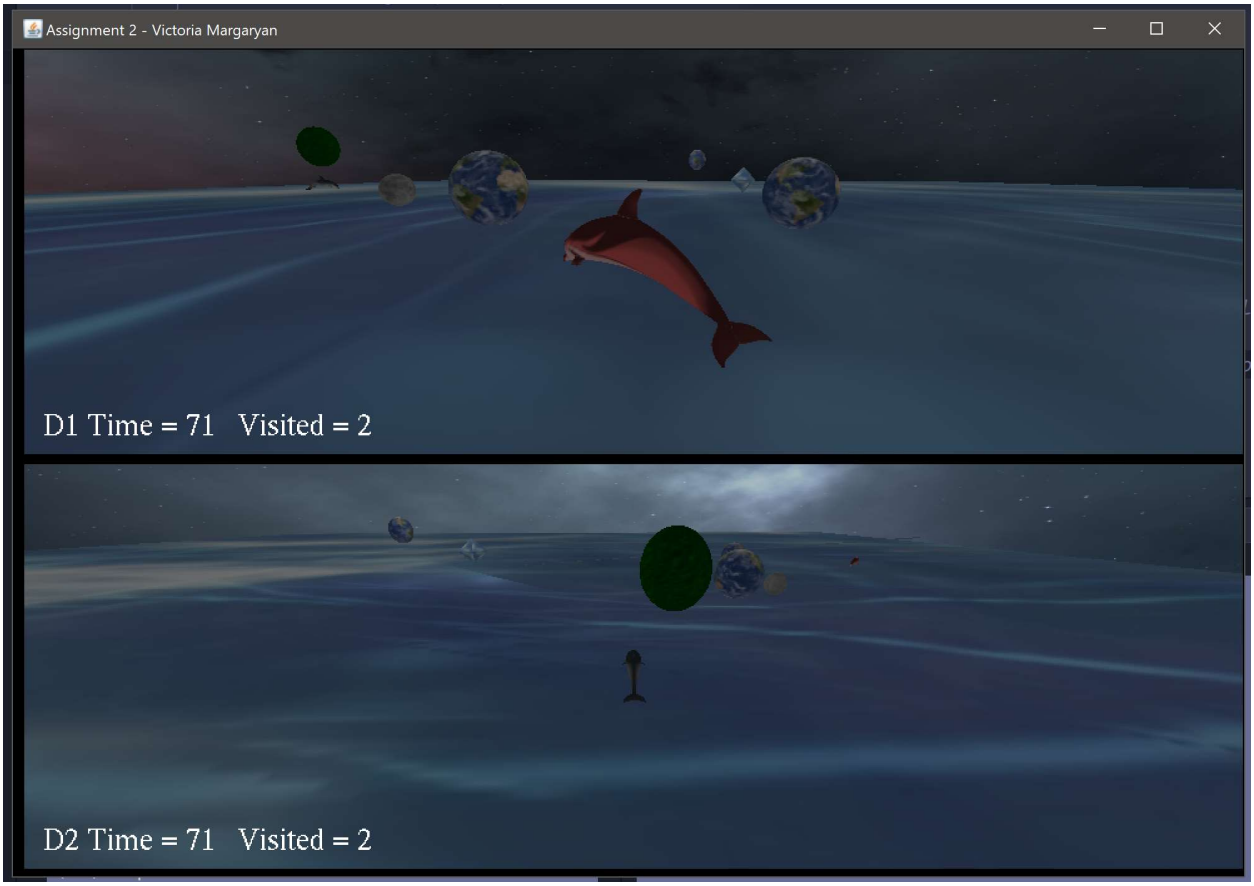
Name: Victoria Margaryan

Screenshot:



How to compile and run program: Run the *compile.bat* file, and then run the *run.bat* file. If you're using VS Code like me, you can run the *vs-script.bat* for user ease (in bash terminals, do ./filename.bat).

