E8: Projection 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 manipulating the Projection Matrix of the camera can drastically change the way the same objects are viewed, even when the camera is in the same position. By adjusting parameters such as the Field of View (FOV), Near and Far Planes, and choosing between Orthographic and Perspective projections, students will learn how to control and fine-tune the visual representation of a 3D scene.

Through this exercise, students will:

* Learn how to construct and manipulate the Projection Matrix to control the camera’s view of the scene.
* Understand the effects of Field of View (FOV) on the perceived depth and size of objects in the scene, particularly how it affects the "zoom" and spatial perception.
* Explore the concepts of Near and Far Planes, and how adjusting these parameters influences the range of visible objects in the scene.
* Understand the difference between Orthographic Projection (which preserves the size and proportions of objects regardless of distance) and Perspective Projection (which creates the illusion of depth and objects shrinking as they move farther away from the camera).
* Gain hands-on experience with adjusting projection parameters to create different visual effects and views of the same scene, including zooming, altering depth perception, and switching between orthographic and perspective views.
* Experiment with how small changes in projection settings can significantly alter the rendering of a scene, particularly when visualizing 3D models or designing camera movements.

This exercise will provide students with a deeper understanding of how projection matrices are used in 3D graphics to control the visual experience, helping them appreciate the critical role of camera configuration in rendering scenes from different perspectives.

