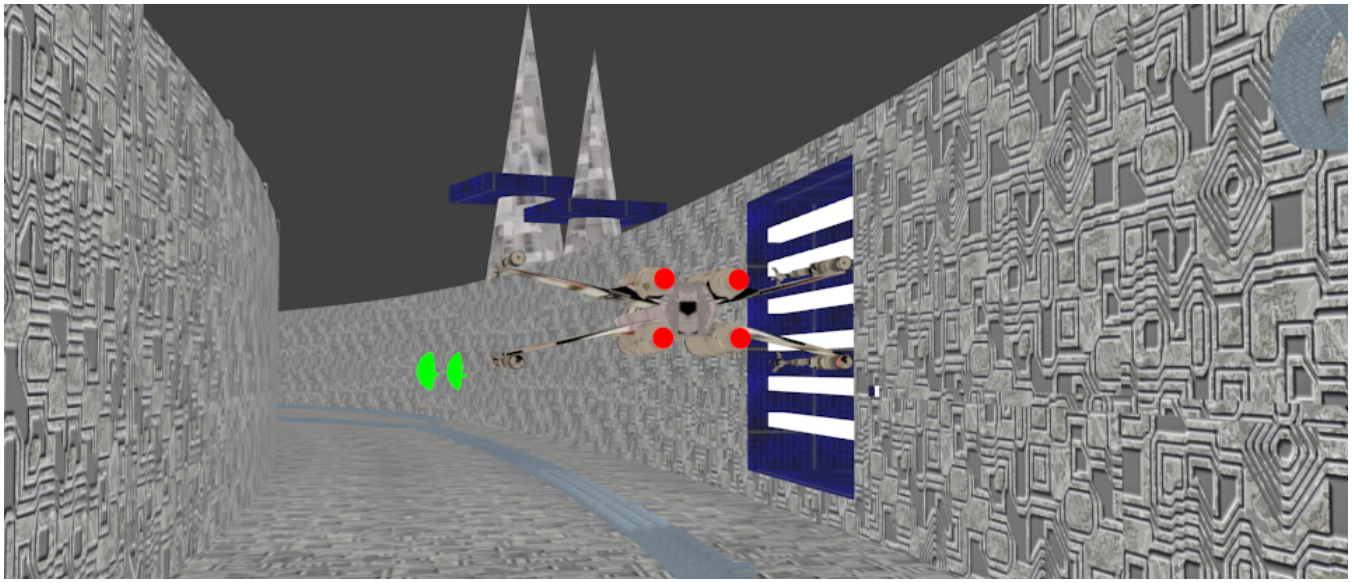
Developing a Jump'N'Fly Game

Results of a practical course at the Chair for Computer Graphics and Multimedia
(RWTH Aachen University, Germany)

Oliver Ney Phillip Kessels Cedric Fokam Fekam Lucian Poth



Abstract

Our main idea of the game is based on a scene from the Star Wars movie where the Death Star has to be destroyed. While we had to create a Jump'N'Fly Game we thought that this might give us a nice content to work with. So the goal of the game is to fly as far as possible through a canyon and to avoid obstacles and laser cannons. To make it easier through the tunnel it is also possible to pick up some powerups which might give you some health back, a shield or ammunition.
The following paper will contain our workflow from the beginning of the project until the final result.

Keywords: game programming, jumpnfly game

1 General Information

After the first meeting we had many ideas that we wanted to implement into the game but after a short period we reduced them to the following main points. We wanted to have four different kind of graphic effects including shadows, particle effects, motion blur and a glow. The game logic should include a nice steering, powerups and a random level creation. A few other different things we wanted to have where a HUD, a skybox, sound and cannons aiming automatically for the player.

2 Milestone 1

For the first Milestone we started to get a first glance at working with ACGL, GIT and GLM. We started to create a basic concept of our game and built a prototyp on which we could work on. So the prototype included a brick flying straight forward and it was possible to direct it through a tunnel. This tunnel just consisted out of a few pieces and wouldn't go on forever. To have this basic prototype we figured out how to create graphic content on the screen and also learned how to create objects in Blender. There was much time spend on trying out graphic formulars from the internet and how to use them in our game. We also had to figure out which calculations we had to make for the shaders or the flight of the spaceship so it would stay parallel to the spline through the tunnel. To get some physics into the game we added Bullet as a physic engine. For the skybox we just took a basic picture of space.

So with our prototype we had a good basic to work with. We were able to add new features and could see the output of our code directly on the screen.

3 Milestone 2

With this basic we started to implement the first graphic effects and improved the graphics of the 3D models. We started to implement the glow effect which works in three steps. Rendering the glowmap, blur the glowmap with an Gaussian blurring algorithm and then blending the glow map with the rendered scene. Another effect that was done was the Motion Blur. This effect uses a texture mask containing the surfaces of static objects and in a second step it uses a smooth radial blur on the finished scene wherever the mask is black to create a feeling of speed.
The models where quite improved so instead of an brick we had a real space ship which was flying through the tunnel. We also added a left and a right curve to the tunnnel so it would become a little more diversified. We also started to add UV maps to the models so they would get a texture. Here we had spent pretty much time to figure out how this works and to find some nice textures to add.
We also improved the level generating so that the tunnel would go on forever and it was randomly chosen if it would go on straight or if there would be a curve. To avoid having way to much storage filled with a giant tunnel we decided to have only four parts of the tunnle as an object at once. There are at all times three parts ahead

of the space ship and one behind. Everytime the space ship entered a new part we deleted the one behind and added a new one to the tunnel.