How my game is played and scoring:

Don't mind the long load in time. Once you're in, the top player will use a gamepad to control the red dolphin. The bottom player will use a keyboard to control the gray dolphin. Visit a planet and your score will go up! The top player will have their visited planets repeatedly grow and shrink. The bottom player's visited planets will rise, before crashing to the ground and splattering (and it will repeat). You will be able to move your dolphin independently of the camera, but you can't move vertically. That's it!

A list of the inputs and what they do:

The keyboard user controls are:

W	Moves the dolphin forward
A	Moves the dolphin left
S	Moves the dolphin backward
D	Moves the dolphin right
E	Yaws the dolphin right
Q	Yaws the dolphin left
Up arrow	Pitches the camera up
Down arrow	Pitches the camera down
Left arrow	Yaws the camera left
Right arrow	Yaws the camera right
Escape	Quits the game
Z	Zooms the camera in
X	Zooms the camera out

The gamepad controls are:

Left Joystick	Moves the dolphin forward/backward/left/right
Right Joystick	Yaws the camera left/right, pitches the camera up/down
(Z Axis) L & R Triggers	Left trigger zooms the camera in, Right trigger zooms the camera out
Button 2 (B)	Quits the game
Button 5 (Left Bumper)	Yaws the dolphin left
Button 6 (Right Bumper	Yaws the dolphin right

A description of your node controllers and what they do:

I'm using rotation controllers, a Scaling Controller for the first player's visited planets, and then a Custom Controller that translates and scales the second player's visited planets. I had to have the visited planets synchronize on the Custom Controller, because there was a bug where the planets would go below the ground if they were visited when other visited planets were translating down.

A description of your use of group/child node relationship(s):

One of my earth's has a moon. Pretty simple.

A description of your camera control, and whether it is an orbit or chase controller:

I implemented the orbit controller, although the chase controller seems really nice. You can't go too high or too low when doing elevation/pitch controls, it'll stop you. You can go the full length around with the around/yaw controls. You can zoom out pretty far, to see most of the map, but you can't zoom in too close. It will cut you off if you are zooming in/out too close or too far. No issues here.

A clear list of the requirements I wasn't able to get working: All done!

~~3rd person Camera controller (orbit or chase)~~

~~Support split-screen multi-player viewports~~

~~Each player has their own dolphin avatar~~

~~Provide HUDs for each player & their relevant data~~

~~HUDs must maintain relative position if window is resized~~

~~A ground plane (manual object) that is not passable~~

~~Players cannot go up continuously~~

~~Two different types of Node controllers, one written from scratch (extending Abstract Controller). One is applied to planets when they are first visited by player A, the other is applied is first visited by player B. Node controller can cause the planet to spin, or bounce, etc (move in place once visited)~~

~~Players gain points by visiting a planet~~

~~Support two input controllers~~

~~Besides dolphin and camera, one example of a hierarchical relationship between 2+ scenenodes~~

~~Camera should not be allowed to go below the ground plane~~

~~There is no on-off dolphin; you constantly watch the dolphin from a 3P camera~~

~~Probably easier to use "n" mode for the camera controller~~

~~Randomize planet locations~~

A list of anything special I added beyond the requirements:

- Skybox is still working! I had to manipulate the rage.properties file though. I read the readme.md and it said to make a new properties file if you make changes, and then direct the configuration to your new file. The default file has a skybox path that is nonexistent.

Tested on which 5029 lab machine: ECS-myst

A list of every asset used in my game, and whether I made it, along with the legal credit:

- I got the skybox texture from the Unity Asset Store, where it was under the Free license:

https://assetstore.unity.com/packages/2d/textures-materials/sky/starfield-skybox-92717

- I used the dolphin, earth, and pyramid found in the assets folder. I made the green texture and the ice crystal. I used the chain texture from the assets folder, the ice texture from the csc155 book add-on files (thank you professor!). I made the plane manual object.