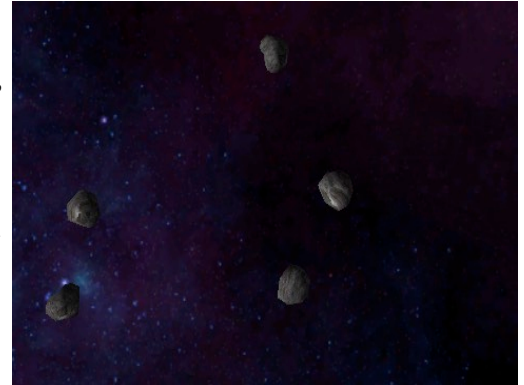


CT3536 Unity Lab 4

Starting a simple Asteroids game

- For the next 4 weeks, we will make a simple overhead-view Asteroids game (similar to the classic 1970s game, except with 3D graphics)
- If you wish to complete this game as your class project, that's ok – you can add extra features, better menus etc. Alternatively, a different game is fine too. The expected standard will be higher if you choose to use the Asteroids game as your base game.



Overview of basic features:

- Game menu
- Level-based game, where each level spawns a number of large asteroids which break into smaller ones on impact with bullets. Small asteroids are destroyed by bullets.
- Higher levels means more asteroids and faster initial speeds.
- Player spaceship can fly around and shoot bullets
- On leaving one side of the screen, Asteroids and spaceship will 'wrap around' to the opposite side
- Use of particle effects for explosions and spaceship engine

Steps this week: get the game set up and have asteroids flying around

1. Create a new game project in Unity. In the Scene create a GameManager object, and attach a new script called "GameManager" to it. Add the asteroid model (from last week), making sure it has a Rigidbody and also a SphereCollider. Put it inside an assets folder named "Resources" so that it can be runtime instantiated.
2. Use one of the skybox images we have been using, as its 'bottom' texture. We won't need any textures on the other 5 sides of the skybox, since the camera will always face downwards.
3. In the GameManager's Start() method set it so that the camera is positioned at 0,30,0 and is facing towards 0,0,0 with 0,0,1 as its 'up' axis (i.e. use the version of Transform.LookAt which defines the camera's "up" axis as well as its "forward" axis)
4. Create a new script called "Asteroid" and attach it to the Asteroid prefab
5. Add an integer member variable called "currentGameLevel" to the GameManager script
6. Create a method called "StartNextLevel" in the GameManager script, and program it to instantiate a set of asteroids towards the edges of the visible screen (the number of asteroids should depend on the value of currentGameLevel). For now, simply call this method from the Start() method. (Later, we will start off with a 'game over' menu rather than immediately being in-game).
7. Program the Start() method in the Asteroid script to set the asteroid moving in a random direction. Make sure the asteroid always has zero for its y value. (Hint: turn off gravity on the asteroid, and set Constraints on the rigid body in the inspector)
8. Have each asteroid check, 5 times per second, whether it has left the edge of the screen. When it leaves the edges of the screen, make it appear on the opposite side, without changing its velocity.