WaveIgnition

Controls:

- **A,D –** Roll Sideways
- **Mouse –** Look around/change direction
- **W,S –** increase/decrease speed
- **Left Mouse Button –** Shoot projectiles

Implementation of Requirements:

- **Gameplay**
  Player can fly around with mouse camera movement.
  W and S keys increase and decrease the speed of the plane.
  Left Mouse button shoots projectiles.
- **Complex Objects**
  Every model (with the exception of the water model) is stored as an external model file and loaded into the game at the start.
- **Animated Objects**
  The fire exhaust of the player plane is animated and translated relative to the player transformation matrix
- **View Frustum Culling**
  Not implemented
- **Experimenting with OpenGL**
  The bloom effect and fire exhaust of the player use blending (with different blend modes)

Features of the game:

- Completely made by one person. (Team-partner dropped out.)
- Game engine and the Game itself are two seperate projects which link against each other (Game Engine is compiled as a seperate .lib file. > Abstraction and re-use is easily possible.)
- Easy to use „Model Builder“ class which allows to quickly build and create models with basic shapes (spheres, qubes, etc…) by using basic commands.
- Easy to use Shader classes which automatically analyse the GLSL code and create appropiate uniform/attribute links at runtime.
- 360 degrees of movement around all axis with properly animated rotation of the plane character model

Illumination:

The game features one single light-soure (the sun) which illuminates all dynamic objects. (the player and enemy objects) with a cel-shading shader.

Static geometry (the inslands) are illuminated by using lightmapping. (Light is baked into the texture.)

Libraries used:

- Glm (math library)
- Assimp (model loading)
- SDL (openGL rendering context and input)
- FreeImage (image/texture loading)

Effects implemented:

- Lightmapping (0,5) – Island model uses lightmapping
- Cel-shading (0,5) – interactive objects (planes, ufos) use cel-shading
- Bloom (1) – screen space effect which blurs white spots on the screen

Tools used:

- Visual Studio Community 2017
- Blender (Models)
- Github
- Paint.net (Textures)