



COLLABORATION USING GITHUB

-ADDITIONAL TASK



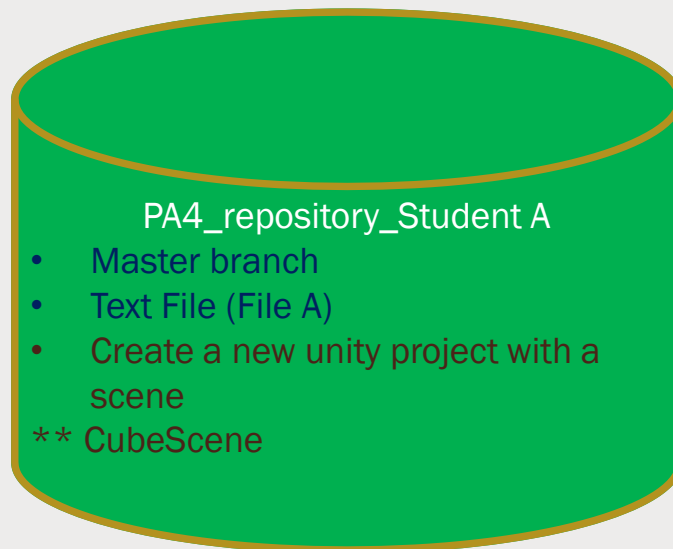
By end of this lab

- Student will understand the basics of
 - *Integration using GITHUB version control*

To complete this lab, you need

- PC
- GitHub Desktop client
- GIT Cloud account

Scenario 1



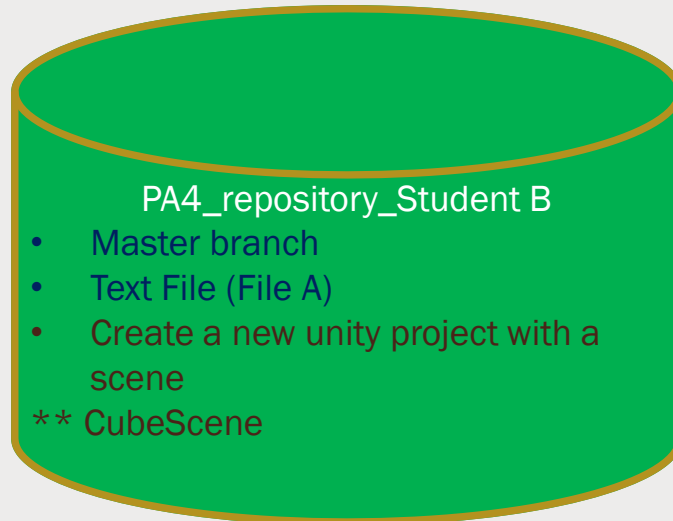
Student A (Owner)

- Open the repository created
- Create the unity project inside the repository
- Give access to Student B
- Merge contents from Version1 (created by student B) to Master

Student B (Collaborator)

- Access to the repository through desktop client
- Create another branch – Version 2
- Update the unity project file with Loading Scene

Scenario 2



Student B (Owner)

- Open the repository created
- Create the unity project inside the repository
- Give access to Student A
- Merge contents from Version1 (created by student A) to Master

Student A (Collaborator)

