Chromatophobia

Postmortem

● Game Developers:

Thomas Pairitsch (Engine, Pixel Shaders)
David Rainer (Gameplay)
Florian Winkler (Java to XNA Port, Artwork, Sound, Game Menu)

● Project Overview

Our idea was to deliver a Space Shooter that causes horrific eye-bleed, leaving the player stunned. To reach that goal we wanted to give the player an information overload, in form of MANY enemies and flying objects. We knew we should make the game as addictive as possible, otherwise he would stop playing too soon for any mental or physical side effects. The game takes place in outer space in a cage, where enemies are spawning rapidly. To survive one has to collect health upgrades and stronger weapons. The secondary weapons can eradicate large groups of enemies. The black hole created by the gravi-gun causes distortions in the space-time continuum and will suck in every unlucky enemy ship in it's event horizon. The rockets from the rocket-launcher will hunt the nearest enemy and change targets if the enemy has perished already.

We chose XNA as our programming environment, whilst our prototype was written in Java. With our prototype we tried to shove in hundreds of enemies and projectiles and reached a hardware limit.

● What went right

At first one member of us planned the architecture. After that we were able to split up the work without major difficulties.

Porting from Java was relatively easy. We didn't use the C# codestyle consequently, but it went right.

We found XNA to be capable of everything we needed and relatively easy to program. We found it quite astounding, that there weren't any bugs that could not be resolved in a minute.

Feature Cut:

At first we had a much bigger plan for our game. We wanted to make a universe where you can fly to planets and complete a mission there. If we had followed our plans we wouldn't have finished in time. After a few weeks we sat together and decided what to do next and

how to cut the features and deliver a satisfying game that could be completed in time. This could be considered as one of the best decision. For future projects we should always make a feasibility study.

● What went wrong

We had a gaming console planned where we would spam funny messages after hitting the meteor or after killing an enemy. The gaming console was in our design document, but our group member Rudi Mentär (name changed) was too lazy to implement it.
We missed a versioning system. We managed to shove our code together by sending e-mails, it worked out of the box, but it was annoying.

● Conclusion & Closing

The course was fun and we learned much from it. By using pixel shaders we had to do programming close to the hardware and the outcome was quite satisfying. We could say that we learned much from the software development process and we broadened our programming skills following the game loop and delivering visual simulations with physics. What we would like to advise is, that there should be a SVN account provided to every student. Sadly the firewall of the university is blocking student hostels from building their own SVN.

Our Dev Tools:

Visual Studio with XNA
Photoshop, Gimp
Adobe Audition