Team ANYmation

Scripting Subsystem

Instructions

- This assignment has 4 questions. The points corresponding to each question/sub-part are given in the right side margin. The assignment is for a maximum of **300 p**.
- **You are free to use the internet (obviously) and any other resources**.
- Questions 1 and 2 are based on some interesting phenomemon in astronomy, 3 and 4 are based on open source animation software called Blender (Install the latest version). **There are no prerequisites for this assignment**. If you do not understand some concept or teminology, use internet!
- Questions 1, 2 and 4 have coding component, so, any coding knowledge would be helpful. Coding **must be done in `python`**. You will mainly be needing `Numpy` and `Matplotlib` libraries, but you are free to use any other python library. For those who are not familiar with `python`, here are some tutorials (click). Tutorials 1, 2, 4 and 7 are relevant.
- Please use `jupyter` notebooks or Google colab (`.ipynb` extension) for questions 1 and 2. Please explain the steps and the chosen variable names clearly. Name the files `Q<question number>.ipynb`.
- For question 3 you have to submit a `.blend` file and the rendered images. For question 4 submit a `.blend` file (it should contain the code that is used to generate the animation). **Try not to use any python libraries except for `math` module in question 4**. If you really want to use `Numpy` etc, you will have to follow some additional steps : link. Windows users can try this link. If it does not work, either contact us or try searching on Stackexchange etc or ChatGPT!!
- We will be using Git Classroom for taking the submissions. Create a GitHub account if you do not already have one. Here is the link to the assignment.
- If you have any comments/clarifications/doubts regarding the questions, contact us on our WhatsApp group.

---

1. **Cosmology** 40 p

   Hubbles' law states that a galaxy's recessional velocity v is proportional to the distance d.
   $$v = H_0 d$$
   Where $H_0 = 70$ kms^{-1}/Mpc is called as the Hubble's parameter/constant. However this law is not a strict law, especially for nearby galaxies. Nearby galaxies have a random velocity v_r over the Hubble velocity with an RMS of σ. That is
   $$v = H_0 d + v_r$$
   Assuming that the number density of galaxies is constant in the distance range 1 Mpc to 100 Mpc, plot the velocity (magnitude) vs distance graph for the galaxies.
   Hint: v_r is drawn from a Gaussian with zero mean and SD σ. Also note that v_r can be in any random direction but the v due to Hubble's law is radially outward.

2. **Transit Simulation** 60 p

   Given a bright circle of radius r_1, uniformly emitting light, let another circle of radius r_2 pass in front of it with uniform velocity. The circles start out separate (no overlap) and end separate. The second circle is opaque and does not emit light, hence it causes a drop in total light intensity when it passes in front. Simulate this situation and plot out the light intensity (or logarithm of that, whichever plot looks better) over the whole period of the transit. Note that you can choose arbitrary units for intensity. For plotting choose some reasonable values of r_1 and r_2.

   Hint: Calculating the area via integration will be complex. Instead, try to maintain an array of points over the whole larger sphere and keep track of which points are emitting light at the given timestep. Take the shadow into account by simply switching off the points within the shadow. Let the light intensity be some constant multiplied by the number of points emitting light. As your grid becomes finer, your answer will become more precise.

3. **I love Blender! (and donuts from our orientation ☺)** 100 p

   Complete the blender tutorial till the end of Level 2. Recreate the donut! You can add your own colors etc. You will have to submit a simple rendered image and the accompanying `.blend` file as well as the Texture image file for this part.
   Here are some more resources:

   - Link to subreddit.
   - Blender Shortcuts.
   - Nodes 4 Noobs : The first 3 videos will be more than sufficient for recruitment test.

4. **Two Body Simulation** 100 p

   In this question you have to use python script to make an animation. Have a look at how to write a simple blender script here. We have also attached a sample script (`sample_script.blend`) in Q4 folder in Git, for your reference.
   Now coming to the actual problem: You have to simulate a two body gravitating system using a script in blender. The two bodies can be any simple objects (a simple UV sphere works or you can use the donuts that you made in question 3!). One object (say A) will be stationary at the origin and the other (say B) will revolve around A in an elliptical orbit (choose any reasonable eccentricity). The simulation must follow Keplers laws (especially the 2^{nd} law). Marks will be given on the following parameters:

   (a) Making B go around A in an ellipse of reasonable eccentricity. 50 p

   (b) Using Keplers laws correctly. Velocity variation should be clearly seen. 40 p

   (c) Animation should have the right speed (not too fast, not too slow). 10 p

   If you are **not able** to make elliptical orbit, you can make B go around in a circle around A. However this will receive a maximum of 30 p.
   Hint: Calculate the orbit as a function of time : Euler-Richardson Integrator.