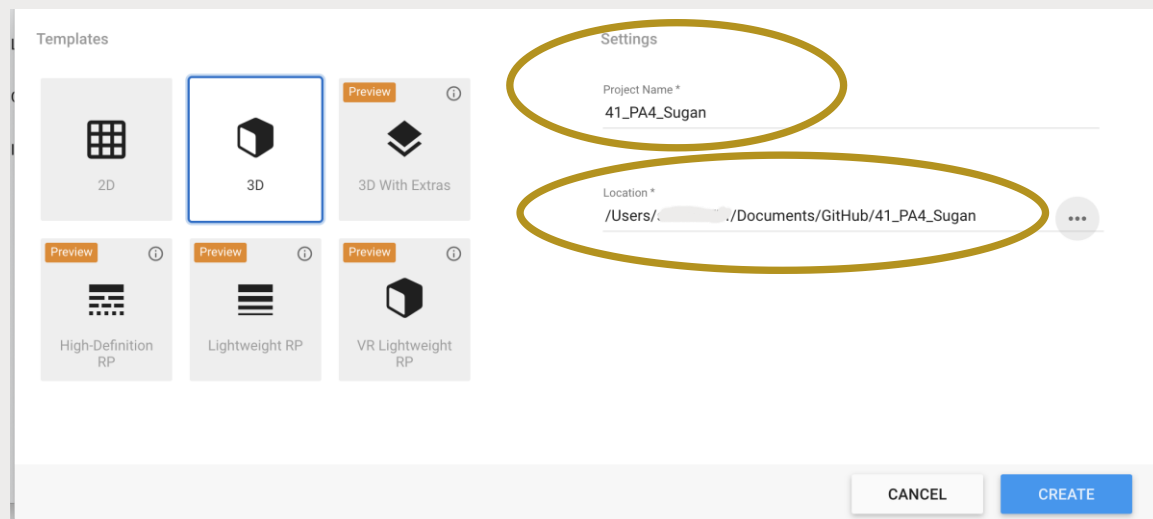
- Access to the repository through desktop client
- Create another branch – Version 2
- Update the unity project file with Loading Scene

Forming a team (2 members)

- Team will be the same as previous PA4 lab

Step 1 : Student A and Student B (Open IndexNumber_PA4_YourName repository)

- Access to GitHub client
- Launch the unity project and save it as inside the repository OR in your own unity projects directory(*Working Directory*). Verify the path
- Save the unity project as IndexNumber_YourName_PA4



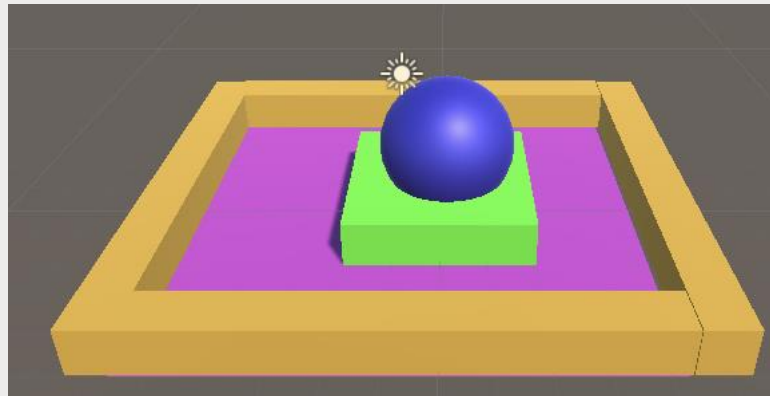
Step 2 : Unity project

- Create three folders:
 - **Materials**
 - **Scenes**
 - **Scripts**
- Create a new scene and save it as **CubeScene**