So now our project already included a few graphic effects and the basic game concept. But we still had to fix a lot regarding the graphic effects and many bugs. For example every now and then the space ship just jumped away from the spline and out of the bounding box surrounding the tunnel. The graphic models did have some parts through which you could see through and space ship was way to large scaled compared to the tunnel. So there was much stuff left to do and many little fine tuning to get a game with nice graphics and a good game concept.

4 Milestone 3

Because there was much stuff to do for the last Milestone we now met more frequent and added many new features like powerups, particle effects or a new skybox picture and also erased many of the small bugs.

4.1 Obstacles and powerups

To give the player a little more to do than just flying straight we added some obstacles which would demolish the space ship if the player doesn't avoid them. We created different obstacles in Blender and added spots in the tunnel where it was possible to have them randomly appearing. With the progress of the game the chances for an obstacle to appear would increase so the game would get harder the further the player flies through the tunnnel. We also implemented cannons to the game which are shooting at the player with lasers. Those cannons consist out of three different meshes so they can actually turn if the player is going up or down or passes one of them.
To get life back after hitting an obstacle or to prevent getting damaged by one we added different powerups for the player which also appear randomly within the tunnel. The health powerup gives the player life back as long as he doesn't have full life yet. The ammunition powerup gives the player the possibility to fire two straight laser shots to destroy an obstacle before hitting it and the speed power up speeds the ship up for a few seconds which might make it harder for the player to avoid the obsacles. One problem with this powerup was that if it lasts too long the player will be in the next section of the tunnel before it ends and the powerup will be deleted so the player will never get a command to slow down. This was fixed by decreasing the time of duration. The last powerup was a shield which got a shield effect if the player flies through an obstacle or gets hit by a laser cannon.

4.2 Gameconcept and HUD

Now that it was possible to get damaged we changed the game concept to a game where it is the goal to fly as far as possible instead of passing through different levels. So if the life of the player is gone we show up a game over with the distance the player made and the possibility to restart the game. Within the process of creating the HUD we also added a small game menu where it is possible to start a new game, take a look at the credits and to exit the game.
To give the player an overview on the different stats we added a heads up display which shows the distance we already travelled, the capacity of ammunition the player has right now, a percentage of the shield and also a percentage of life left.

4.3 Fixes and improvments

There were many different little things we had to improve after the second milestone. We had a few problems with the collision detection by Bullet. For example every now and then the space ship jumped away from the spline and was lost in space.
Another thing to be done was to make the following camera more smooth because it was very choppy while the player was flying through a curve. We also improved the fact that the camera was always on the same height, so if the player was avoiding an obstacle the camera would just fly through the obstacle and block the view onto the upcoming tunnel. So we tried to improve this by changing the height of the camera depending on the height of the space ship.
Furthermore we had to make many changes regarding the models of the tunnel and the space ship. We added way more details and fixed little problems such as wrong normals or that the curves were not fitting perfectly together. To make another sensible use of the glow effect beside the lasers and the engine of the space ship we also added many different parts in the tunnel which are glowing.
In addition we had to fix the sensibility of the steering because it was very rough. So we lowered the acceleration to make it more smooth and easier for the player to navigate the ship through the tunnel.
To get a better game atmosphere and more realistic effects we also added a particle and damage effect. The particle effect is used if the space ship gets hit by a laser from one of the cannons or collides with an obstacle. We are also using the particle effect to create fume coming from the space ship if the life is low. So the less life the player has left the more the ship fumes.
To show the player that he lost life more obviously we added the damage effect. This is implemented as a red frame surrounding the screen.

To get a real computer game we also added sound effects to the game and a tune playing in the background. The different sound effects appear if the player fires a laser, gets hit by one or collides with an obstacle.

5 Final result

So in the end we created a jump'n'fly game which includes a couple of different graphic effects, a generic level generation and increasing difficulty created by the increasing chance of obstacles or cannons and a slow permanent acceleration of the space ship.
During the process a few problems occured concerning wrong textures which caused a misbehaving framebuffer or the use of Bullet. We also didn't create a nice shadow mapping but all in all we are satisfied with the result of our game.

The practical course gave us the possibility to get a nice overview over the capabilities of computer graphics and in some parts a deeper understanding. We were able to see how it is to work in a little software project and figured out where we had to make cuts in our project to get everything important finished until a special date. We liked the open way the project proceeded so everyone was able to contribute own ideas and arrange the time we are working for the project by ourself. Another good point is the fast and friendly support from the i8 team and other competitors. The only thing that could make it a little easier would be to give the students a fundamental code where students without any knowledge concerning computer graphics can see and understand how such a project could be structured and to have some examples. All in all even if it took much time it was a nice project and a good experience for everyone to see if he wants to take a further step working with computer graphics or if he is able to work in a software project at all.