

Erasure

Documentation

Aaron Wedral 01633070

The current 3DObjects were either created with the ECG_Solution's Geometry class or loaded from the assets with a mesh loader using the Aasimp library. The meshes use material and textures chosen manually. For this to work an object loader was implemented, iterating through the different parts of the mesh.

The GameObjects handle general functionality like updating and movement of different game objects with more specific methods for specific objects e.g. transforming user input into action in Player. Also a first person camera was implemented as a GameObject moving with the player.

The GameLoop runs in the main.cpp after initializing the game and updates as well as renders all the game objects in the game.

The post processing steps necessary for the contours is done by using a Framebuffer and additional Shaders drawing the rendered image again, after applying a filter to draw the contours.

All of the games objects are textured.

The models are downloaded from various sites offering either open or non-commercial use licences.

Gameplay

Your goal is to reach the end of the level (which is the end of the tunnel) without getting hit too many times by the enemies. To achieve this you can use simple first person movement and your Dashing ability, allowing you to quickly leave enemies behind (or below) every now and then. When you launch the game you see a directionally illuminated level with a tank, a dalek and a drone protecting the dalek as enemies moving around and causing some trouble. If you take 100 damage (in this version 3 collisions with any enemy) you lose the game and the game stops. You also see a hall which is illuminated by a point light in about the center. These enemies are moving their patrol routes trying to stop you from reaching the end of that hallway. With only basic movement, this is hard to achieve without being hit. So you will need to use your Dashing Ability in conjunction with the other movement to get there unscathed. The game either ends with you getting hit too many times or reaching the end of the tunnel. In both cases the game ends and the console tells you if you have won or not.

Controls

WASD - Directional Movement

Mouse - Camera

Right Mouse Button - Dashing Ability

Space - Jumping

Esc - Exit the game

F1 - Wireframe

F2 - Backface culling

Features

Smooth Movement - The movement feels smooth and uninterrupted, going smoothly between player states to handle walking, jumping and dashing.

Intuitive Camera - The camera moves like you would expect it to in a first person game. Even the mouse speed is adjustable in the settings.ini

Moving enemies - Enemies patrolling the level and hurting you on contact.

Complex mesh - The object loader was used to load a tank as a dummy enemy

Frame independence - All the movement (player, enemy) and mouse input uses the calculated delta time to ensure frame independent movement and turn speed

Textures - The game uses the ECG_Solution to map textures to different Meshes.

Adjustable Parameters - The requested parameters and more are adjustable via the settings.ini file

Win and Lose Condition - You can lose the game by being hit for 100 damage by the enemies which currently means 3 times or you can win by reaching the end of the tunnel.

Effects:

Cel Shading - to give this game a cartoonish feel, the colors and the lighting are quantized in steps. This gives the feeling of a less realistic game.

Contours via Edge detection - This Toon look is greatly improved by drawn contours on the edges of the meshes giving the game an even more comic like appearance. The edges are detected from the frames normals in a Post Processing step.

Hierarchical Animation - The dalek in the scene comes with an attached drone which is bobbing up and down while following the dalek's every move. This is done via Hierarchical Animation.

Additional libraries

Assimp was used for loading meshes. <https://github.com/assimp/assimp/releases>