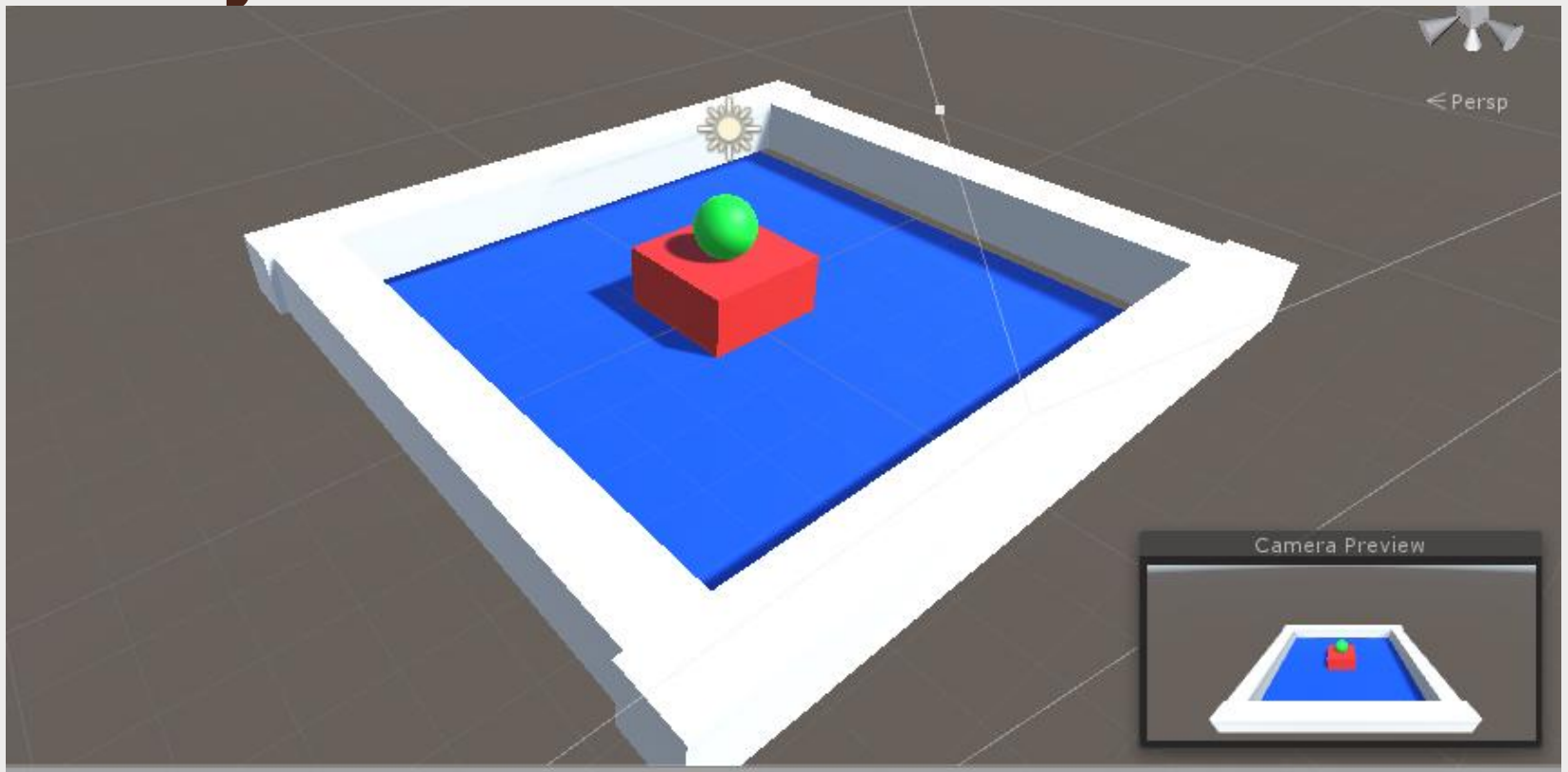
Step 3 : Unity project

- Create a simple environment
- 3D objects used :
 - *Plane*
 - *Cube*
 - *Sphere*

Reference image :



Step 4 : Capture and paste a screenshot below of the environment, you created from Unity

