

CW2 Report

Setup

- Clean and rebuild
- If errors occur reinstall GLM and NupenGL
- Run and play!
- If using a .exe ensure the media folder and the vertex and fragment shaders are in the same file location as the exe

-Warning Currently has a bug where FPS becomes limited to 60 (includes other applications)

Controls

Jumping on a new platform gives you 1 score, the goal is to get the highest score you can.

- Forward - W
- Backward - S
- Left - A
- Right - D
- Rotate right - E
- Rotate left - Q
- Jump - Space Bar
- Double Jump - Space Bar (in air)

Program Functionality

Functions

Parse- This is the model loading function which was taken from my coursework

1 LoadTexture - function taken from coursework 1 to load textures for shaders

randomf - returns a random float between min and max inputs loadfloor-

initialises the VAOs and buffers for the platform objects

init - Initialises the player object VAOs and buffers with a texture using the parse and loadtexture functions.

updateInput - looks for keyboard inputs and performs the required action based on buttons pressed

CheckCollision - checks the input position against positions of all platforms to see if collision has occurred

Notable sections in main

lines 488 to 502 generate random positions for new platforms to spawn and remove old ones lines 504 to 516 apply gravity and jumping physics Loop at 532 draws all platforms lines 552 to 570 draw the player object

What separates my program

This program was built on top of the model loader i created for coursework 1, i have not used any additional libraries, this has resulted in creating my own physics and collision. This idea came from me wanting to understand movement and physics in a 3d environment (from the programming side) so i figured that a platformer where those are fundamental would be best.

Evaluation

I think what ive created is close to being the vision i had for the original game, in the sense that the mechanics are all there it just needed a little polishing, (additional

platform shapes more refined physics and textures for platforms). If i were to do this again i would try a much more object orientated approach as i found myself creating a number of global variables which could have been avoided and would make the code significantly more readable, and doing this would allow me to more easily change between shaders for objects and lead to a smoother program.

Video-<https://www.youtube.com/watch?v=zRsuYHaabxM&feature=youtu.be>