Project Journal - Team Green

Team members:

Joshua Bender (Team Leader)

Logan Stout (Proxy Leader)

Canyn Black (Documentation - Controls)

Deekshitha Kallem (Documentation - Project Journal)

Dingyi Kang (Asset Manager)

Weekly meetings held on Friday 5:30 pm to 6:30 pm

Week 1

Thursday (09/23)

Had our first meeting. We used this session to figure out everyone's role and general assignments with plans to assign more specific assignments tomorrow. Joshua Bender was chosen to be the team leader because he had the most experience and knowledge with using Unity. During this meeting, Joshua made a small prototype of orbits using Animation Curves and Trail Renderers and explained the concepts to the team. We agreed to have weekly meetings every Friday from 5:30 PM to 6:30 PM.

Friday (09/24)

Today we had a shorter meeting just to assign specific tasks to get done by the end of next week. Joshua was assigned to create the orbits for all 9 planets. He plans to either do them manually by hand using the template he had or make a script that can generate them. Logan was assigned to have distances from the sun for all planets and relative sizes as well. Canyn was assigned to set up OpenXR in the project. Deekshitha was assigned to write a proof-of-concept for the speedup function. Dingyi was assigned to get the textures for the planets and skybox. We also set up a git project in GitHub and added each other as collaborators.

Week 2

Joshua - He wanted to create a script that could generate animations for orbits given their major and semi-major axes, orbital period, speedup ratio, and speedup locations. He went ahead and built a script and tried it out for a few orbits. In those tests, it behaved how he expected it to. He couldn't implement the calculations for the tangents (but had most of the code ready and had to clean up a little), which made it appear weird and caused it to pause at each key, but it did correctly set the values for both curves at each keyframe. He also set up the scene for orbits using some simple placeholder animations so he could easily drag and drop in the animations later once the generation script is completed and get the data of each planet's orbit from Logan later so he can generate them.
Later, he decided to look up the equation for determining the location of planets at any given time and ended up striking a diamond and finding a script written in C that had all the data necessary for all 9 planets already built-in. he went ahead and rewrote it as a C# script, and it worked perfectly. He then proceeded to add in some extra controls for the script, such as a speed multiplier, distance scaler, and multi-object editing support to make it easier to control and animate. All the work he did earlier essentially became useless, so he trashed those files and rebuilt the test scene using the new script. He then set up the animator controller to easily control the speed multiplier for all 9 instances of the script at once using the controller's "Speed" float, binding 0.0 to 1x and 1.0 to 10x. He also made a placeholder animation for planet rotations, which he planned to configure manually.

Logan - Since last week's meeting, he compiled some details about the planets: Their relative sizes, distances from the sun, and how fast each revolution will take around the sun, both with and without the speedup function.
He just used Earth as a baseline for all the data and converted everything to the relative unity units with some basic math. However, since the difference in size/distance from the sun between the planets is stupidly big in some cases, he did have to "break the rules" in order to make everything at least slightly reasonable to each other. Most of this was just playing around with sizes in Unity, it was just monotonous.
He felt that what he has will work, relative to each other. The overall sizes may need adjusted relative to each other, as he didn't know how large the planets will actually be in the VR space.

Canyn - He implemented the Vive headset and its controllers into the project. he based this work on the videos supplied on canvas.

Deekshitha - She has written a speed-up script prototype to speed up the rotation and revolution of planets based on the input taken from the user.

Dingyi - He searched through the internet for appropriate free assets for the sun and the planets of the solar system. After a comparison of different assets, finally, he chose some assets and downloaded them and Customizable Planet Shaders, from Unity Asset Store. He cloned the

project from GitHub and applied those collected textures of 8 planets into the objects in the team project.

Friday (10/01) Meeting:

We also had our official third meeting today, and two people didn't show up because of personal reasons. Nonetheless, we mainly came to the general conclusion that we needed to start working together and combining all our work since the deadline is sneaking upon us. Thankfully, we're still on track for getting everything done. Canyn, Dingyi and Joshua planned to meet in the lab on Tuesday (10/05) to test OpenXR and the Input System and to figure out how to scale everything for the headset. We also caught up on everyone's progress and assigned tasks for next week.

