Markus Krugel

AGP AE2 Report

Contents

[Controls 2](#_Toc536015951)

[Decisions 2](#_Toc536015952)

[Problems 2](#_Toc536015953)

[Testing 3](#_Toc536015954)

[Conclusion 3](#_Toc536015955)

[References 4](#_Toc536015956)

# Controls

The player is being moved by pressing the WASD keys. To rotate you can press Q and E. By pressing the X key you can perform a jump. At last you can attack a nearby enemy by pressing the space bar key.

# Decisions

After a while I decided to open a github repository as a backup for this project (Reference 4).

The GameTimer class has been taken from the book “Introduction to 3D Game Programming with DirectX11” from Frank D. Luna because it was easy to understand and had the functionality that I needed (Reference 1). I used the GameTimer inside the main class to update the game based on the time passed. Furthermore the timer is being used inside the characters to simulate an attack cooldown.

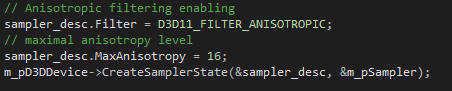
I decided to do the state machine of the enemies with a simple enum after a first attempt by solving it with own classes. Because of many errors and the realization that an enum should be enough, this decision was made.

The referred rendering, mip mapping and the spot light were not implemented because of the complexity of these subjects because of the many issues I had during the development of this project.

The menus were also removed because of the small priority they had in this project.

Implementing the anisotropic filtering was made by creating a D3D11\_SAMPLER\_DESC and afterwards using these simple commands:

(Reference 2)



The samplerDesc is basically the settings for the samplerState creation.

The first command sets the filter type to be anisotropic filtering and the second command sets the maximal level of anisotropy. Finally we create the samplerState based on these values.

# Problems

One problem, which occurred multiple times during this project was an exception when I tried using XMMatrixTranspose(). As it seems that there is a problem with the alignment of the matrices. To stop this problem to happen I worked further I changed the Visual Studio Project to run in x64 instead of x86. After a while I returned back to this problem and could solve it by following the answer of reference 3.

Another problem I had was to display the Skybox. My first attempt was to draw the box of vertices like in the beginning of the lab tutorials. For unknown reasons it did not display the skybox. Afterwards I copied the code from the Model class and modified it to fit to the skybox. That would still not solve the problem with the skybox.

Furthermore, I had a problem with the state machine. There appeared many errors when I tried to do the state machine and their states as own classes. After that I decided to implement the state machine by an enum and simply add the functionality behind the states inside the enemy class.

Another error I got was the LNK2019 error when I tried to work with the GetLookDirection function. I solved this problem by using the xyz structure from the objectfile class instead of an xmvector. After the change I noticed that this problem could have been occurred because I included the xnamath and the math file. My theory behind that is that maybe both classes have their own implementation of xmvector and therefore could not figure out which xmvector to use.

The last problem I could not solve was that during the draw call of the Model an “Access violation reading” error appeared when updating the constant buffer.

# Testing

Because of the many problems I faced during the development I could not test this project properly. The only things I could properly test were the enemy state machine and the controls. Both work as intended.

I tested the state machine by changing the distance between the enemy and the player. When the player is not in the sight range of the enemy, the enemy will follow their routine in patrolling between two points. Once the player is in sight range the enemy will follow the player and attack him when he is nearby. On the contrary the enemy will return to patrolling when the player escaped out of his sight range.

The controls were tested by simply pressing the specified keys I used and seeing if the correct actions were performed.

# Conclusion

My biggest problem was that I underestimated the complexity of this project. During the planning phase I should have invested more time in the research and afterwards should have started way earlier with the work on this project. Because of my poor time management I could not finish this project with a satisfying result. On the other hand during this project I learned many valuable information about game programming with C++.

# References

1. Frank D. Luna, (2012). Introduction to 3D Game Programming with DirectX 11 Dulles: David Pallai. pp. 116–123

2. Microsoft, (2016). Microsoft Docs. [Viewed 23 January 2019]. Available from: <https://docs.microsoft.com/en-us/visualstudio/debugger/graphics/point-bilinear-trilinear-and-anisotropic-texture-filtering-variants?view=vs-2017>

3. Necrolis, (2014), Stackoverflow. [Viewed 23 January 2019] Available from: <https://stackoverflow.com/questions/22133742/dx11-crash-when-accessing-xmmatrix>

4. Markus Krugel, (2019), Github. [Viewed 23 January 2019] Available from: <https://github.com/Markus-Krugel/AGP-AE2>