DotallRTTCG

Defense of the Ancients Real Time Trading Card Game

(Tested on nVidia and Intel Iris)

Controls:

WASD	Move the free floating camera
Space	Move up
Shift	Move down
F	Wireframe Mode
X	Starts the Game and sets the camera to overview
1	Player Red card on Position 1
2	Player Red card on Position 2
3	Player Red card on Position 3
4	Player Red card on Position 4
Num1	Player Blue card on Position 1
Num2	Player Blue card on Position 2
Num3	Player Blue card on Position 3
Num4	Player Blue card on Position 4
F3	Wireframe Mode
F4	Texture Sampling
F5	Mip Mapping
F6	Cheat Mode
F8	View Frustum Culling

Starting up the Program:

Starting up, you will see 2 trees and 2 red footies as render test objects. To begin the game press the X button.
The Camera will be positioned automatically and the competing players can start playing. You have full control of the camera all the time.

Step by Step Easy win:

Press F6 to activate cheat mode, spam Keys 1 to 4, and watch the horde.

Gameplay:

The Players have 4 hand cards. If they have enough mana, they can spawn units (keys 1-4, num 1-4). The mana for each player is displayed at the bottom on the resepective side. The units will run towards the enemys castle and attack it as well as any other enemy unit in their way. If one castle is destroyed the other team has won the game.

Units:

Footie	Melee DPS
Ranger	Range DPS (fucking OP right now, pls nerf)
Wizard	Range DPS (glasscanon)
Cleric	Healer

Cheatmode:

You cheap bastard!
By pressing F6 you enter the cheatmode, in cheatmode you no longer spawn the units on your hand but you spawn the following units for FREE.

1	Red footie
2	Red archer
3	Red wizard
4	Red cleric
Num1	Blue footie
Num2	Blue archer
Num3	Blue wizard
Num4	Blue cleric

Textures and Lights:

All the objects you can see were textured with handdrawn textures. The models except the heightmap were selfmade and UV-mapped in blender. On the heightmap a set of tiling textures was used. These textures blend based on the heightlevel.
The world objects are illuminated by a directional light. There is a specular reflection on some parts of the unit. We choose not to have specular highlights on heightmap, but it is illuminated by the same directional light as the other objects.

Cel Shading with Backfaces:

All units on the field and their projectiles are cel shaded with black outlines.

Shadows:

All units cast a shadow on the heightmap but not on each other.
We used shadowmaps and PFC filtering on the shadowmap.

A spinning footie that casts a shadow on the heightmap is located directly left of the starting position of the Camera.

Tesselation:

The heightmap is generated by a tesselation algorithm which calculates the screensize of a patch and tesselates it according to its size. This effect can best be viewed in Wireframe Mode.

EffectsTabel:

Shadow Maps (with PCF)	1.5
Tessellated Terrain LOD	2
Cel Shading	0.5
+ Contours (backfaces)	0.5

Frustum Culling:

Can be activated as defined in the cgueWiki.
It will also display how many objects are culled and how many objects are drawn.

Texture Quality:

Can be activated as defined in the cgueWiki.

Usage of the Alpha channel:

We use the alpha channel on our interface objects but we haven't implemented the possiblity to turn it off.

Animations:

The footies spin their swords when they attack and the clerics raise their staff when they heal friendly units. The ranged units lob projectiles on their enemys.

Wireframe Mode:

Can be activated as defined in the cgueWiki or if you press "F"

Frametime:

The Frametime is displayed in the Information section and cannot seperatly be hidden.

Modeling and Texturing:

All our models were selfmade and UV-mapped in Blender. Our textures and the Interface art were selfmade in Paint.net.

Used Libraries:

http://www.assimp.org/ for modelloading
http://openil.sourceforge.net/ for loading the raw data of images which we then turn into textures
https://www.freetype.org/ for loading glyphs to display Text

For Opengl we used glfw and glut.
http://www.glfw.org/
https://www.opengl.org/resources/libraries/glut/

Used Tutorials:

http://www.mbsoftworks.sk/index.php?page=tutorials
http://www.opengl-tutorial.org/
http://learnopengl.com/
http://sunandblackcat.com/tipFullView.php?l=eng&topicid=15
Tesselation:
http://codeflow.org/entries/2010/nov/07/opengl-4-tessellation/
http://prideout.net/blog/?p=48
http://in2gpu.com/2014/07/12/tessellation-tutorial-opengl-4-3/