Progress so far:

- Orbits are now completed for all 9 planets and are synchronized to 1 year per 3 minutes.
- The Animator Controller is set up for controlling the speed multiplier.
- Details of the relative size and distances of all planets are acquired, but not yet applied.
- OpenXR is now understood but has yet to be added into the repository and the final scene.
- Input System is not imported or configured for the project yet either.
- All textures and materials for the planets and skybox (excluding the Sun) have been acquired but have yet to be placed into the repository and the final scene.
- The speedup script affects planets but has yet to be added into the repository or be modified to also affect trails.

Assigned Tasks:

Joshua

- Scale the orbits
- Setup the trails
- Scale the planets
- Clean up comments for orbit script

Canyn

- Add the Input System & OpenXR into the repository (by 10/5)
- Test it in VR and work with Joshua to scale everything accordingly
- Create the Controls file

Logan

- Configure the lighting for the Planetarium scene
- Project light from the Sun onto other planets
- Configure ambient lighting using the skybox

Dingyi

- Get the Sun texture and setup its material
- Put materials and texture assets into the repository (by 10/3)
- Create the References file

Deekshitha

- Implement the script for the speedup function once the Input System has been added
    - Use the trigger input to set the Speed parameter of the Root's Animator Controller between 0.0 (1x) and 1.0 (10x).
    - Using the same input, modify the time of all trails within an array to be equivalent to, where speed is between 1.0 and 10.0, and start time is the time the trail has when the program starts.

        "Name" | Type | Values

        "Speed (#)" | Float | Range (1.0 - 10.0)
        "Speed (%)" | Float | Range (0.0 - 1.0)
        "Outlines" | Float | Range (0.0 - 1.0)

        "Speed (#)" is just the actual speedup multiplier itself which is needed for the rotation of individual planets.

        "Speed (%)" is the speedup multiplier bound between 0.0 and 1.0 which is used for the speed of orbits. This should be equivalent to the trigger's analogue input.

        "Outlines" is the float that controls the visibility of the orbit outlines. You should gradually set this value depending on if the trails have been toggled on or off using either touchpad click or some placeholder key.In other words,

        ```
        blend = value <op> (<number> * Time.deltaTime);
        value = Mathf.Lerp(0.0f, 1.0f, blend);
        ```

        Where <op> is + when toggled on and - when toggled off, *value* is the value of "Outlines" and <number> is a positive float, less than one, that controls how much it changes per frame.

- Begin putting together the combined project journal

Week 3

Joshua - He scaled the planets and their orbits using the data given by Logan. He set them according to how high they were on the Y-axis at the peak of their orbit relative to the scene. Afterwards, He modified the OrbitPos script to apply the orbit positions to the local coordinates rather than the global coordinates so that they would rotate with parent objects. He then realized he had no way to tell if any of the orbits were intersecting and that trails would be too slow to outline the orbit even at 10x speed. He then decided to use Line Renderers to outline the orbits by sampling a set number of points on the orbit in the Start function of OrbitPos using fractions of the planet's orbital period as the time. After multiplying the points by the global rotation and by the distance scale, it worked perfectly. He decided to keep the trails and make the orbit outlines toggleable by the Animator Controller. He began working on setting the orbit speeds and planet rotations. He decided to have a single day on Earth be a second long and base everything else on that since otherwise the planets would spin too fast. He also separated the rotation of each planet into separate clips so he could just change them using the speed value in the Animator Controller. He also attempted to implement axial tilts, but the animations kept overriding my values.

Unfortunately, there were several issues that occurred during Tuesday's lab session that they had to resolve. In short, Git Bash left out some files from the commit, and it pushed to a new branch rather than the main branch. Aside from that, the shader Dingyi added for the planets didn't have Single Pass Instanced support for VR, meaning that objects using it didn't appear in both eyes. Joshua had prior experience in writing shaders, so he went ahead and added support for it which had the side effect of breaking everything involved with receiving shadows. He basically had to completely redo shadows in the shader since the method it uses is included in Unity and has no alternative for Single-Pass Instanced rendering.

He then got things set up for making planets grabbable for Canyn. He configured the trails and fixed rendering issues he noticed in the project. Logan contacted him and informed me that the shader being used for the planets didn't work with point lights. Both of them got into a call and worked together to fix it. They mostly just replaced all code regarding the directional light with data from the first point light found in the scene (since there was only one) and that seemed to fix everything. He added comments to the OrbitPos script and removed the custom editor since it was redundant and unnecessary for editing multiple instances at once. He also spoke with Kallem about what she specifically needed to have in the script. They also took the time to understand how to access the controller inputs from the script and use them in the function.

