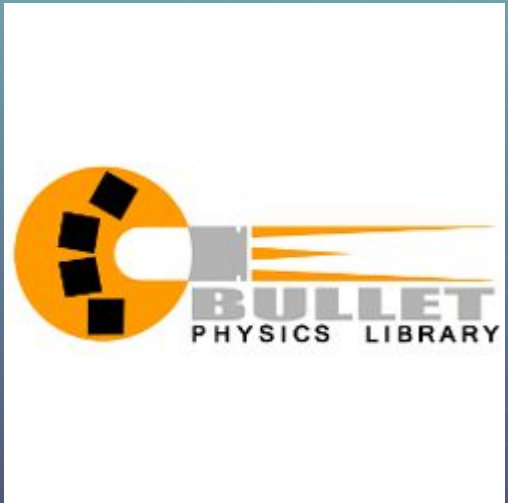
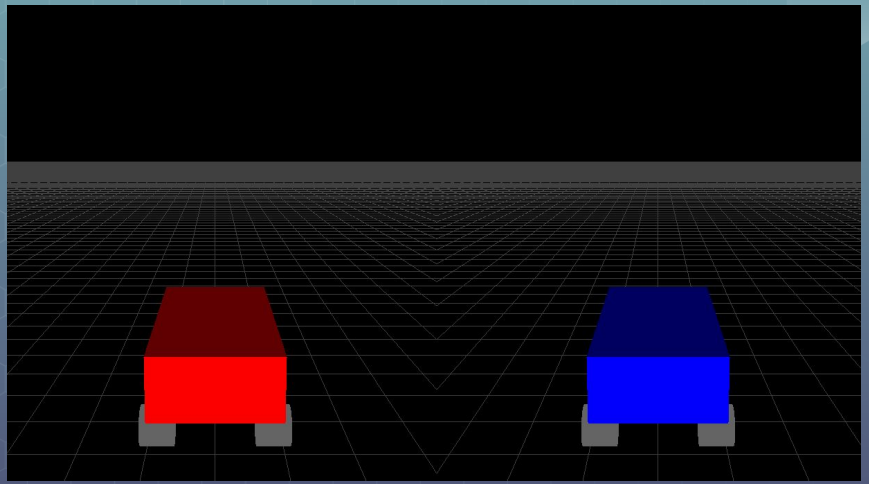
# Split Screen Research

