

CT3536 Unity Lab 6

Continuing the Asteroids game

From last week, we have:

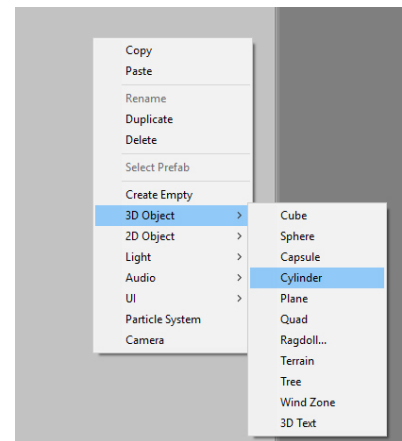
- A game capable of spawning a number of asteroids, depending on what level it's on
- Asteroids move using physics, always constrained to the x/z plane (where $y=0$)
- A player spaceship that moves using physics, with y-axis rotation and engine thrust under control of the keyboard
- On leaving one side of the screen, Asteroids and the player ship 'wrap around' to the opposite side

This week, we are adding:

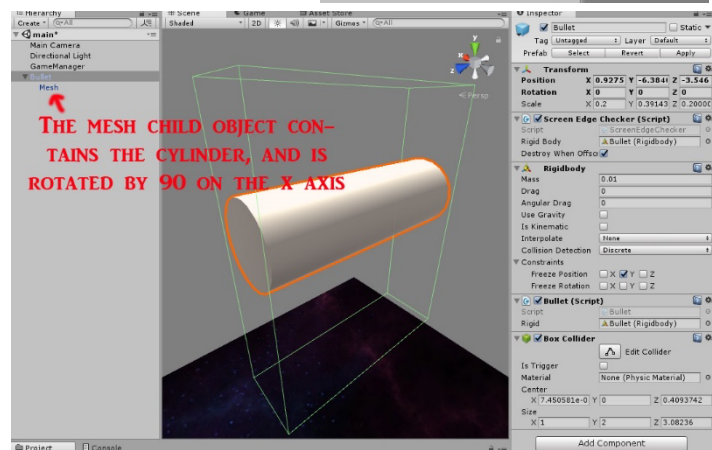
- Firing bullets from the player ship
- Collisions between the player ship and the asteroids: this destroys (and re-creates in the screen centre) the player spaceship
- Collisions between bullets and the asteroids: this destroys the bullet and breaks the asteroids up and/or destroys them, depending on their size

Steps this week:

- Create a new empty object in the hierarchy and call it Bullet
- Create a new 3D object nested inside this, based on a Cylinder.
- Scale the cylinder appropriately and rotate it by 90 on the x-axis. (The default major axis Unity gives a cylinder is the y-axis, so this rotation aligns it as needed for our game). As with the spaceship itself, doing the rotation inside a child object means that we can use its parent as the Bullet and treat the parent's y/z axes in the normal/intuitive way.



- Add a rigid body and a box collider to the Bullet (parent) object, then drag it from the hierarchy to your Assets/Resources folder, to make a prefab from it. Delete it from the hierarchy.
- Add a new script, called Bullet, to your bullet prefab
- Program the Spacebar to instantiate a bullet from the front of the spaceship. Its transform should be positioned and rotated



- appropriately, and its rigidbody should be given an appropriate velocity. Hint: think about the position and rotation of the spaceship at the time the bullet is created.
- When a bullet leaves the screen, destroy it
- Limit to 4 the number of bullets that can be fired per second.
- Program the asteroids' `OnCollisionEnter()` handler to do the following:
 - If the asteroid hits the player's ship, destroy the player ship and re-create it in the centre of the screen (later, we'll implement lives in the game)
 - If the asteroid hits a bullet, destroy the bullet and the asteroid. Create small 'debris' fragments (see last week's lab). If the asteroid was large, create a number of smaller asteroids in its place. (Later, we'll implement scoring in the game).