguage (LDL by MWar)
Development Tools: Microsoft Visual Studio, Paint .NET, GIMP, Notepad
Frameworks: Microsoft .NET 3.5, Microsoft XNA 3.0
Resources: 3 developers, approx. 7.1415926 man days, and very long and manly nights
Lines Of Code: ~2000
Classes: 23
Nerves lost: *infinity*+1
Bricks broken: 1k+
Switchboys died: 100+