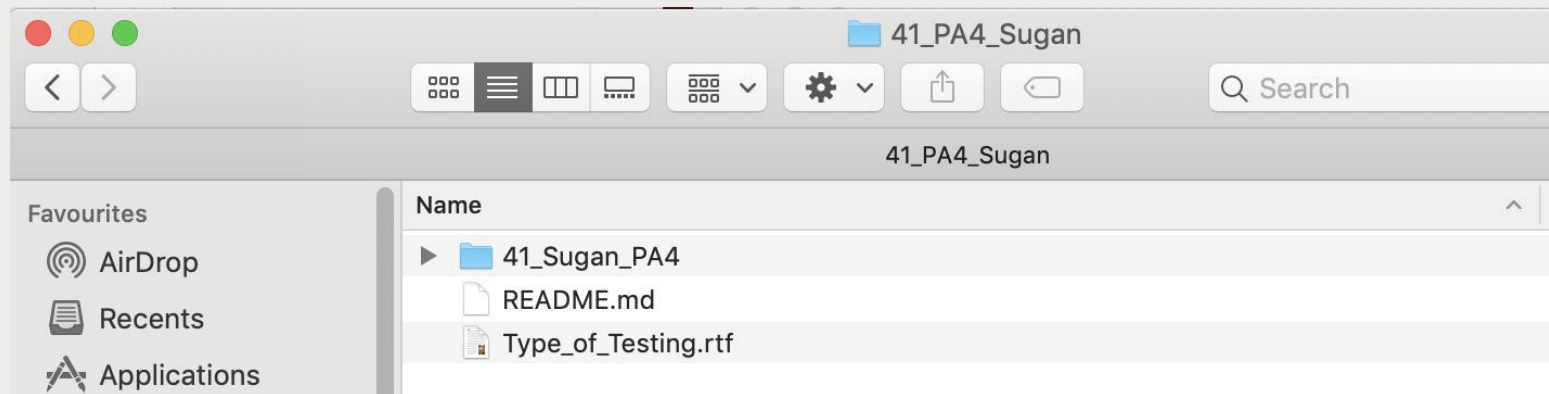


Step 5 : Create a simple C# script

- Create a new C# script to rotate the sphere over the cube.
- Please refer to the demo by your Lecturer

Step 6 : Push the unity project to the repository and publish

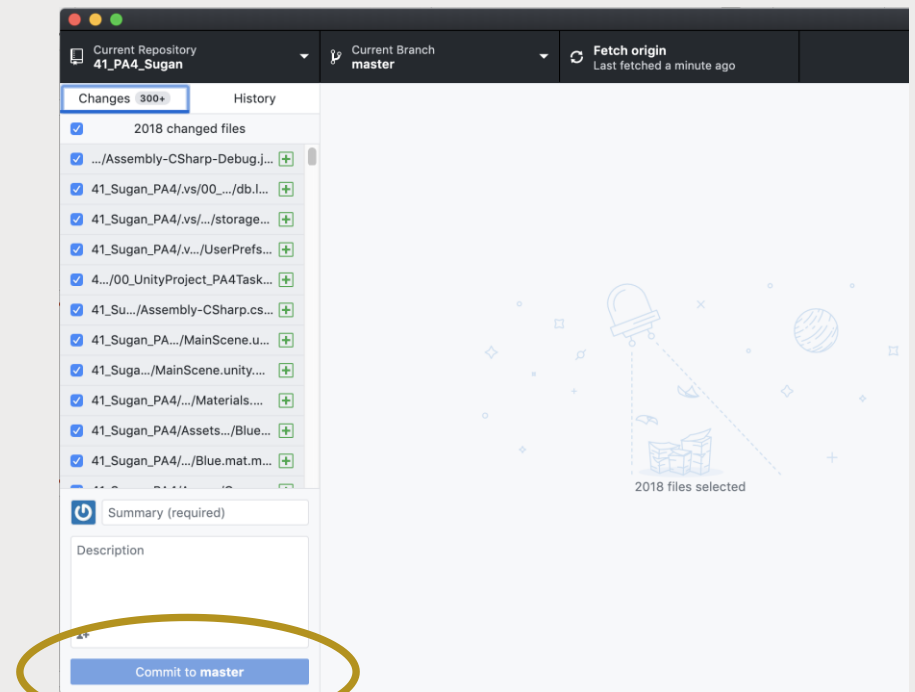
- Move the unity project from working directory to the repository folder.



