ASSIGNMENT - 4

VULKAN FINAL ASSIGNMENT

MODEL URL: *Porsche 911 GT2 Free 3D Model - .obj .max - Free3D (Doesn't require login)*

GitHub: COMP392_Advanced_Graphics_Assignments/Assignment_4 at main · Plaban9/COMP392_Advanced_Graphics_Assignments

```
void FirstApp::loadGameObjects()
{
    std::shared_ptr<LveModel> lveModel = LveModel::createModelFromFile(&: lveDevice, filepath: "Models/Porsche_911_GT2.obj");
    auto lve::LveGameObject porsche = LveGameObject::createGameObject();
    porsche.model = lveModel;
    porsche.transform.translation = { 0.25f, 0.0f, 2.5f };
    porsche.transform.scale = glm::vec3{ 0.5f };
    porsche.transform.rotation = glm::vec3{ glm::half_pi<float>() / 50.0f, glm::pi<float>() / 4.0f, glm::pi<float>() };
    gameObjects.push_back(_Val: std::move(& _Arg: porsche));
}
```

Fig 1.1: Loading of Porsche 911 GT2 Test model in code

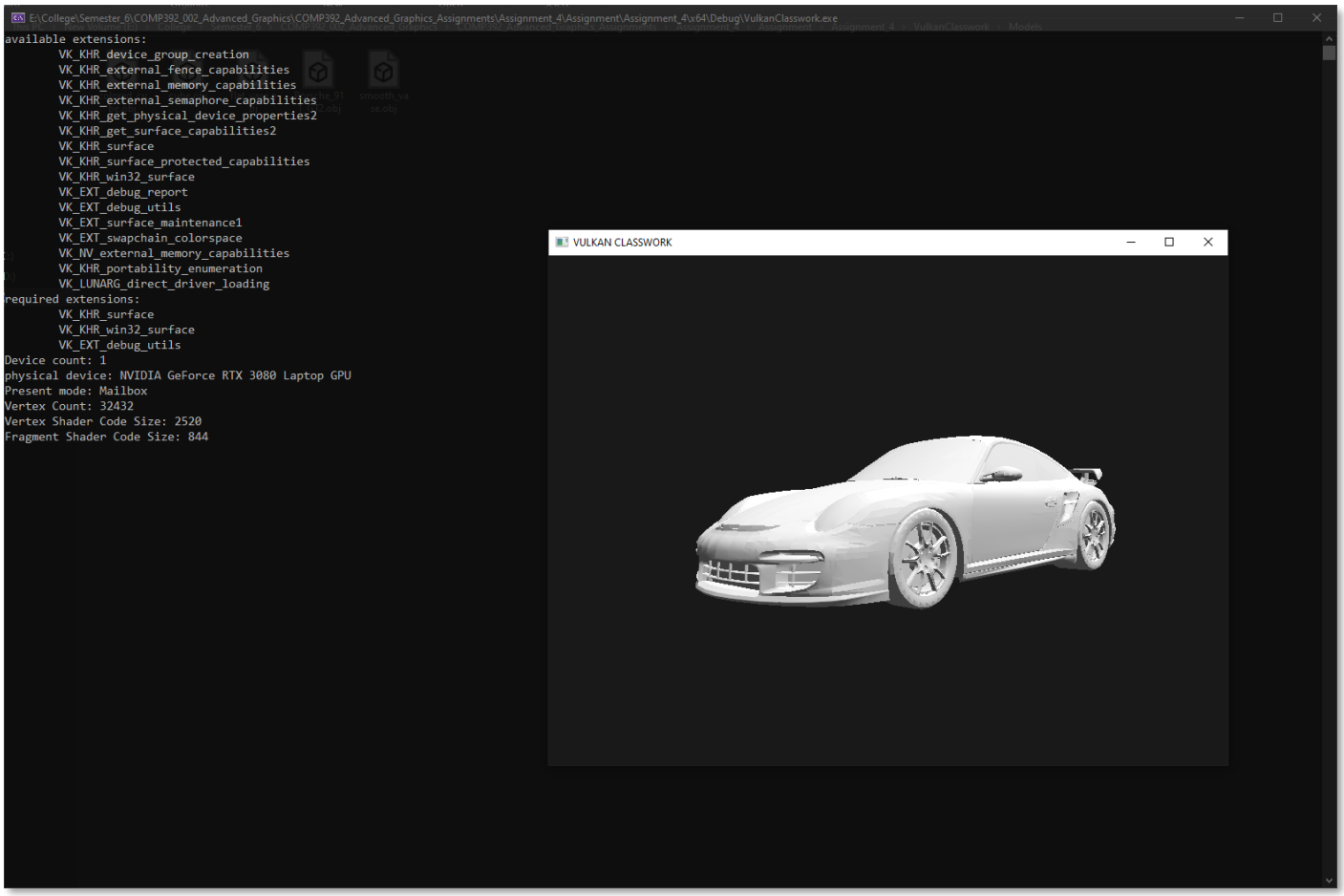
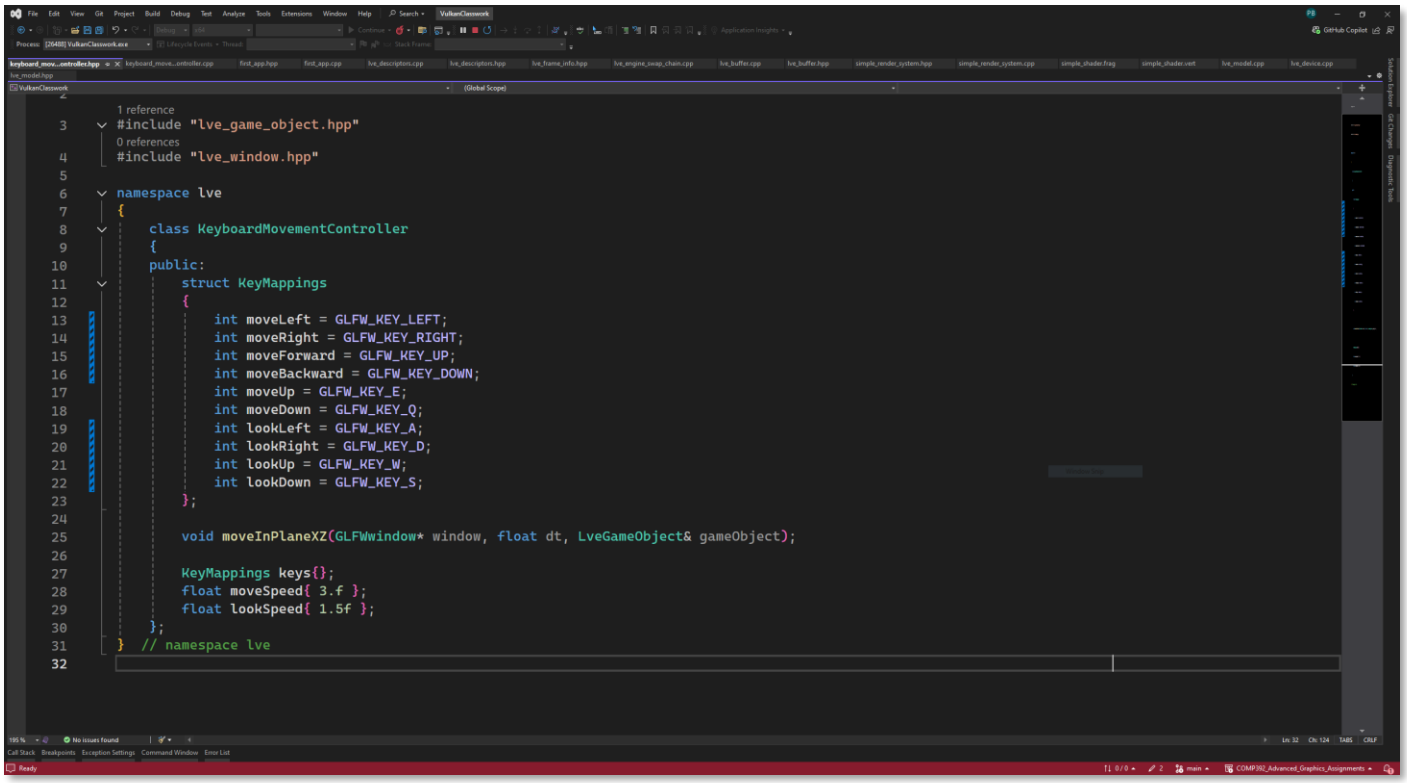


Fig 1.2: Execution of loading code of Porsche 911 GT2 Test model (No compiler warnings).



