CMP405 Documentation

Adam Borek – 2002519

Summary

Here is a comprehensive list of features that have been added to the tool:

* Improved camera and navigation system with a dedicated camera class
* Object selection via mouse picking
* Object highlighting when selected
* Object dragging along three axes (X/Y/Z) using gizmos
* Object creation / deletion
* Object copying, cutting and pasting

Controls

Here is a list of controls for each feature in the tool:

* Camera & general controls:
  + W/A/S/D – move forward, left, right or backward
  + Q/E – move up or down
  + Right click and drag – rotate
  + Mouse wheel – change modes
* Selection mode controls:
  + Left click on object – select
  + Left click on gizmo and drag – object moving on certain axis
  + Left ctrl + C – copy selected object
  + Left ctrl + X – cut selected object
  + Left ctrl + V – paste copied / cut object
  + Z – delete selected object
* Spawning mode controls:
  + Left click – spawn new object

Features

This section will explain each feature’s design philosophy, and how the feature was implemented in the tool.

Camera system

A basic camera system was already implemented in the tool, but it needed some improvements. The improvements are meant to mimic the way the camera works in other popular video game enines. It enables the user to move freely in the scene easily and intuitively.

The rotation of the camera works via mouse input. Each frame, the last mouse position gets compared to the current mouse position. This difference value will determine how much the camera rotates both on the Y and the X axis. Before the rotation is applied, this value is multiplied by the camera’s rotation speed.

The W/A/S/D keys will let the user move in relation to where the camera is facing. Since we have the camera’s forward and right vectors, they can be used to apply offset to the camera’s position when the buttons are pressed. Moving up and down is not relative to where the camera is facing, rather it just moves the camera on the Y axis, no matter where it is facing.

Object selection

Object selection enables the user to focus on a certain object they picked. Object selection enables the user to further manipulate the object in different ways, therefore it was an essential addition to the tool. It is fairly intuitive to use, as the user just has to click on the object they want to select, with the mouse.

For this, first, two vectors are created to represent the mouse’s position on the near and far planes. Then, the tool iterates through the list of objects in the game world. For each object in the loop, its scale, translation and rotations are retrieved. The rotation is converted from Euler angles to a quaternion. A transformation matrix is constructed based on the scale, rotation and translation of the object in the world.

Then, the near and far points are unprojected from 2D screen coordinates to 3D world coordinates using the transformation matrix and the projection and view matrices. A vector that represents the direction from the near point to the far point is computed. It represents the ray along which the picking is performed.

For each mesh in the object’s model, an intersection test is performed between the ray and the mesh’s bounding box. If an intersection is found, the “selectedID” is set to the index of the object in the display list, and is returned.

Object highlighting

Object highlighting makes sure the user can see clearly which object has been selected.

Once an object has been selected, in the render function, a copy of the object is rendered again, in a slightly higher scale, in wireframe mode. This gives the illusion that the object has a highlight around it, making it more visible.

A screenshot of a video game

Description automatically generatedA video game screen shot of a cube with a globe and a sign

Description automatically generated with medium confidence  
Figure 1 – Object without and with highlights

Object dragging

Dragging of objects enables the user to reposition them anywhere on the map. This allows for precise placement of objects, once they have been spawned in the environment, to make sure they end up at the exact place the user has intended.

Three directional arrow models are loaded in when the tool is starting up. These are stored in the gizmo list. These three arrows represent the three axes the user is able to drag the object on. During the rendering process, these arrows are only rendered if an object is selected, and will show up on 3 sides of the object. Gizmo selection works the same way regular object selection works, as explained above. Once a gizmo is selected, via the left mouse button, the user can hold and drag the object along the selected axis. This is done by checking how much the mouse moved on the X and Y axis since the last frame, and moving the object by that amount. One noteworthy thing to mention is that this works if the camera is on the positive side of the object on both the X and the Z axis. If the camera is on the negative side on any of these axes, the distance moved by the object must be multiplied by -1 to correct for the camera’s position. This makes sure the mouse’s movement is aligned with the movement of the object when being dragged.

Object creation / deletion

Creating new objects allows the player to add more objects to the scene. Extra objects that are not needed anymore can be deleted to free up space and clear up the scene.

Since all objects in the displaylist are rendered every frame, all that is needed for creating a new object is adding it to the end of the vector. Once an object has been generated, it’s position needs to be determined. This is done by defining a near and far point as vectors, to represent the mouse’s position on the near and far planes. This is then projected into 3D space, then a new vector is created by subtracting the near point from the far point to represent the direction of the mouse cursor cast into the 3D scene. The new position for the object to be spawned is calculated by adding the aforementioned vector to the near point. This effectively extends the mouse cursor’s cast line, determining the position where the new object will be spawned.

Deleting a selected object consists of removing it from the display list, as it will not be rendered anymore after that.

Object copying / cutting / pasting

Copying, cutting and pasting objects is a standard feature in many game development engines that allow for faster development time by not having to create new objects every time multiple of them need to be added to the scene. The controls for this are standard as well. Using the control, and another key to use these features have been a standard in multitudes of software throughout the years.

When a selected object it copied by the user, it is saved inside an DisplayObject variable, that keeps track of the copied object until a new one takes its place.

When an object is cut, it is first copied, then deleted.

When an object is placed, a position for it is first calculated with the same method as when spawning a new object. It is then placed into the environment.

Conclusion

The implementation of these features has significantly enhanced the tool’s functionality and user experience. While much went right, some challenges were encountered particularly in ensuring smooth object dragging and positioning. In future iterations, addressing these issues would be a priority.

[Link to video demonstration](https://youtu.be/MSxQLRp9dgM)