Logan - This week, his goal was to make sure that the lighting was working properly with the planets. Adjusting the brightness and colour of the ambient lighting was easy enough for him. He just increased it enough so one could see the texture of the planets regardless of how much light was on them, but not so much that there was no difference between which side was facing the sun. When he was playing around with point lights, the planets weren't being lit up properly and were still only being affected by the default directional light. After lots of googling and

messing with stuff, he found that it was the shaders that came with the NASA textures that were causing the issue. Originally, he had switched the planets to just using the simple texture files, which solved the issue but made the planets look worse. Luckily, Joshua was able to go in and mess with the shader files and get them to work, so they got to keep the high-quality version. After that, he remembered that they each needed to script something, so he quickly wrote up a few scripts that allowed both the colour and intensity of the light to be changed using the keyboard. This was pretty quick, compared to figuring out the point light issue. It only took a little bit of reading the documentation and looking at the console to figure everything out for him.

Canyn - He had implemented a control system for grabbing celestial bodies in the planetarium. When the grip buttons are pressed, any object in the solar system can be taken from its position and held near the user's hand.

Deekshitha - She tried implementing the speed-up function on the planets added by other team members in the scene. She tried to speed up the planets using the script she had done last week. But she couldn't exactly figure out how the input system worked and how to read the values to the script and it didn't help that she missed last week's meeting. She then had a talk with Joshua and understood what needs to be done on the speed-up script. They looked up the sample scripts provided by OpenXR and understood the process of taking inputs from the controllers. She went ahead and made changes required for the speed-up of the planets but couldn't add changes for outlines of the planet orbits.

Dingyi - He pushed the textures collected for all planets into the repo. He also searched textures for Sun and Pluto on the internet. He learned how to make textures for the sphere through watching tutorials on YouTube. He searched and downloaded the high-solution seamless pictures of Sun and Pluto from the Internet. He made materials for Sun and Pluto based on the downloaded seamless images. He applied the materials made to the Sun and Pluto objects in the team project. He also watched some tutorials on how to create a ring for Saturn in Unity. Used the ring model in the Customizable Planet Shaders asset and added it to the Saturn object in our project. Did quick research and applied the material of the ring of Saturn to the ring in our project.

Tuesday (10/05)

It was lab day. Joshua arrived early and noticed that the only git software on it was Git Bash. So he had to figure out how to use it so he could push any changes we made later. Joshua showed Canyn and Dingyi how to push into GitHub. They attempted to use the headset with Canyn's configuration but had some issues. However, after a bit of debugging and troubleshooting, they fixed it and began to test everything. They basically had to scale everything down by a factor of 10 which looked nice, but they determined they'd really need to implement a form of movement for it to be easier to see everything. Additionally, they had to modify the orbit pos script to apply the global position and scale to the orbit outlines. They also used the time in the lab to correctly apply axial tilts and have everything work with it.

Friday (10/08) Meeting

Caught up on everyone's progress and listed things to get done for the lab session the next day. We also discussed what was left to be done for the entire assignment.

Saturday (10/09)

Today was a longer day than expected, to say the least. Team Red finished a bit early, so Joshua headed over to the lab at around 5:15 PM so that the others could get inside since the doors were locked. He made sure the repository was still fetchable and configured it so that we could both stash changes and commit to the correct branch in Git GUI on the lab computer.

Deekshitha was the second to arrive. She had the speedup script that we committed and pulled to the lab computer to test and work on. It had a few issues to say the least. The most noteworthy issue was the controller inputs flickering between their correct value and 0, and we didn't figure out what was wrong until much later in the night. We also took the time to modify the script to function how Joshua had intended: take the maximum value of both triggers and use that as the speed multiplier.

Logan was next to arrive. Admittedly, he was already done with his portion of the project before arriving at the lab, so he essentially played spectator to the hours of debugging and testing that ensued.

Canyn and Dingyi arrived a bit later at the same time. Deekshitha and Joshua took a break from trying to resolve the problems with the speedup script so that Canyn could pull and test his setup for grabbing planets. He had yet to add sphere colliders to the planets so ray casts would hit them, but after adding those it functioned as expected. The initial controls for it were a bit unintuitive, so we eventually got around to modifying them to behave more naturally. There was also a problem where the planets wouldn't return to their original rotation after grabbing and releasing them. This was another issue we didn't resolve until much later.