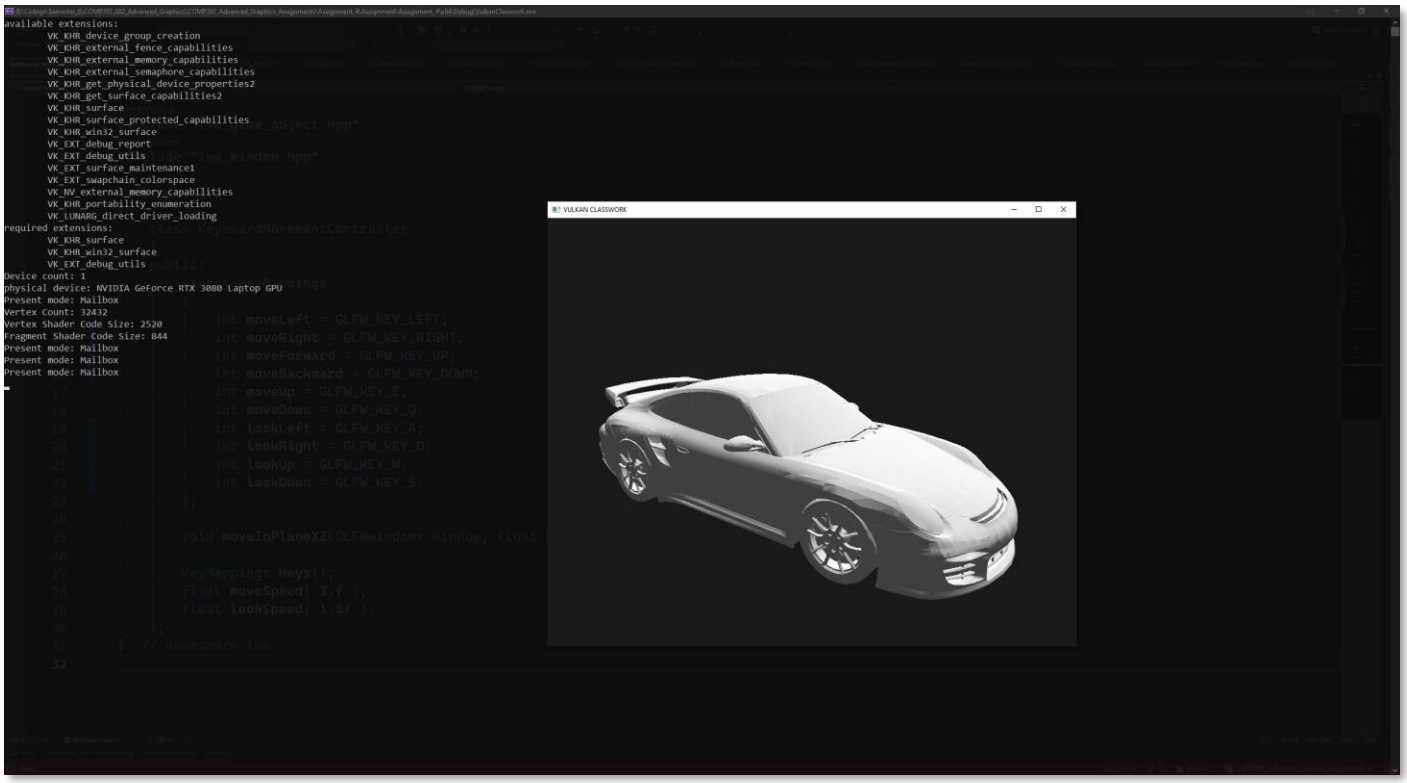
```
1 reference
#include "lve_game_object.hpp"
0 references
#include "lve_window.hpp"

namespace lve
{
    class KeyboardMovementController
    {
    public:
        struct KeyMappings
        {
            int moveLeft = GLFW_KEY_LEFT;
            int moveRight = GLFW_KEY_RIGHT;
            int moveForward = GLFW_KEY_UP;
            int moveBackward = GLFW_KEY_DOWN;
            int moveUp = GLFW_KEY_E;
            int moveDown = GLFW_KEY_Q;
            int lookLeft = GLFW_KEY_A;
            int lookRight = GLFW_KEY_D;
            int lookUp = GLFW_KEY_W;
            int lookDown = GLFW_KEY_S;
        };

        void moveInPlaneXZ(GLFWwindow* window, float dt, LveGameObject& gameObject);

        KeyMappings keys{};
        float moveSpeed{ 3.f };
        float lookSpeed{ 1.5f };
    };
} // namespace lve
```

Fig 1.3: Remapping of controls as per requirements with bonus controls.



```
available extensions:
        VK_KHR_device_group_creation
        VK_KHR_external_fence_capabilities
        VK_KHR_external_memory_capabilities
        VK_KHR_external_semaphore_capabilities
        VK_KHR_get_physical_device_properties2
        VK_KHR_get_surface_capabilities2
        VK_KHR_surface
        VK_KHR_surface_protected_capabilities
        VK_KHR_win32_surface
        VK_EXT_debug_report
        VK_EXT_debug_utils
        VK_EXT_surface_maintenance1
        VK_EXT_swapchain_colorspace
        VK_NV_external_memory_capabilities
        VK_KHR_portability_enumeration
        VK_LUNARG_direct_driver_loading
required extensions:
        VK_KHR_surface
        VK_KHR_win32_surface
        VK_EXT_debug_utils
Device count: 1
physical device: NVIDIA GeForce RTX 3080 Laptop GPU
Present mode: Mailbox
Vertex count: 32432
Vertex Shader Code Size: 2520
Fragment Shader Code Size: 844
Present mode: Mailbox
Present mode: Mailbox
Present mode: Mailbox
```


Fig 1.4: Running after remapping of controls (No compiler warnings or errors).