SLASH OR DIE by VideoGameEnjoyers

Quick Start

Objective: Kill all enemies and retrieve the key

Controls:
- WASD - Movement
- Space - Jump
- Mouse - Camera
- Left Click - Attack

Implementation

The objective of the game is to “slash” all enemies in the room and then collect the key. To win, the player must kill all enemies and jump over a pool of lava onto a platform to retrieve the key. If the player’s hit points drop to 0, the game is over. All models in the game are wavefront obj files, made in blender and imported with Assimp. Textures were created in photoshop and saved in the mtl files of the models. The game is illuminated by a directional light source, which also provides the basis for the shadow mapping. The player is controlled by the WASD keys (movement), space (jump) and the mouse (camera control and attack). The camera is a third person camera that is always pointed to the back of the player. The game starts with a start screen. Upon pressing enter, the scene is drawn, and the player is free to start playing. Physx is used to detect collision between:
- Player and enemy
- Player and lava
- Player and key
- Sword and enemy
- Player and stage
- Player and platform

By left clicking, the player can attack in front of them. If the player hits an enemy enough times, it will die. Upon doing this a key appears on a platform, on which the player must jump in order to beat the game. If the player wins, a win screen appears; if they lose, a game over screen appears.

Features

Adjustable Parameters:
- F1 - toggle HUD
- F2 - toggle bloom
- F3 - toggle fullscreen
- F8/F9 - turn brightness up and down
- F - toggle FPS counter

HUD: The HUD displays the player's and enemies' HP. It can be toggled on and off

Attack: Left clicking will load the sword in front of the player and create a temporary hitbox.

Start/Win/Lose screens: Depending on the game state, a different screen will appear.

Third Person Camera: When the camera rotates, so does the player, meaning the movement is always relative to the front of the player.

Reset: The game can be reset upon winning or losing without closing the program.

Libraries

- Assimp: model loading (https://github.com/assimp/assimp)
- FreeType: on screen text (https://freetype.org/)
- Physx: collision detection (https://developer.nvidia.com/physx-sdk)

Effects

Bloom/Glow: Bloom is implemented on the lava that surrounds the platform. The tutorial at https://learnopengl.com/Advanced-Lighting/Bloom was used to help achieve this effect.

Vertex Shader Animation: Vertex shader animation is used on the lava again. It simulates the movement of the liquid. The tutorial at https://tuwel.tuwien.ac.at/pluginfile.php/2415211/mod_page/content/32/Animation_SS18.pdf?time=1619523068902 was used to help achieve this effect.

Shadow Mapping with PCF: Shadow mapping is implemented on the player, enemies, key and platform and sword. It uses the perspective of the directional light to create an orthographic projection, as if the light source was infinitely far away. The tutorials at https://learnopengl.com/Advanced-Lighting/Shadows/Shadow-Mapping and https://www.youtube.com/watch?v=9g-4aJhCnyY&ab_channel=VictorGordan were used to help achieve this effect.

Special Features

- Walking into the lava will bounce the player back