

PTVC Submission 2

The game is in a good playable condition and the necessary effects have all been implemented. For a better playing experience some gameplay-related changes could still be improved to make the game more exciting but all requirements are met and the game is as we imagined it.

3D Geometry (6 Points)

using the Assimp - Open Asset Import Library

imported OBJ - Wavefront Objekt

The complex Object in our Scene is the airplane orbiting the scene

Playable (3 Points)

3d Movement of a 1st Person character. Jumping from Platform to Platform to reach the top.

Dynamic generation of platforms to jump on.

Min. 60 FPS and Framerate Independence (3 Points)

Framerate independence and min. 60fps given.

Win/Lose Condition (3 Points)

The game is developed as an endless runner and therefore the win condition is to get as high/long as possible without falling. Falling down is the lose condition. The player gets an endscreen where he sees how long he managed to survive.

Intuitive Controls (2 Points)

Intuitive 1st person controls. Movement with WASD and Space to jump.

sprint with shift

ground pound with strg

Intuitive Camera (2 Points)

Intuitive 1st person camera. look around with your mouse.

Illumination Model (2 Points)

Each object has an assigned material

Point and directional light

Textures (2 Points)

Most of the objects in the scene have a texture attached to it

Moving Objects (2 Points)

The player is moving through the scene controlled by the user.

The Plane is constantly orbiting the scene

Documentation (1 Point)

Adjustable Parameters (1 Point)

You can toggle fullscreen Mode by pressing f4
toggle water animation pressing M
show Fps pressing f3
flymode pressing right control

Advanced Gameplay (5 Points)

Your time gets tracked to see how long you managed to survive. Platforms despawn after a certain time. You can collect a Powerup that improves your jump height permanently.

Collision Detection (Basic Physics) (6 Points)

We handle the collision detection in Physics between our character controller and the static environment. To see if the Player is able to jump a groundcheck gets performed by sending a raycast and check for collisions at the players feet.

Advanced Physics (4 Points)

Collision callback to collect the Powerup

Heads-Up Display (4 Points)

The player can see the fps the game is currently running at.
Displaying the timer of how long the player already survived

Blobby Object using Raycasting (12 Points)

Blobby object which is used as powerup.

Vertex Shader Animation (8 Points)

Water below the platforms is animated using Vertex Shader

Vertex Shader Animation (8 Points)

Water below the platforms has an animated procedural texture

Lens Flares (8 Points)

Loading the lens flare textures and rendering them when looking into the lightsource.
Therefore an imaginary line is drawn from the lightsource through the center of the camera and along this the textures are rendered.

total points= 82