Albert Robles

# Split Screen in video games


# Intro to the project



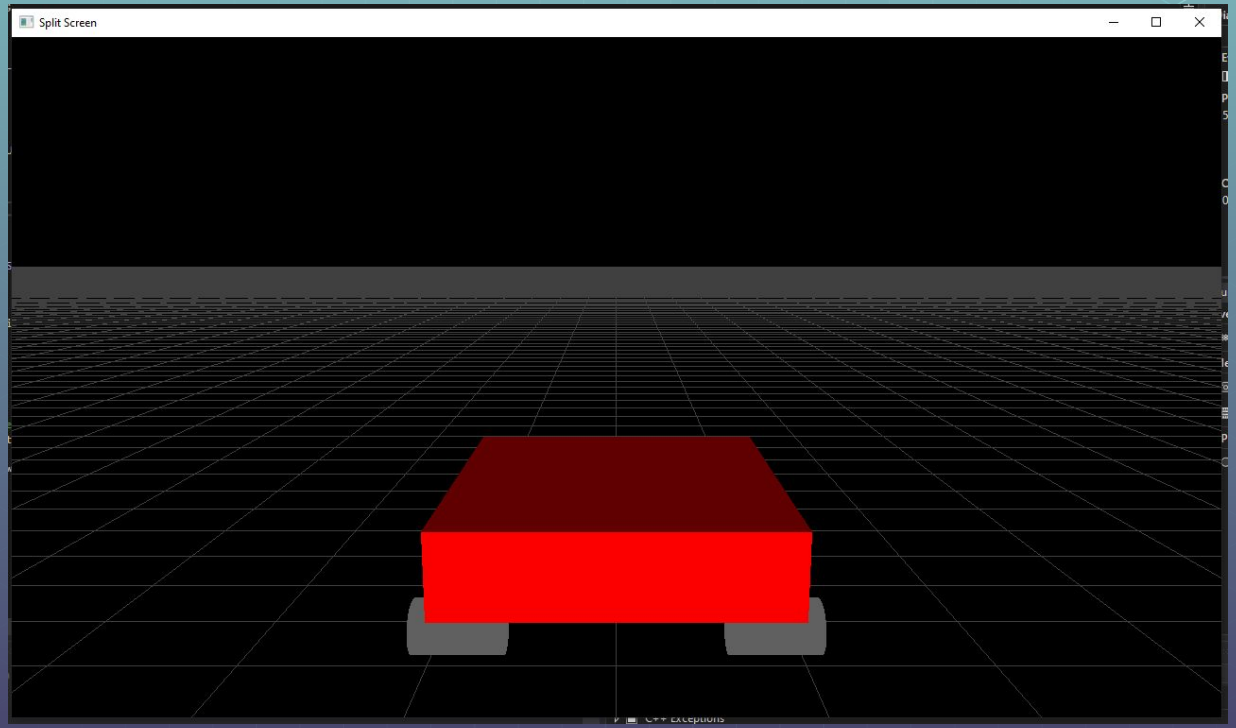



# Todo I Creating Second Camera

```cpp
Application::Application()
{
    //TODO 1 (Create Modules)

    window = new ModuleWindow(this);
    input = new ModuleInput(this);
    audio = new ModuleAudio(this, true);
    scene_intro = new ModuleSceneIntro(this);
    renderer3D = new ModuleRenderer3D(this);
    camera = new ModuleCamera3D(this);
    physics = new ModulePhysics3D(this);

    // The order of calls is very important!
    // Modules will Init() Start() and Update in this order
    // They will CleanUp() in reverse order

    // Main Modules
    AddModule(window);
    AddModule(camera);
    AddModule(input);
    AddModule(audio);
    AddModule(physics);

    // Scenes
    AddModule(scene_intro);

    // Renderer last!
    AddModule(renderer3D);
}
```

```cpp
class Application
{
public:

    //TODO 1 (Create Modules)

    ModuleWindow* window;
    ModuleInput* input;
    ModuleAudio* audio;
    ModuleSceneIntro* scene_intro;
    ModuleRenderer3D* renderer3D;
    ModuleCamera3D* camera;
    ModulePhysics3D* physics;
```

# Todo I Result



# Solution

```
//TODO 1 (Create Modules)

window = new ModuleWindow(this);
input = new ModuleInput(this);
audio = new ModuleAudio(this, true);
scene_intro = new ModuleSceneIntro(this);
renderer3D = new ModuleRenderer3D(this);
camera = new ModuleCamera3D(this);
camera2 = new ModuleCamera3D(this);
physics = new ModulePhysics3D(this);

// The order of calls is very important!
// Modules will Init() Start() and Update in this order
// They will CleanUp() in reverse order

// Main Modules
AddModule(window);
AddModule(camera);
AddModule(camera2);
AddModule(input);
AddModule(audio);
AddModule(physics);
```

```
//TODO 1 (Create Modules)

ModuleWindow* window;
ModuleInput* input;
ModuleAudio* audio;
ModuleSceneIntro* scene_intro;
ModuleRenderer3D* renderer3D;
ModuleCamera3D* camera;
ModuleCamera3D* camera2;
ModulePhysics3D* physics;
```

