Assignment 3: Angry Bird Palace

In this assignment, you are invited to step into the role of the master architect in the Kingdom of Angry Birds, a realm where your prowess in computer graphics will be tested. Your primary task is to harness the power of transformation matrices to manipulate objects within this vibrant virtual 3D kingdom. The first part of your royal commission involves employing transformation matrices with precision and skill to position and orient a selection of objects to construct the Angry Bird Palace. In the second part of your quest, you embark on a more creative and prestigious task: constructing a royal garden for the Angry Bird King.

1 Reading

- Course slides (particularly Angry Bird examples for constructing composite matrices)
- Tiger Book Chapter 6: Transformation Matrices
- Tutorial Code: tutorial_transform

2 Starter Code

Make sure to execute 'git pull' from our git repository to ensure you have the most recent updates. After pulling the latest code, make sure to run the setup script to set up the program (for Windows: run `.\scripts\setup.bat`; for Mac or Linux: run `./scripts/setup.sh`). After completing the setup, an OpenGL window should appear, as depicted below. The starter code for Assignment 3 is located in the directory `assignments/a3`, primarily consisting of the C++ file `main.cpp` along with two shaders and a number of mesh objects. The mesh objects for Part II are under the folder `garden`. See Fig. 3 for a complete view of all meshes. Your task involves coding within two specific functions: `Create_Angry_Bird_Palace()` and `Create_Angry_Bird_Garden()`, both contained in `main.cpp`.

3 Implementation Tasks

In this scenario, imagine yourself as the royal architect in the kingdom of Angry Birds, tasked with the grand endeavor of constructing the illustrious Angry Bird Palace and Angry Bird Garden. Your role is pivotal in ensuring that every mesh object - each a building block of this grand palace - is positioned with precision, reflecting its proper **size**, **orientation**, and **location**. As the architect, you start with all objects in their foundational state, set with an identity transform matrix and zero translation. Note that we have normalized all models to unit size. All mesh objects have been initialized and placed with their origins aligned with the world's origin. The unit cube mesh covers all other meshes within its interior.

World Coordinates The Royal Construction Bureau provides the following guidelines regarding the world coordinates you will work with. As depicted in Figure 1, your construction will take place within a 3D world framed by a 20x20 grid on the ground. Each grid cell represents a unit square. The x, y, and z axes, colored red, green, and blue, respectively, originate from the center and extend five units each. The camera stands at some point in the positive z-axis and looks at the origin. The code for drawing the ground and axes is located in `Add_Coord()` and `Add_Ground()`.

Initially, all objects are placed at the origin without any linear transformations or translations. They default to their unit size, often concealed within the standard unit cube (if you adjust the camera and zoom in, these models are visible within this cube).

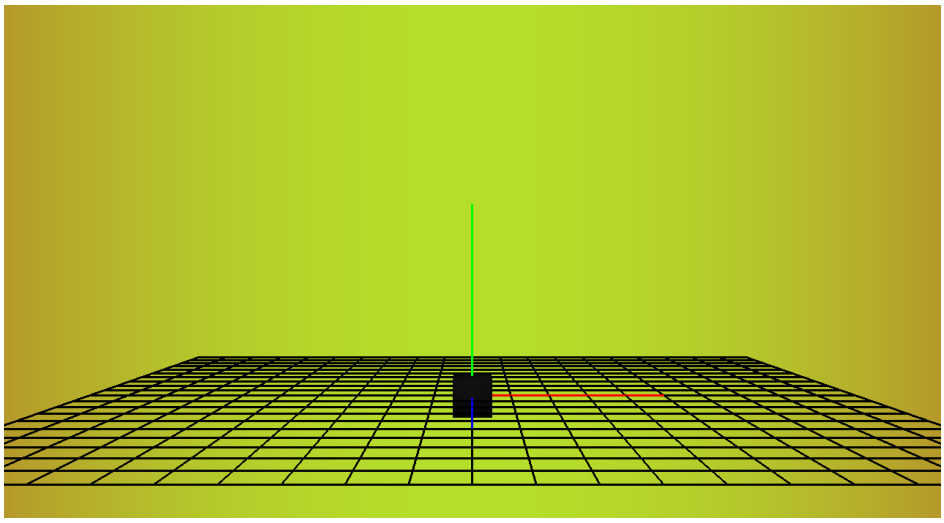

Figure 1: The initial world coordinate system.

3.1 Angry Bird Palace

Your first task to build the Angry Bird Palace unfolds in seven steps. In every step, your primary duty is to craft and apply a 4x4 transformation matrix for each mesh object. As the kingdom's trusted architect, you must exercise precision in the order of operations, ensuring that the palace's layout resonates with the regal elegance depicted in the provided pictures (see Figure 2). While you have the creative liberty to bestow a unique color upon each mesh object, enhancing the palace's aesthetic appeal and adhering strictly to the coding edicts regarding their transformation is of utmost importance. Detailed instructions for each mesh object are in the starter code comments below.

In your implementation, you might want to be extra careful with setting up a rotation around the Y axis (e.g., for Steps 1, 2, and 3). We give you the matrix formula for conducting a counterclockwise rotation around Y with angle θ as below (notice its difference from the rotation around X and Z; also check our course slides for details).

$$Rotation\_Y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

Step 1: Add the Castle Load the `"castle.obj"` model and apply a series of transformations: (1) rotate 90 degrees **counterclockwise** around the y-axis, (2) scale uniformly by a factor of 5, and (3) translate upwards by 1.3 units in the y-direction. Implement the transformation matrix.

Step 2: Add the Axes Statue Load the `"axes.obj"` model and perform transformations: (1) rotate 90 degrees **counterclockwise** around the y-axis, (2) scale uniformly by a factor of 2, and (3) translate 6 units in the positive x-direction and 1 unit in the y-direction.

Step 3: Add the Magic Tower Load `"tower.obj"` and apply the following transformations: (1) rotate 45 degrees **clockwise** around the y-axis, (2) apply non-uniform scaling (2, 6, 2) in the x, y, z directions, and (3) translate 6 units in the negative x direction and 3 units in the y direction.

Step 4: Add Trees Along Inner Circle Place 24 trees, using `"tree1.obj"`, evenly around an inner circle with a radius of 8, centered at the origin. Translate each tree 0.5 units in the positive y-axis. Calculate and apply the appropriate transformation matrices for each tree.

Step 5: Add Trees Along Outer Circle Distribute 36 trees, using `"tree2.obj"`, around an outer circle with a radius of 10. Translate each tree 0.5 units in the positive y-axis.

Step 6: Add Stone Steps Create five stone steps using the `Add_Cube()` function. We provide you with the `Add_Cube()` function, which will add a unit cube with its center in origin (i.e., bottom-left corner (-0.5, -0.5, -0.5) and top-right corner (0.5, 0.5, 0.5)). In your construction, each stone step should a cuboid with dimensions (1, 0.1, 0.5) along the x, y, z axes. Put the center of the first stone step at (0, 0, 3) and each subsequent stone step 1 unit further in the z-direction.

Step 7: Add Keyframes of a Throwing Angry Bird Create six keyframes of an angry bird following a parabolic trajectory. Simulate a throw from (-5, 0, 0) with initial velocity (5, 9.8, 0) units/sec and angular velocity of 150 **degrees**/sec **clockwise**. Consider gravity $\boldsymbol{g}$ at 9.8 units/sec^2 in the negative y direction. Draw keyframes at times (0.2, 0.5, 0.8, 1.1, 1.4, 1.7 seconds). Calculate and apply the transformation matrix for each keyframe. According to Newton's second law, the parabolic trajectory is given by:

$$\begin{aligned} \boldsymbol{x} &= \boldsymbol{x}_0 + \boldsymbol{u}_x \cdot t, \\ \boldsymbol{y} &= \boldsymbol{y}_0 + \boldsymbol{u}_y \cdot t - 0.5 \cdot g \cdot t^2, \\ \boldsymbol{z} &= \boldsymbol{z}_0 + \boldsymbol{u}_z \cdot t. \end{aligned} \tag{2}$$

The orientation of the angry bird is specified by its rotated angle along the z-axis, calculated as $\Theta = \omega t$, with ω as the angular velocity.

The Final Construction After finishing these steps, you should be able to construct the following scene as shown in Figure 2. We took a few snapshots from different camera angles for you to calibrate your own implementation.

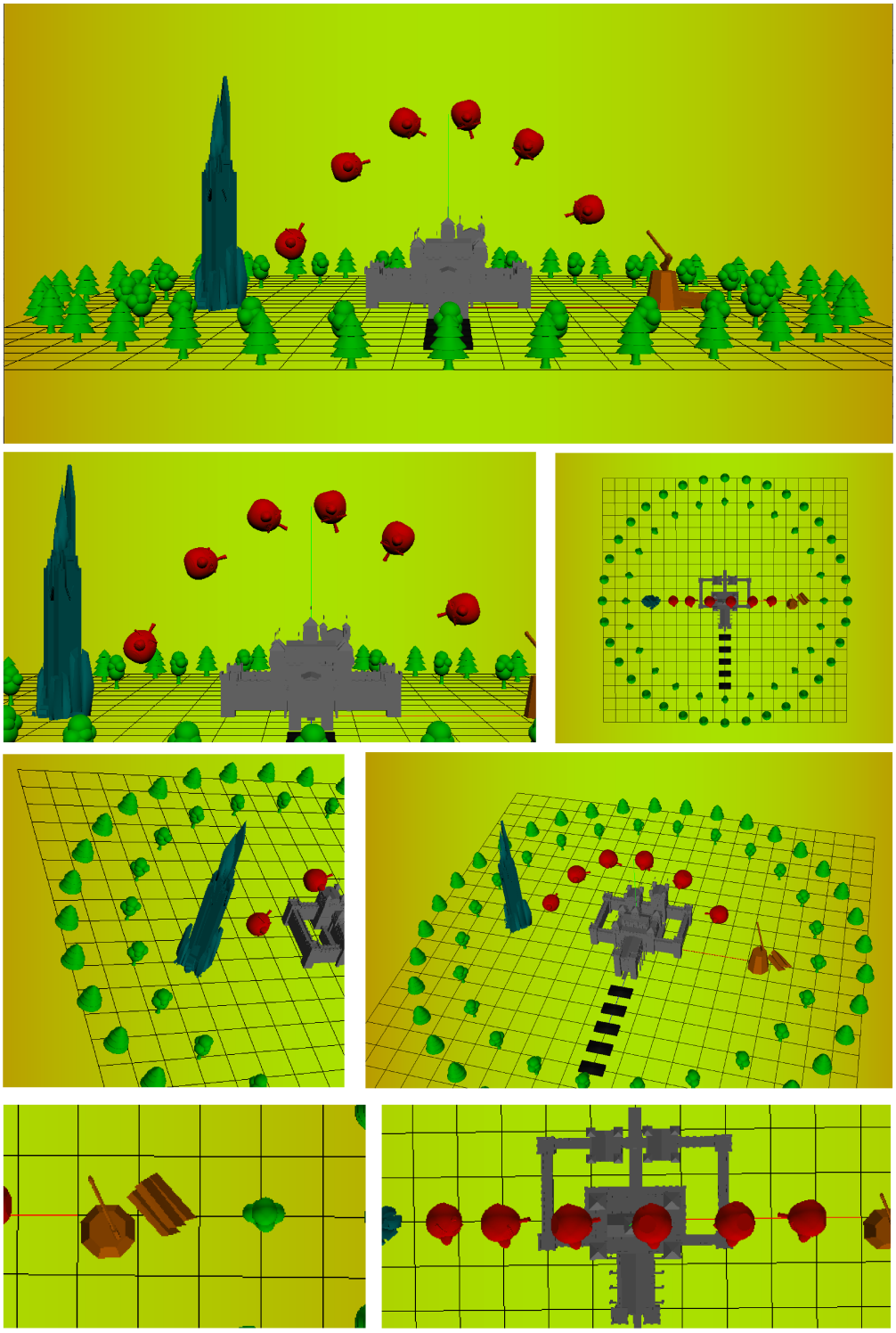

Figure 2: The final construction of the Angry Bird Palace from different camera angles.

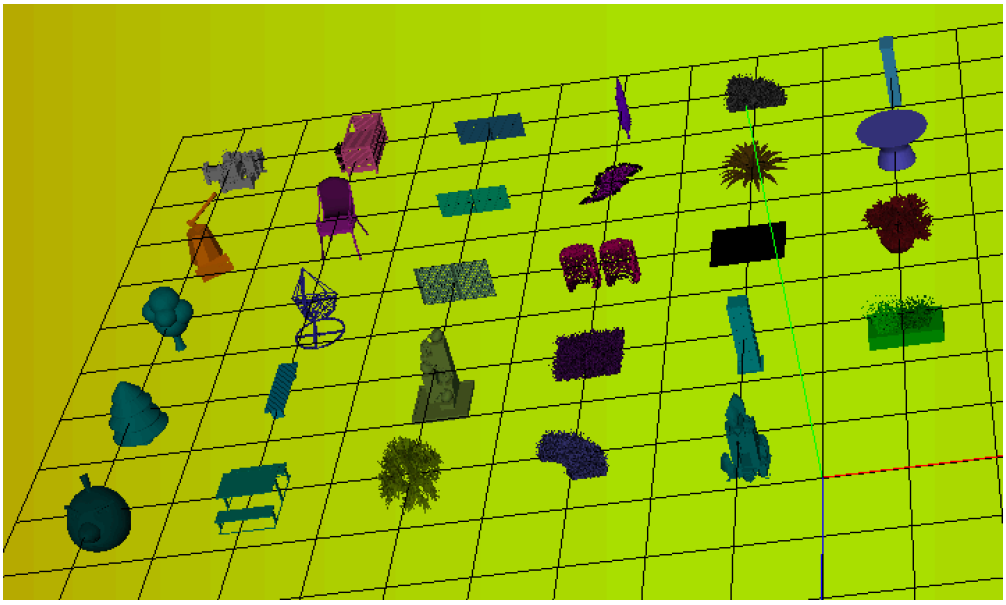

Figure 3: This screenshot depicts a selection of mesh objects provided for garden construction (in the starter code they are all in the origin without transform).

3.2 Create Your Own Scene

Step 8: Construct the Angry Bird Garden In this task, you are entrusted with creating a royal garden for the Kingdom of Angry Bird. Utilize the provided mesh objects or source others online to design this serene landscape. We have a set of mesh objects you might want to use to construct your garden. These mesh files can be read from the folder `garden` under `a3`. **You are very welcome to download other mesh objects online and add them to your scene to enrich its visual appearance.** During your construction, you may mimic the code for initializing and placing each mesh object and apply matrix transformations to them skillfully, as you have learned in the `Create_Angry_Bird_Palace()` function. This step requires custom implementation, focusing on crafting a garden that demonstrates your technical abilities and creative landscape design vision.

4 Submission

Submit the following components for evaluation:

- Your source code `main.cpp`;
- A video or a few screenshots from different camera angles demonstrating the correctness of your implementation for Steps 1-7;
- A video or a few screenshots from different camera angles demonstrating your customized 3D garden scene;

- A concise paragraph that provides a technical explanation of your implementation for the customized scene.

5 Grading

This assignment is worth a total of 8 points, with the grading criteria outlined as follows:

1. Technical contribution (6 pts): The core of the grading is based on the correct implementation of matrix transformations on different mesh objects. The distribution of points is as follows:
   - **Step 1-3:** 2 points
   - **Step 4-5:** 1 point
   - **Step 6:** 1 point
   - **Step 7:** 2 points
2. Creative expression (**Step 8**) (2 pts): This aspect focuses on your ability to create a new garden scene and effectively apply the matrix transformations to enhance its visual complexity.

6 Sharing your Work

If you want to share your graphical work with the class, please upload your image to the Ed post **A3 Gallery: Angry Bird Garden**. This is an excellent opportunity to engage with your peers and gain recognition for your work!