While Kallem, Canyn, and Deekshitha were debugging the various scripts, Dingyi took the time to write up the references file for the submission.

At this point, the rest of the night was almost entirely dedicated to correcting the known issues since they were all that remained. Joshua had a part in most of the fixes and alternatives that were implemented. It's worth mentioning that these problems were being worked on concurrently, and the order described below isn't representative of that.

The issue primarily focused on fixing first was the speedup script with Kallem. They spent around an hour in total comparing what they had to articles online and the sample scene that was included with the OpenXR Plugin. They noticed, much later than they should have, that the only significant difference between the sample scene and their setup had nothing to do with the script, but rather the inclusion of another component on the same object, Player Input. Joshua removed the component and tested the controls, and it started to work correctly.

Unfortunately, the component that they removed was being used for toggling the outlines of the orbits by calling a method whenever either touchpad was clicked. They then spent the better half of an additional hour attempting to retain the toggle behaviour before Joshua caved and made the outlines appear only while either touchpad was clicked.

The final issue that they discovered after this was that the trigger clicks and the touchpad clicks change their type from a Boolean to a float whenever they are pressed down. They still have no idea why this was the case, but they just accepted it and changed to the type read to always be a float type, since Booleans in C# can be coerced into floats of either 0 or 1 anyway.

During all this, Joshua was also spending some time with Canyn to help him with his issues as well. The first thing they worked on was making the planets appear in the correct location, relative to the controller when they were grabbed. This was simple enough, but initially they had thought that they would also need to disable the position constraints of the planets to grab them. They eventually realized that this wasn't necessary (for some reason) and discarded the code they had made for it.

They then moved on to try and make the planets return to their original rotation when released. They tried storing their local rotation and restoring it when they were dropped, but this wasn't working. Joshua assumed that the delegate for picking it up triggers a frame after it's grabbed, so it never received or restored the correct values. Regardless, they abandoned this approach and eventually settled on just preventing the planets from being rotatable while held. This worked better, but the planets were still retaining some angular momentum upon release. They eventually noticed that the IsKinematic parameter of the Rigidbody component wasn't being enabled when the object was released, which resulted in it having the momentum to begin with. They asserted that it always stayed enabled, and that resolved the issue.

Before they made the planets scale to a uniform size when held, they took the time to correctly adjust the distances of the planets so that they could all be reached while standing in the play area (during at least two points of their orbits). They recreated the size of the play area in Unity by having a controller placed in either corner of the space, noting down the controller positions, and then scaling a plane to fill the space. In hindsight, this was a much better idea than they had anticipated since we can now reuse this plane in other projects to scale everything without needing to go to the lab.

Afterwards, Canyn went ahead to work on the script to scale the objects being held. Joshua had already setup scale constraints on all the planets so that they could change the size of all held objects at once. However, for whatever reason, the process of changing the weights of constraint sources in C# is either super awkward or something they completely misunderstood, because it wasn't letting them set the weight parameter for the source even though it had a public accessor and setter according to documentation. They then decided that they should just recreate the list of constraint sources using their transform sources and setting the weight manually. This worked, but there is probably an easier and more efficient way to do it than that.

Another problem was that the laser pointers (RayCast) which indicate the direction of the controllers would stay on when a planet was grabbed. RayCast.enabled = false did not work as it was constantly being set to true by something else in the packages they were using. Ultimately, they just set its length to 0 while it was being held and back to ten upon release.

The two final changes they made before leaving were making the planets stop spinning while held and disabling the trails while held. Thankfully, these changes were extremely quick to implement. Joshua added an extra state to each planet's spin layer in the animator that had an empty animation to transition to while being held that was controlled by a Boolean. This Boolean was set in the same script that the constraint sources were modified since those methods were already triggered when a planet is grabbed or released. For the trails, they just dragged in the renderers as an extra event when grabbed or released and set their enabled value through the inspector.

When Joshua and Canyn had everything done, it was already past midnight. Everything related to Unity was complete. They made the final commit of all the progress made at the lab and went home.

Monday (10/11)
Dingyi - He wrote a script of the trail so that the trail will restart instead of just re-appearing when a selected planet is released. He also added the trail script into interactive events of the planets objects.

Joshua - He communicated with Dingyi about the script which Dingyi needs to write. Also, he helped Dingyi with writing and adding the trail script.

Improvements/Learnings:

- Communication between the team could have been improved.
- Distribution of workload.
- Getting things done on time by or prior deadlines.