DirectX Term Project

GMNG 4310

Jace Regenbrecht, Eduardo Brilanti, Aaron Cardinale, Juan Esparza

May 2, 2017

1 Member responsibilities

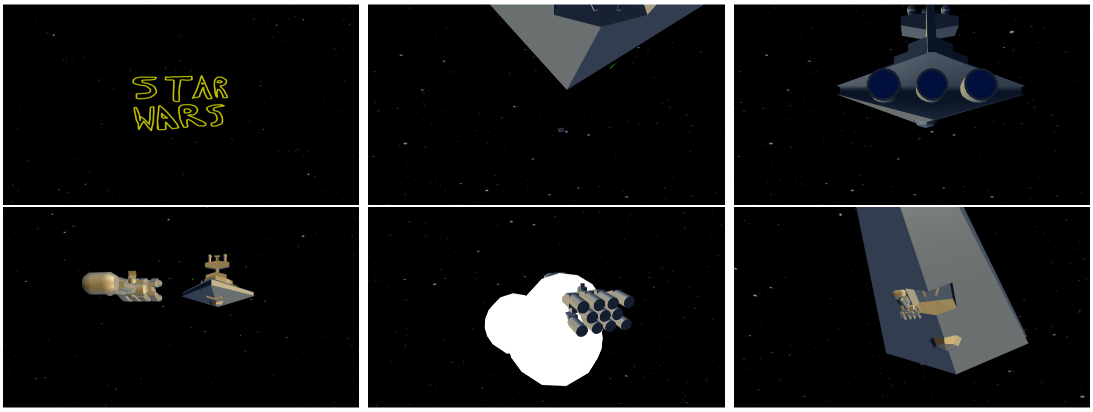
1.1 Eduardo Brilanti. Provided model and texture work for the title crawl text and rebel cruiser.

1.2 Aaron Cardinale. Provided model work for the star destroyer and blaster shots.

1.3 Juan Esparza. Implemented audio code work and provided files for intro music and explosion sound effect.

1.4 Jace Regenbrecht. Designed code base utilized by group, provided blaster audio and background texture work to project.

2 Application description



Our application is a two-minute short animation recreating the Star Wars: A New Hope opening scene. The animation opens with the famous Star Wars title crawl before panning down at the 1-minute mark to reveal the ensuring chase scene. Around the two minute mark the scene switches to a closeup of the rebel cruiser being disabled by a shot from the star destroyer, allowing it to be captured and ending the animation.

3 Program status

Program has no known runtime issues and has been built and run in multiple Windows 10 environments using Visual Studio 2015 without issue. Only issues with compilation may arise in environments running Windows versions earlier than Windows 8, as the project utilizes XAudio 2.8 for it's DirectX audio implementation.

4 Code breakdown

Our project utilizes the Microsoft's DirectX Tool Kit[1] library used in the labs, providing a useful set of classes to simplify the use of more complicated DirectX features. In addition, Chuck Walbourn's DirectX Visual Studio Templates[3] was used as template code for the project, providing pre-completed window draw and update functions and thereby allowing us to focus on developing our project's DirectX infrastructure. Overall, our project includes around 750 lines of custom code with most work being done in the Game.h and Game.cpp files, plus the addition of completely custom files Maths.h, BlasterFlash.h, BlasterFlash.cpp, Blaster.h and Blaster.cpp.

Majority of the work done in the Game.h header is that addition of multiple objects and variables past line 75. In Game.cpp most of the game logic is handled in the Game::Update function with our custom code ranging between lines 68 and 290. Our render and drawing work is done in the Game::Render function between lines 306 and 373, with object creation and configuration being handled in the Game::CreateDevice function between lines 543 and 708. Additional setup work is also done in the Game::CreateResources function between lines 832 and 866. Some additional logic was also added to the Game:OnDeviceLost function between lines 873 and 899.

5 Acknowledgments

The DirectX tool Kit GitHub wiki[2] was instrumental in providing the much-needed documentation that helped to make make this project possible. It's concise tutorials and thorough documentation are what ultimately led to the choice of DirectXTK as our DirectX library of choice over OGRE 3D.

References

[1] Microsoft. Directx tool kit. `https://github.com/Microsoft/DirectXTK`.

[2] Microsoft. Directx tool kit wiki. `https://github.com/Microsoft/DirectXTK/wiki`.

[3] Chuck Walbourn. Direct3d game vs project templates. `https://github.com/walbourn/directx-vs-templates`.