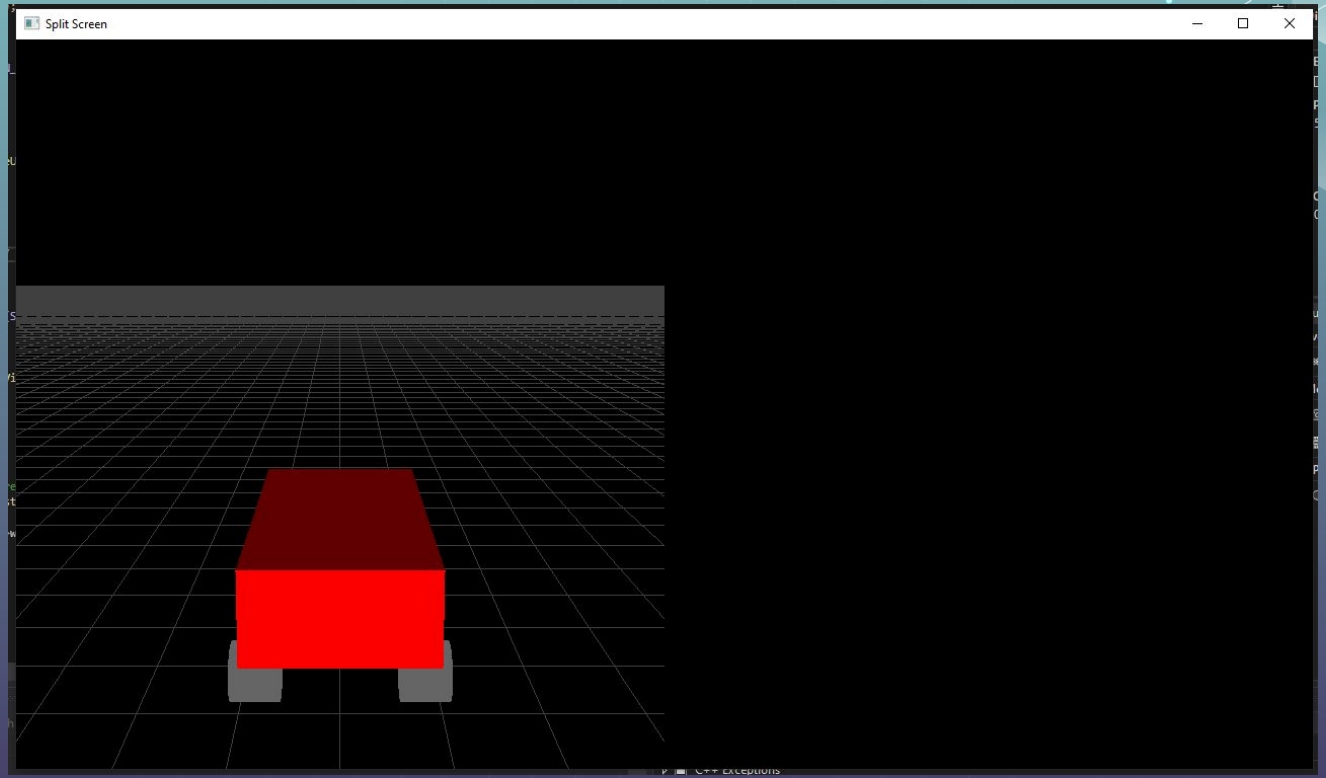
# Todo 2 Setting Up Ist Viewport

```cpp
// PreUpdate: clear buffer
update_status ModuleRenderer3D::PreUpdate(float dt)
{
    glClear(GL_COLOR_BUFFER_BIT);       //Clear Buffers

    //TODO 2 & 3

    glMatrixMode(GL_PROJECTION);    //Specify Current Matrix
    glLoadIdentity();               //Replaces Matrix
    gluPerspective(45.0, (GLfloat)(SCREEN_WIDTH) / (GLfloat)(SCREEN_HEIGHT) / 2, 0.1f, 500.0);      //Set Prespective
    glMatrixMode(GL_MODELVIEW);     //Specify Current Matrix
    glClear(GL_DEPTH_BUFFER_BIT);   //Clear Buffers

    glLoadMatrixf(App->camera->GetViewMatrix());        //Assign Camera

    App->Draw();

    return UPDATE_CONTINUE;
}
```

C++

```cpp
void WINAPI glViewport(
   GLint   x,
   GLint   y,
   GLsizei width,
   GLsizei height
);
```

# Todo 2 Result



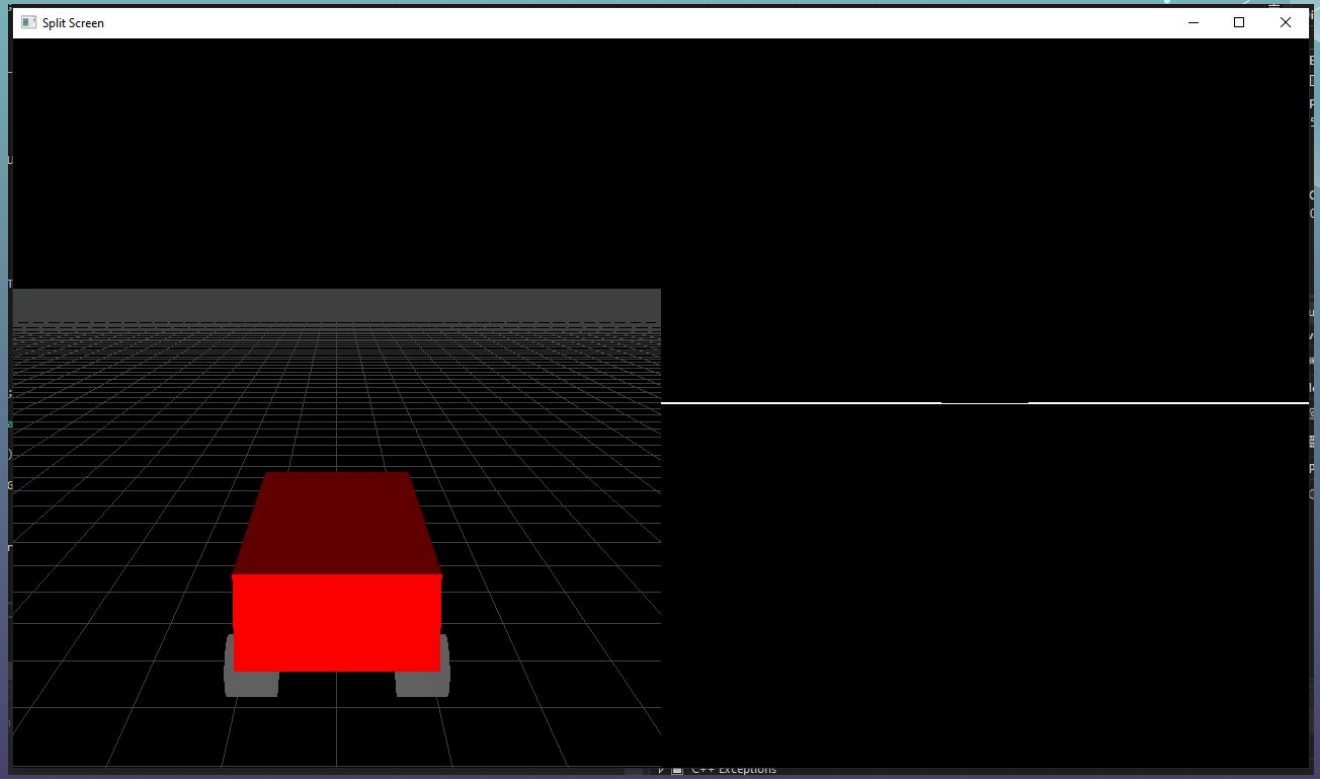
# Solution

```
glClear(GL_COLOR_BUFFER_BIT);       //Clear Buffers

//TODO 2 & 3 (Viewports)

glViewport(0, 0, SCREEN_WIDTH / 2, SCREEN_HEIGHT);      //Set Viewport

glMatrixMode(GL_PROJECTION);    //Specify Current Matrix
glLoadIdentity();               //Replaces Matrix
gluPerspective(45.0, (GLfloat)(SCREEN_WIDTH) / (GLfloat)(SCREEN_HEIGHT) / 2, 0.1f, 500.0);      //Set Prespective
glMatrixMode(GL_MODELVIEW);     //Specify Current Matrix
glClear(GL_DEPTH_BUFFER_BIT);   //Clear Buffers

glLoadMatrixf(App->camera->GetViewMatrix());        //Assign Camera

App->Draw();

return UPDATE_CONTINUE;
```

# Todo 3 Setting up 2nd Viewport

```
glClear(GL_COLOR_BUFFER_BIT);       //Clear Buffers

//TODO 2 & 3 (Viewports)

glViewport(0, 0, SCREEN_WIDTH / 2, SCREEN_HEIGHT);      //Set Viewport

glMatrixMode(GL_PROJECTION);    //Specify Current Matrix
glLoadIdentity();               //Replaces Matrix
gluPerspective(45.0, (GLfloat)(SCREEN_WIDTH) / (GLfloat)(SCREEN_HEIGHT) / 2, 0.1f, 500.0);      //Set Prespective
glMatrixMode(GL_MODELVIEW);     //Specify Current Matrix
glClear(GL_DEPTH_BUFFER_BIT);   //Clear Buffers

glLoadMatrixf(App->camera->GetViewMatrix());        //Assign Camera

App->Draw();

return UPDATE_CONTINUE;
```

