Connect Four

Lu Wang

L236wang

20389964

CS488 Final Project

Purpose:

Linking the things we learn to 3D game development by making a simple Connect Four game.

Statement:

Create an interactive scene which mainly contains a Connect Fore board, pieces and a table. User could play the game. For example, pick a piece and put it into the board. Each model will have its own texture and material effect. Such as the board itself will looks like plastic, which have a little bit light reflection. Pieces will also have plastic material but also have its own color and texture, which will use texture mapping and bump mapping. Will use mip-mapping to pre-calculate, optimize sequences of textures.

The goal is make user play it as they play the game in real world. To achieve this, first I need to let the scene looks real. This mainly require implementing algorithms using OpenGL shaders. Which will include texture mapping, bump mapping, shadows mapping, and possibly moving light source. Second, add simple game logics, application will know when game is end and after each play finish his or her tern camera will rotate to the other plays side. Also when we putting a piece into position, the piece will fall in itself by gravity and stop when it hit to another piece. Third (additional factures), add a simple AI let player play with it and p2p connection to let players play with each other.

CS488 Final Project

Technical Outline:

Texture mapping:

The pieces are just cylinders without color which could not detect the side of each player map it with color can solve this problem. Also we could add pattern to the piece and table.

Bump mapping:

Cylinders have no carved pattern on it which normal piece do have it. Using bump mapping will make it much easier to show the pattern then using a carved 3d model.

shadow mapping:

Shadows are created by testing whether a pixel is visible from the light source, by comparing the pixel to a z-buffer or depth image of the light source's view, stored in the form of a texture.

Mirror Reflection:

The table and some other model may have higher shininess which may cause mirror reflection. Will involve a stencil buffer.

Ray picking:

Using another picking technique to select objects also a boundary detect technique to optimize selecting time

Modeling:

Use my own model instead of downloading from internet. Mainly by creating a game board, and pieces texture.

Text rendering:

Add a notebook like scoreboard which will display game status. Render different character in different statues.

Rotating camera:

Player could change the direction which they are looking at. This viewing direction will be saved so after the camera switch back from one player to another, play do not need to adjust the camera again. Also, after a play place a piece, the camera will rotate to the opposite side change to the other play’s view.

Game logic: :

Once a player gets four pieces in a row, the game ends. Play cannot push piece in a row which is already full.

CS488 Final Project

Mip-Mapping

are pre-calculate, optimize sequences of textures, each of which is a progressively lower resolution representation of the same image.

Bibliography:

[1] Blythe, David. *Advanced Graphics Programming Techniques Using OpenGL*. Siggraph 1999.

[2] William Donnelly, Andrew Lauritzen. "Variance Shadow Maps". Retrieved 2008-02-14

[3] Tobias Martin, Tiow-Seng Tan. "Anti-aliasing and Continuity with Trapezoidal Shadow Maps". Retrieved2008-02-14.

[4] Whitted T. (1979) *An improved illumination model for shaded display*. Proceedings of the 6th annual conference on Computer graphics and interactive techniques

[5] A. Chalmers, T. Davis, and E. Reinhard. Practical parallel rendering, ISBN 1-56881-179-9. AK Peters, Ltd., 2002.

CS488 Final Project

Name: Lu Wang User ID:

l236wang Student ID:

20389964

Objectives:

__1: Texture mapping: Texture are correctly map to pieces.

__2: Bump mapping: carved pattern are shows on pieces correctly.

__3: shadow mapping: shadow is displayed correctly.

__4: Mirror Reflection: correctly implies the reflection.

__5: Ray picking: pieces can be correctly picked.

__6: Modeling: game board are made by student himself.

__7: Text rendering: can correctly display game record.

__8: Rotating camera: camera direction and angel can be changed.

__9: Game logic: can detect when game is end.

__10: Mip-Mapping: will pre-calculate, optimize sequences of textures.

Declaration:

I have read the statements regarding cheating in the CS488/688 course handouts. I arm with my signature that I have worked out my own solution to this assignment, and the code I am handing in is my own.

Signature: