# COSC363 Report

Jack van Heugten Breurkes — 23859472

## Control functions

* Arrow keys: up and down to move camera forward and back, left and right to rotate the camera.
* Space bar: trigger jump animation
* ‘x’ key: trigger paper plane crash

## Model 1: Parabolic Pool

The parabolic pool model was created using surface levels drawn using GL\_TRIANGLE\_STRIP. The shape of the pool is a paraboloid and is calculated using an equation instead of storing the points in arrays.

INSERT PHOTO OF POOL HERE

## Model 2: Paper airplane

The paper airplane model was created using a GL\_TRIANGLE\_FAN for both the sides and for the back face. The model is animated to fly in circles around the museum using a rotation about the centre of the museum. The plane also moves up and down as it flies in a sinusoidal manner. Pressing the ‘x’ key on the keyboard will cause the plane to crash, activating the particle system to show fire effects.

A close up of a sign

Description automatically generated

## Model 3: Teapot, table and glasses

The teapot, table and glasses exhibit was created using glut built-ins. The table is made up of cubes that have been scaled to form wooden plank shapes. The glasses are made up of cones and a cube for the chute. The teapot has been animated to tip tea into each of the glasses continuously via a repeating sequence of pitch and yaw rotations.

INSERT PHOTO OF SCENE HERE

## Extra feature 1: Billboarding

Cylindrical billboarding (billboarding involving rotation on the x and z axes only) has been implemented by taking an object’s position in world coordinates and using it to calculate a vector from the camera to the object. This vector is then used to calculate the angle by which to rotate the object so that it is facing the camera.

A close up of text on a white background

Description automatically generated

The position of the object in world coordinates has been calculated using the GL\_MODELVIEW\_MATRIX and the camera position. Using the submatrices M and V of the modelview matrix, the position of an object in world coordinates can be found using the equation objPosWC = camera position + MT \* V where MT is the transpose of submatrix M.

A picture containing clock

Description automatically generated

Cylindrical billboarding has been used both for the trees displayed in the scene, and for the flame effects in the particle system.

## Extra feature 2: Particle system

The particle system can be seen by causing the plane to crash by pressing ‘x’ on the keyboard. Particles are generated with a random rise speed, start height, x position and z position. Particles rise based on their speed and return to a height of zero once they reach the max particle height.

A picture containing table

Description automatically generated

## Extra feature 3: Physics-based jump animation

A jump animation has been implemented for the camera which can be activated using the space bar. The player’s vertical velocity is tracked and used to update the player’s y position. After updating the player’s vertical height the vertical velocity is updated using the equation

Where delta time is the time since the last display update in seconds.

## Extra feature 4: Skybox

A skybox has been included in the scene. The skybox is created by rendering a large box centred around the camera and displaying textures on each surface.

## Extra feature 5: Mathematically generated surface

The bowl shape of the “Parabolic Pool” exhibit is generated using

## References

<http://www.lighthouse3d.com/opengl/billboarding/billboardingtut.pdf> for information on how to get an object’s position in world coordinates