# Todo 3 Result



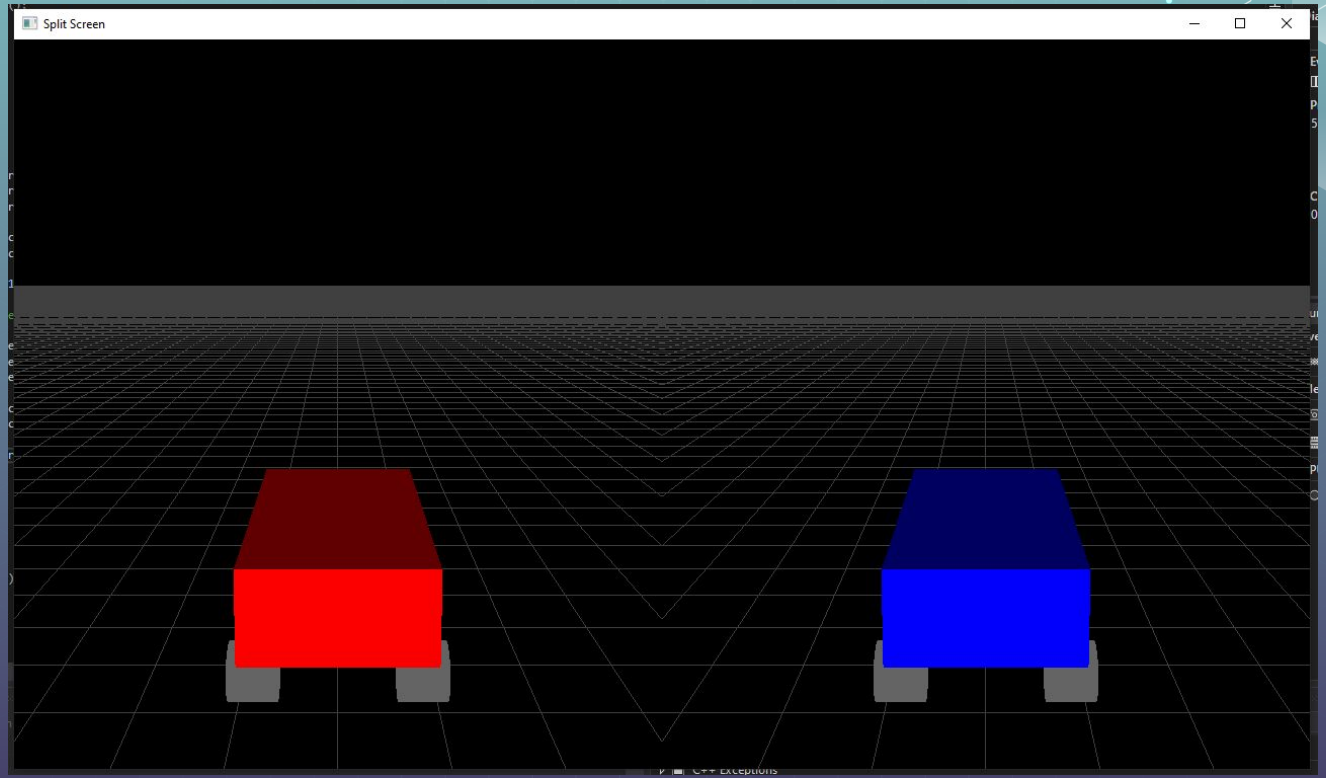
# Solution

```
glClear(GL_COLOR_BUFFER_BIT);       //Clear Buffers

//TODO 2 & 3 (Viewports)

for (int cameras = 0; cameras < 2; cameras++)
{
    glViewport(0, 0, SCREEN_WIDTH / 2, SCREEN_HEIGHT);      //Set Viewport

    if (cameras == 1)
    {
        glViewport(SCREEN_WIDTH / 2, 0, SCREEN_WIDTH / 2, SCREEN_HEIGHT);       //Set Viewport
    }

    glMatrixMode(GL_PROJECTION);    //Specify Current Matrix
    glLoadIdentity();               //Replaces Matrix
    gluPerspective(45.0, (GLfloat)(SCREEN_WIDTH) / (GLfloat)(SCREEN_HEIGHT) / 2, 0.1f, 500.0);      //Set Prespective
    glMatrixMode(GL_MODELVIEW);     //Specify Current Matrix
    glClear(GL_DEPTH_BUFFER_BIT);   //Clear Buffers

    glLoadMatrixf(App->camera->GetViewMatrix());        //Assign Camera

    if (cameras == 1)
    {
        glLoadMatrixf(App->camera2->GetViewMatrix());   //Assign Camera
    }

    App->Draw();
}

return UPDATE_CONTINUE;
```