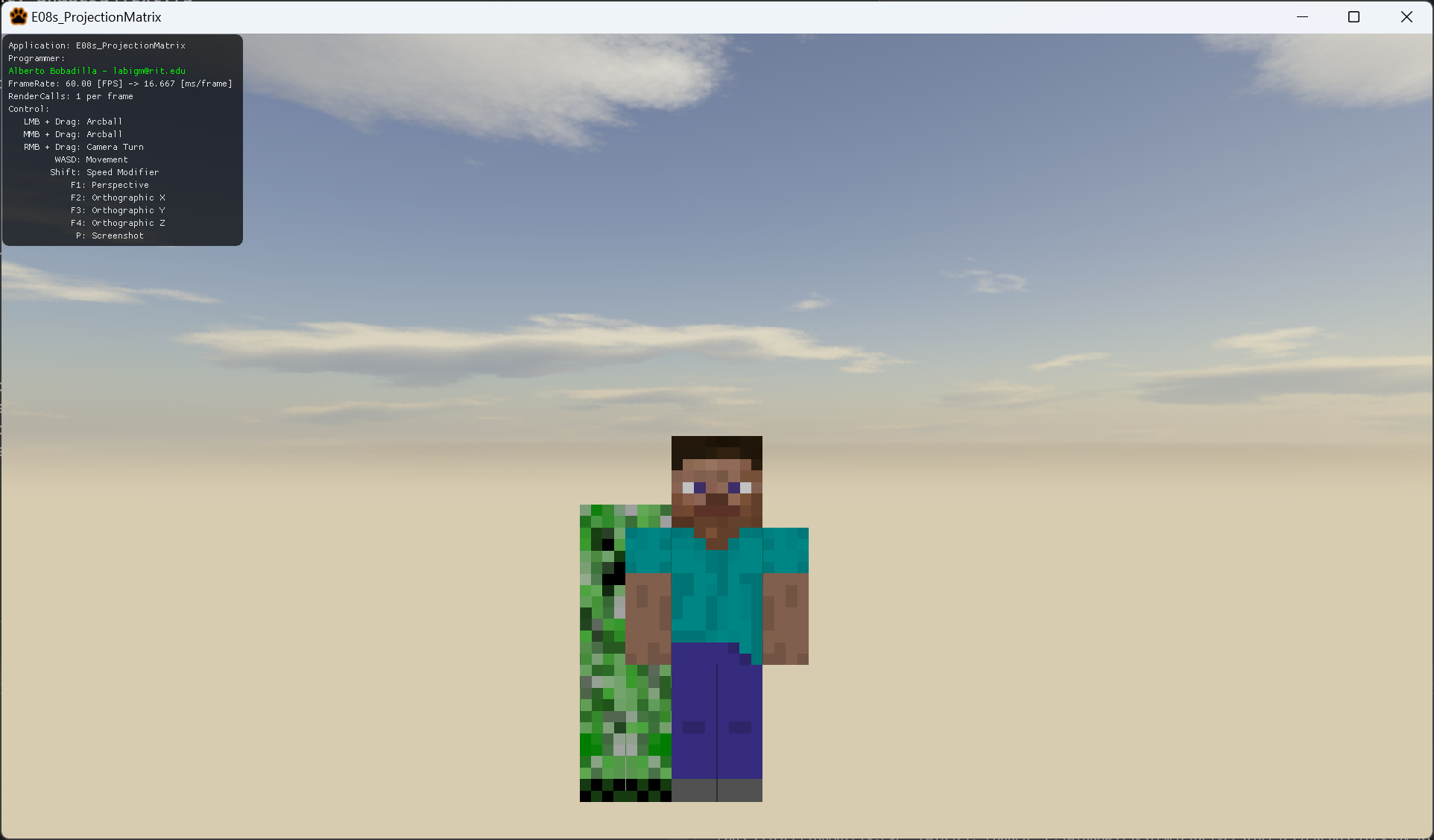
**Instructions:**

This exercise follows lecture D8

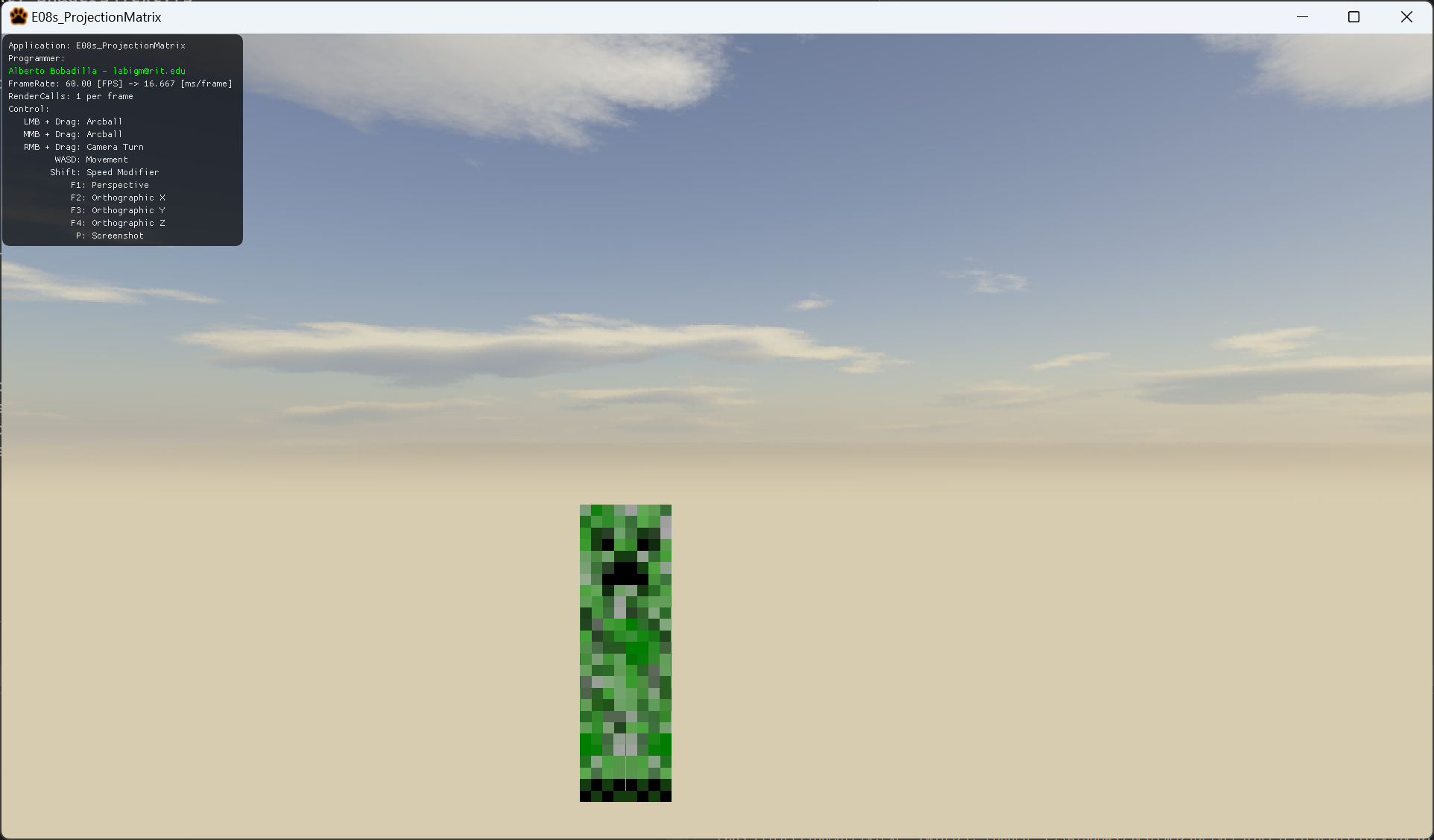
1. Under \_Binary look for the example solution. It will look like this:



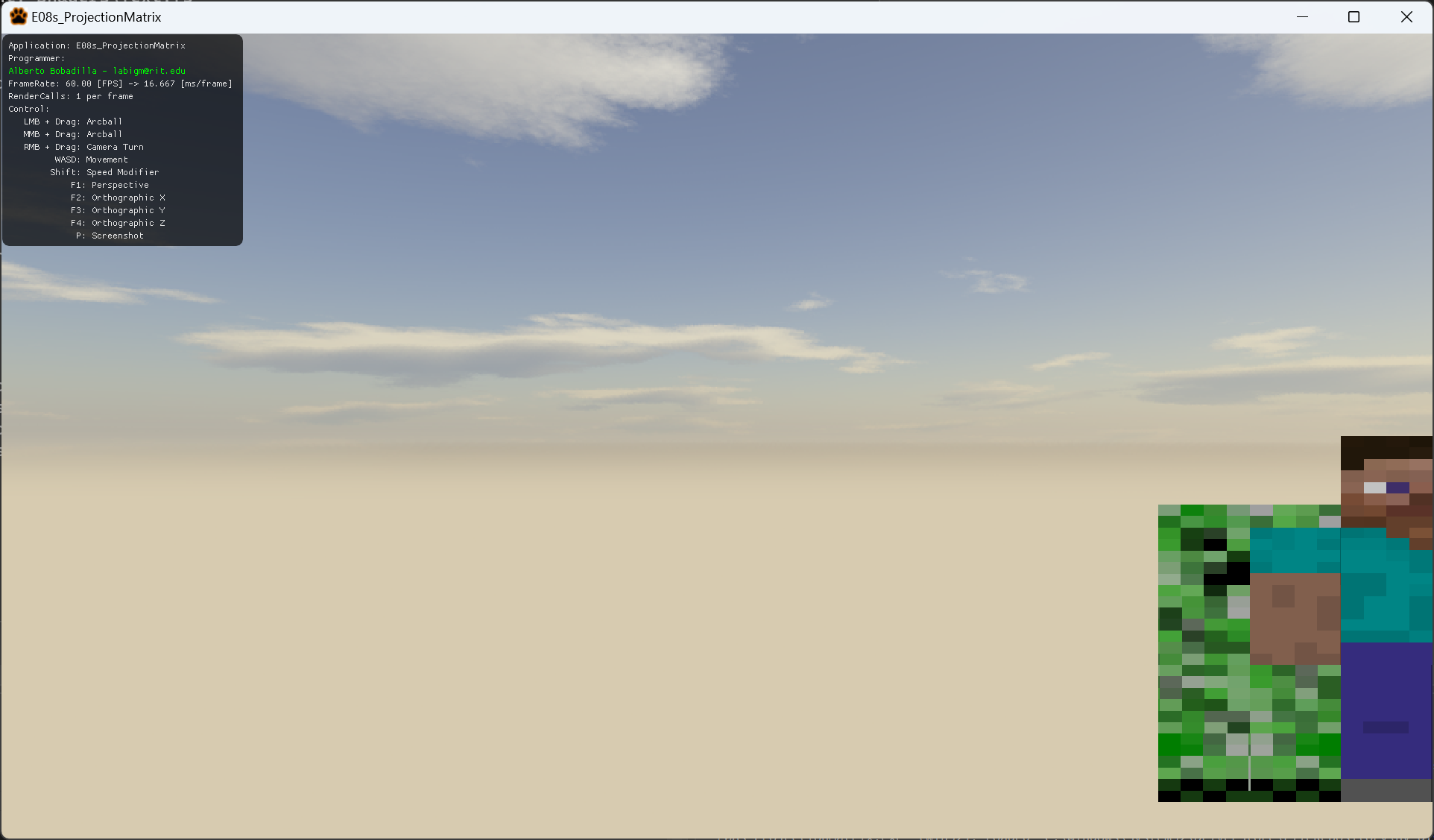
When you press buttons 0 to 1 the scene changes as follows (there is a small description as well on the AppClass.cpp file):

