Sprout's Adventure: Documentation

Computergrafik UE SS 2021

Florian Wagner (11907095), Gabriela Anamaria Olar (11907089)

June 16, 2021

1 Documentation

1.1 Important Information

After our first submission it was brought to our attention, that our game might be just a black screen. We found out, that this occurs when the game is run with the integrated (CPU) graphics of the system. If the game is forced to be played with a dedicated (only NVidia was tested) GPU everything works as expected.
We also noticed that on a laptop it can occur that the game is being locked at 30FPS without vsync being enabled. Why exactly this occurs, we do not know.

1.2 Gameplay features

We have a small story driven game where Sprout, our main protagonist has to rescue his friend. The player can be moved using the **W A S D** keys and can jump by using **SPACE**.
Sprout will face a few enemies along the way which he can defeat by jumping on them. Jumping on enemies will decrease their health.
Sprout's Adventure is separated into levels:

- Beginning area.
  Here Sprout finds out his friend has been kidnapped.
- Tutorial Area.
  Here the player gets to know the games mechanics.
- 2 Levels where Sprout has to fight enemies.
- The Boss Level

Sprout can only advance into the next level (except the Tutorial level) after he defeated all of his enemies. If Sprout loses all his health the current level will reset. The player can anytime pause and or quit the game after they press the **ESC** key.

1.3 Effects

- Shadow-Mapping with PCF.
  For the directional light (Sun) in our scene. As described at learnopengl.com[1]
- CPU Particle System using instancing
  It is visible when Sprout moves. As described at opengl-tutorial.org[2]
- Specular and Environment Maps.
  (although they both were already implemented in ECG) were also implemented.
- PBR and Normal (Phong) rendering.
  For rendering we use both PBR and normal rendering. By default everything will be rendered with the PBR shader. The render method can be toggled using the **F1** key. The normal maps are either generated or loaded from normal map textures. To see those, Press the **F3** key (see keys below). Both as described at learnopengl.com[1]
- Bloom/Glow.
  As described at learnopengl.com[1]
- Something we also implemented a DOF (Depth of Field) shader.
- Sprout's Healthbar also features a procedurally generated (moving) noise texture using FastNoiseLite.
- We are rendering in HDR and convert into LDR using an ACES color correction curve taken from[3]

1.4 Additional Libraries

- IrrKlang for audio.[4]
- Freetype for Text and Font rendering.[5]
- Assimp for Model-Loading.[6]
- FastNoiseLite for noise generation.[7]
- Bullet Physics.[8]

1.5 Key Controls

- W = move forward
- S = move backward
- A = move left
- D = move right

- R = interact
- SPACE = jump

These following key controls only work during a level:

- F1 = toggle render method
- F2 = toggle Face Culling
- F3 = toggle draw Normals
- F4 = toggle show collision boxes
- F5 = toggle show DOF map
- F6 = toggle control camera with mouse (rotation only!)
- 3 = increase focal range for DOF (press F5 to see changes live, changes will apply regardless)
- 4 = decrease focal range for DOF (press F5 to see changes live, changes will apply regardless)
- ESC = Opens a menu

References

[1] LearnOpenGL `https://learnopengl.com/` last accessed: -16.06.2021

[2] Opengl-Tutorial `http://www.opengl-tutorial.org` last accessed: -16.06.2021

[3] Aces Curve `https://knarkowicz.wordpress.com/2016/01/06/aces-filmic-tone-mapping-curve/` last accessed: -16.06.2021

[4] IrrKlang `https://www.ambiera.com/irrklang/` last accessed: -16.06.2021

[5] FreeType `https://www.freetype.org/` last accessed: -16.06.2021

[6] Assimp `https://www.assimp.org/` last accessed: -16.06.2021

[7] FastNoiseLite `https://github.com/Auburn/FastNoiseLite` last accessed: -16.06.2021

[8] Bullet-Physics `https://pybullet.org/wordpress/` last accessed: -16.06.2021