E7: View Matrix

Course: IGME 309 – Real Time Simulations for Games II

Golisano College of Computing and Information Sciences

School of Interactive Games and Media

Rochester Institute of Technology

Due: Check in MyCourses

Deliverable: AppClass.cpp file (single file, unzipped)

**Objective:**

The objective of this exercise is to help students understand how to manipulate the View Matrix of a camera to achieve different views of the same scene. By adjusting parameters such as the camera's position, target, and upward vectors, students will gain practical experience with camera transformations in 3D graphics.

Through this exercise, students will:

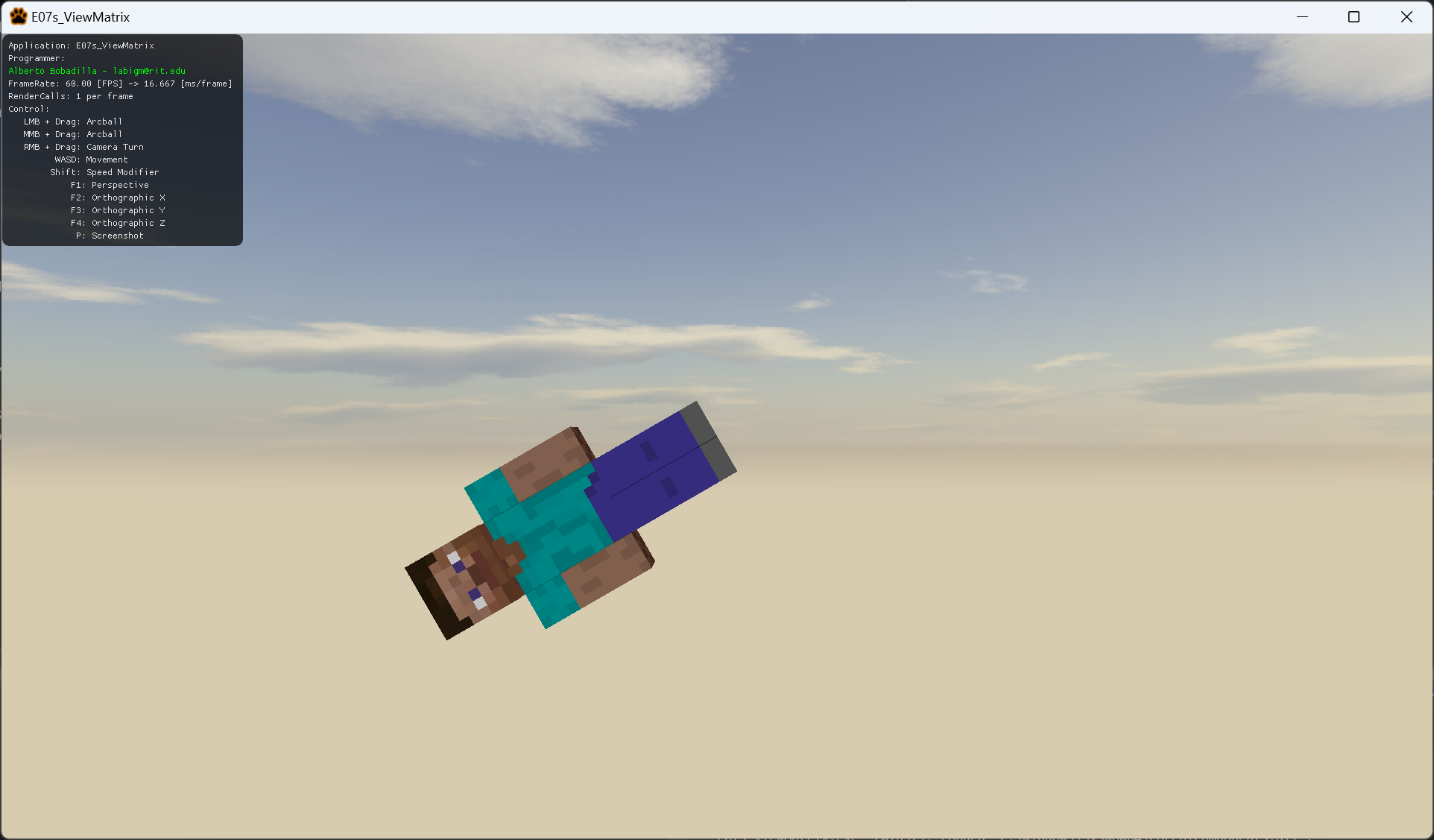
* Learn how to construct and manipulate the View Matrix to transform the camera's perspective on the scene.
* Understand the roles of the camera's Position, Target, and Upward Vector in defining the orientation and location of the camera in 3D space.
* Gain hands-on experience with the mathematical relationships between the camera's view and the scene's coordinate system.
* Explore the concept of "viewing transformation" and how the camera's position and direction affect the rendered image.
* Understand how altering the camera's parameters (Position, Target, Upward Vector) influences the final perspective and the apparent position of objects in the scene.
* Develop a deeper understanding of how to compute and implement the camera's "look-at" function, which is critical for controlling the viewpoint in 3D environments.

This exercise will allow students to experiment with and visualize how various changes to the camera's configuration impact the scene and its rendering, enabling them to build the foundational knowledge needed for creating dynamic and interactive 3D scenes.

**Instructions:**

This exercise follows lecture D7

1. Under \_Binary look for the example solution. It will look like this:



1. For this exercise you will recreate what happened on the example project. This means a couple of things:
   1. When the camera gets closer to Steve he will look larger.
   2. Once the camera gets past Steve it will not look at him anymore.
   3. Steve will not rotate but the camera will.
2. You are not allowed to modify the Model matrix or the Projection Matrix, only the View Matrix.
   1. The skybox is not using the same matrix, if it was connected the sky would rotate with Steve, but a couple of people could get dizzy, so the functionality is disconnected.
3. All your code will be coded in the AppClass.cpp file in said function so this is the only file you need to submit to the dropbox in MyCourses, please do not zip this file.

