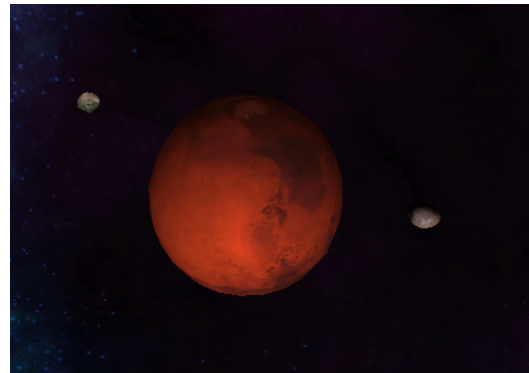


CT3536 Games Programming/Unity3D Lab 2

“Mars has two moons you know!”

Today we're going to animate two moons rotating around Mars, and control the camera using the arrow keys.

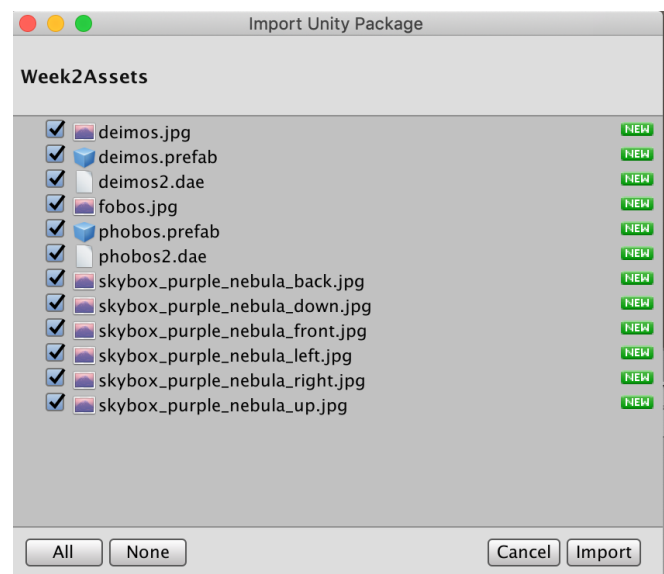
Your work should be submitted on Canvas in advance of next week's lecture. Please paste your code into a Word document. There is no need to submit anything else.



Step 1 – Import new assets

Assets to import: see the file “Week2Assets.unitypackage” file on Canvas

- skybox textures
- phobos model+texture and deimos model+texture
- To import in Unity, use the menus: Assets > Import Package > Custom Package



Step 2 – Skybox

Create a skybox and apply the 6 textures provided (see lecture notes).

Step 3 – Display Mars and position the camera

Program the Start() method of a script attached to a singleton called GameManager (we did this last week):

- Position the camera
- Start Mars rotating

Step 4 – Add Phobos and Deimos in the correct position

Add the Phobos and Deimos prefabs to the scene. Make sure it's the prefabs and not the dae (3d model) files that you add, as the dae files have incorrect scales and other settings wrong!

Edit the values in their Transform components to make sure that Mars is positioned at (0,0,0), Phobos is positioned at (75,0,0), and Deimos is positioned at (100,0,0).

Step 5 – Rotate Phobos and Deimos a little on each frame Update

(We cannot use the physics engine to move the moons, since they do not move in straight lines, and gravity always acts in one direction in Unity.)

Program the Update() method of your GameManager script to rotate the Phobos and Deimos objects around Mars a little on each frame (use e.g. `phobos.transform.RotateAround`).

The script will need references to the objects; the easiest thing to do is add these as Public members (of type `GameObject`) and connect them in the inspector. (As discussed in class, and done in last week's lab for the Mars object).

Step 6 – Bonus! - Control the camera's position using the arrow keys

In the Update() method of your GameManager script, use `Input.GetKey()` to identify whether any of the arrow keys are pressed, and rotate the camera around Mars accordingly (you can again use the `transform.RotateAround` method but you might find it tricky to have things not go wonky after a while.. I'll explain in my solution next week).