1 

2



3

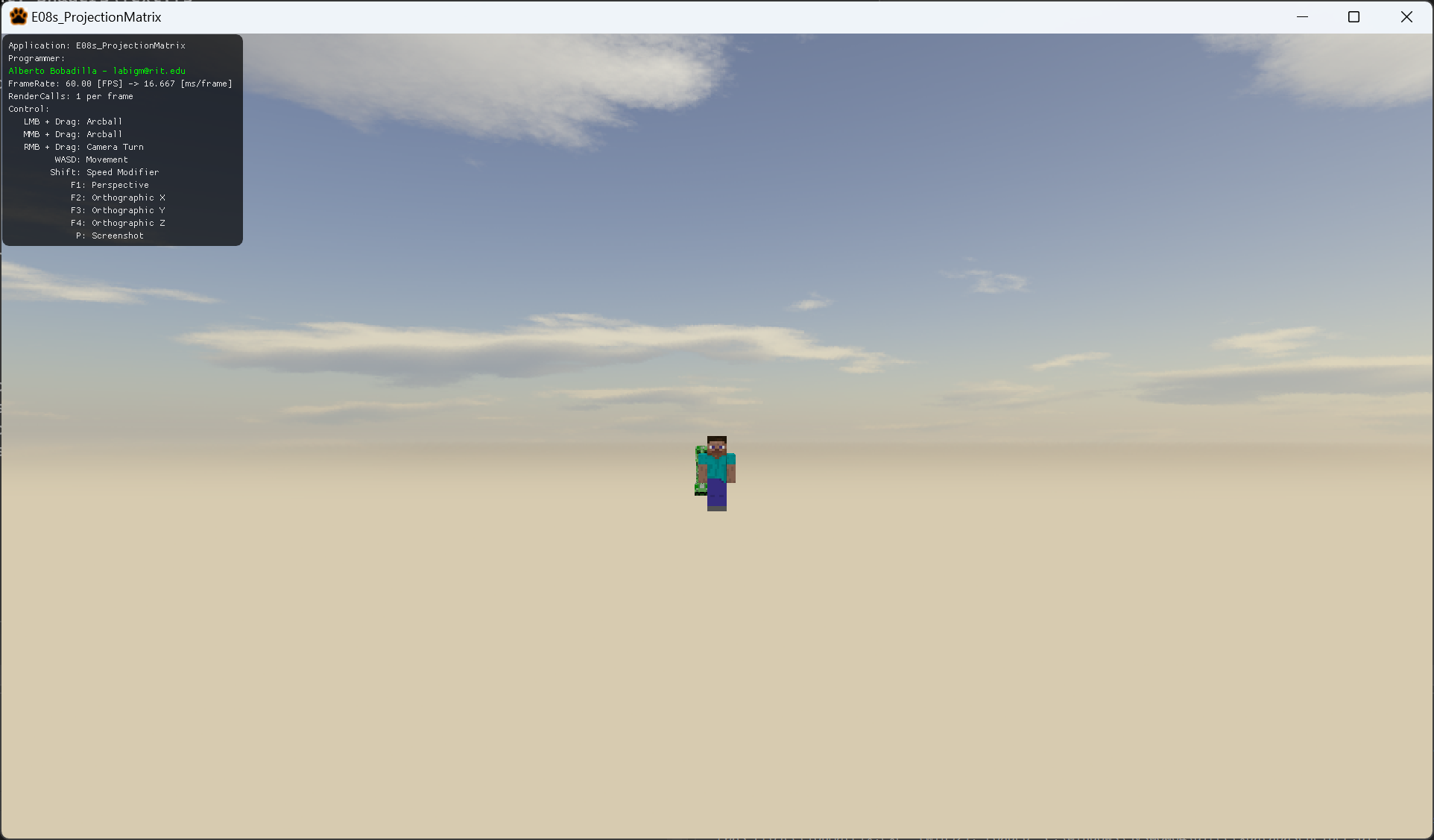
4

5

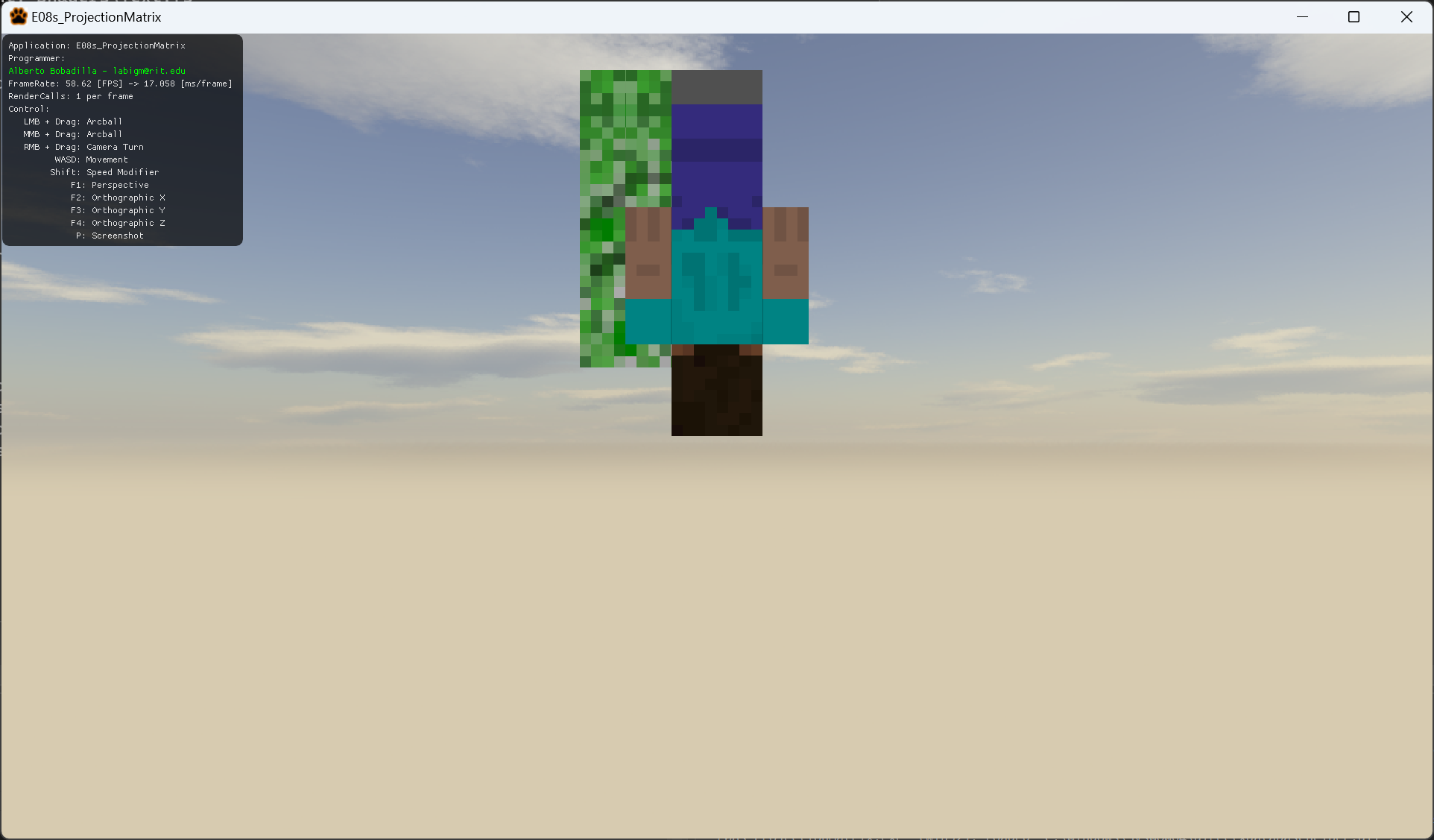
6

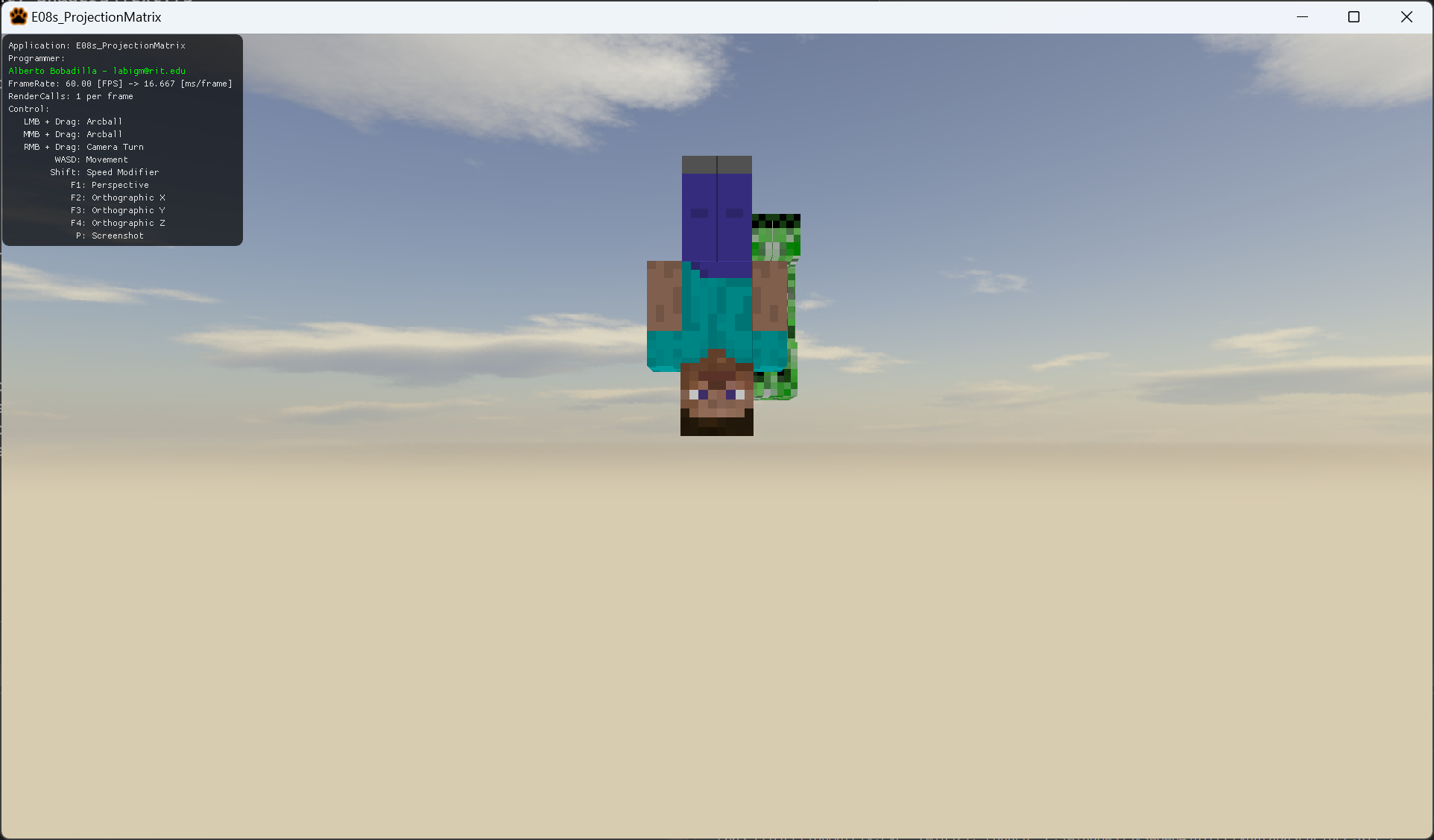


7



8



9

1. Your exercise is to replicate this scenes as close as possible (I do not care about exact numbers so long they resemble the scenes).
   1. You are only allowed to work in the AppClass.cpp file, under the switch statement in Update
   2. You are not allowed to change the camera’s position in any way. Only the projection matrix components.
   3. Everything is setup for you in the AppClassControls.cpp but you may take a look at how things work for the controllers if you want. There is nothing to modify in there.
2. As all your changes are in the AppClass.cpp please only upload this file to the dropbox