# Todo 4 Assigning Camera 2 To Player 2

```
App->camera->Position.x = player1->vehicle->vehicle->getChassisWorldTransform().getOrigin().getX() - 10 * player1->vehicle->vehicle->getForwardVector().getX();
App->camera->Position.y = player1->vehicle->vehicle->getChassisWorldTransform().getOrigin().getY() + 5 * player1->vehicle->vehicle->getUpAxis();
App->camera->Position.z = player1->vehicle->vehicle->getChassisWorldTransform().getOrigin().getZ() - 10 * player1->vehicle->vehicle->getForwardVector().getZ();

float player1_x = player1->vehicle->vehicle->getChassisWorldTransform().getOrigin().getX() + 10 * player1->vehicle->vehicle->getForwardVector().getX();
float player1_z = player1->vehicle->vehicle->getChassisWorldTransform().getOrigin().getZ() + 10 * player1->vehicle->vehicle->getForwardVector().getZ();

App->camera->LookAt(vec3(player1_x, 1, player1_z));

//TODO 4 (Assign Camera to Player)

if (endGame && !start)
    resetLevel();
else if(!endGame && start)
    startRound();

char title[80];
sprintf_s(title, "Radio Control League");
App->window->SetTitle(title);

return UPDATE_CONTINUE;
```

# Todo 4 Result



# Solution

```
App->camera->Position.x = player1->vehicle->vehicle->getChassisWorldTransform().getOrigin().getX() - 10 * player1->vehicle->vehicle->getForwardVector().getX();
App->camera->Position.y = player1->vehicle->vehicle->getChassisWorldTransform().getOrigin().getY() + 5 * player1->vehicle->vehicle->getUpAxis();
App->camera->Position.z = player1->vehicle->vehicle->getChassisWorldTransform().getOrigin().getZ() - 10 * player1->vehicle->vehicle->getForwardVector().getZ();

float player1_x = player1->vehicle->vehicle->getChassisWorldTransform().getOrigin().getX() + 10 * player1->vehicle->vehicle->getForwardVector().getX();
float player1_z = player1->vehicle->vehicle->getChassisWorldTransform().getOrigin().getZ() + 10 * player1->vehicle->vehicle->getForwardVector().getZ();

App->camera->LookAt(vec3(player1_x, 1, player1_z));

//TODO 4 (Assign Camera to Player)

App->camera2->Position.x = player2->vehicle->vehicle->getChassisWorldTransform().getOrigin().getX() - 10 * player2->vehicle->vehicle->getForwardVector().getX();
App->camera2->Position.y = player2->vehicle->vehicle->getChassisWorldTransform().getOrigin().getY() + 5 * player2->vehicle->vehicle->getUpAxis();
App->camera2->Position.z = player2->vehicle->vehicle->getChassisWorldTransform().getOrigin().getZ() - 10 * player2->vehicle->vehicle->getForwardVector().getZ();

float player2_x = player2->vehicle->vehicle->getChassisWorldTransform().getOrigin().getX() + 10 * player2->vehicle->vehicle->getForwardVector().getX();
float player2_z = player2->vehicle->vehicle->getChassisWorldTransform().getOrigin().getZ() + 10 * player2->vehicle->vehicle->getForwardVector().getZ();

App->camera2->LookAt(vec3(player2_x, 1, player2_z));

if (endGame && !start)
    resetLevel();
else if(!endGame && start)
    startRound();

char title[80];
sprintf_s(title, "Radio Control League");
App->window->SetTitle(title);

return UPDATE_CONTINUE;
```