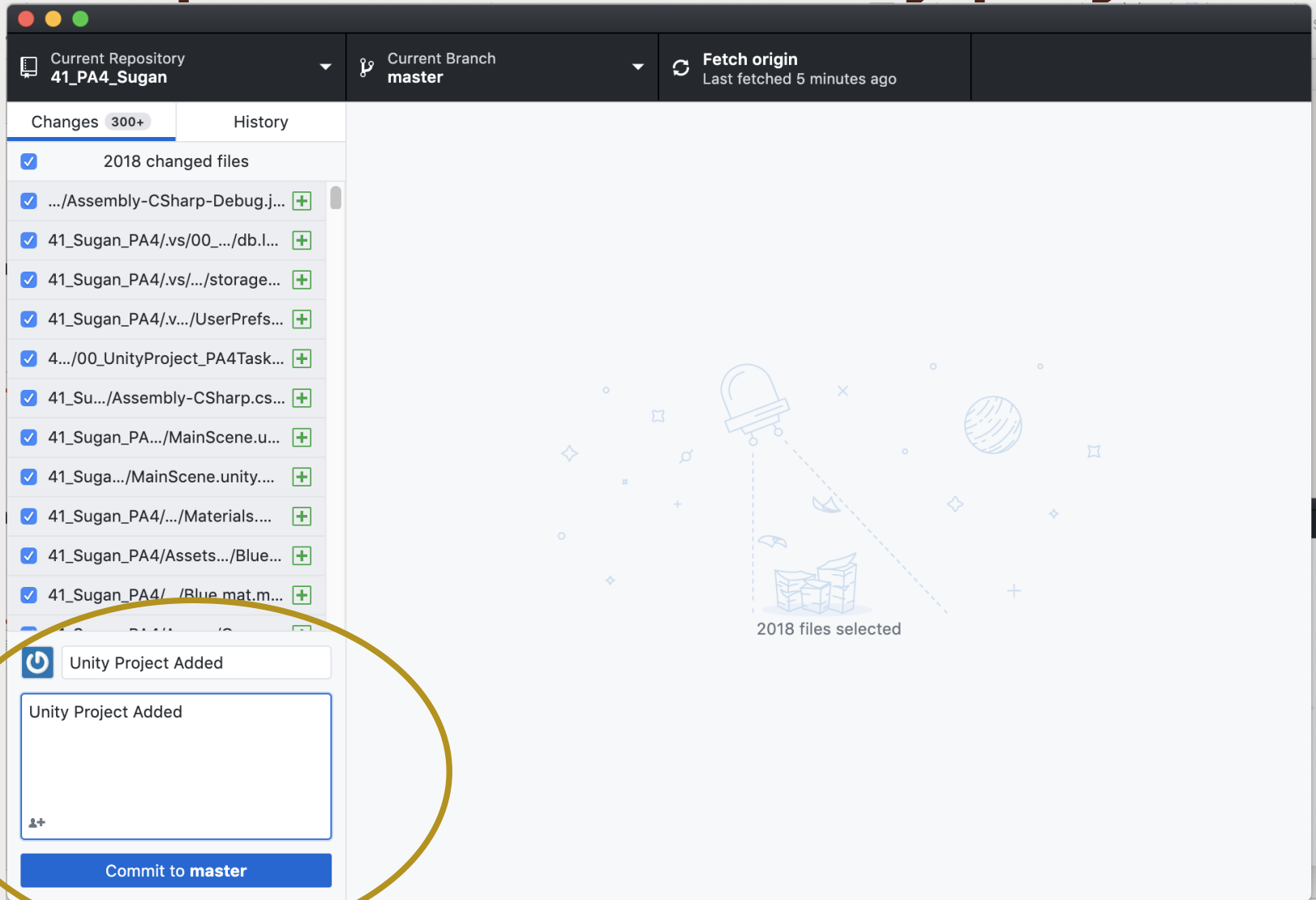
Step 6 : Push the unity project to the repository and publish

- Commit the unity project with a message to Version 1 branch
- Message : Added a unity project
- Push the contents to GitHub cloud
- Create a pull request

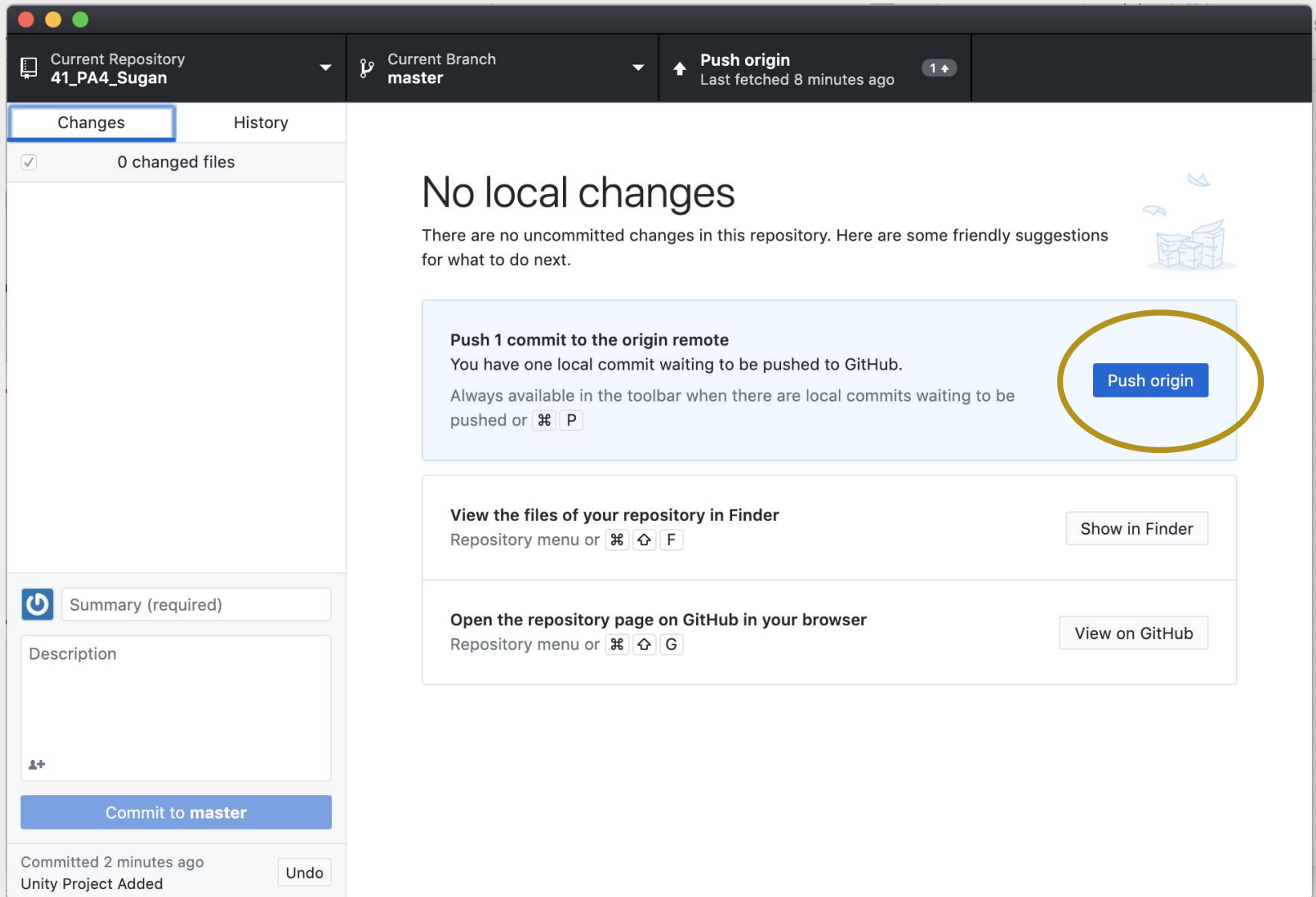
Reference image from history:



Step 6 : Push the unity project



Step 6 : Push the unity project



Step 7 : Verify GitHub Cloud

- Verify in GitHub cloud for the unity project created
- Merge the contents from Version 1 branch to Master

The screenshot shows a GitHub repository page for 'Sugan-hub / 41_PA4_Sugan'. The repository has 1 star and 0 forks. The main tab is 'Code', with other tabs for Issues (0), Pull requests (0), Actions, Projects (0), Wiki, Security (0), Insights, and Settings. The repository name is 'Practical Assignment 4'. Below the name, there are statistics: 5 commits, 2 branches, 0 packages, 0 releases, and 1 contributor. There are buttons for 'Branch: master', 'New pull request', 'Create new file', 'Upload files', 'Find file', and 'Clone or download'. A commit history table is shown with the following data:

Commit	Message	Time
41_Sugan_PA4	Unity Project Added	5 minutes ago
.gitattributes	Initial commit	6 days ago
README.md	Initial commit	6 days ago
Type_of_Testing.rtf	Updated Type of Testing.	6 days ago

Below the commit history, there is a section for the 'README.md' file, which contains the text '41_PA4_Sugan' and 'Practical Assignment 4'.

Step 8 : Inform your collaborator about the changes (Student A/Student B)

- Collaborator access the repository
 - *Student A access Student B's repository*
 - *Student B access Student A's repository*
- Use the client and pull the updates (fetch Origin) from the repository
- If the above step doesn't work, please clone the repository
- Verify the Unity project

Step 9 : Access to unity project

- Open the Unity project
- Create a new scene > MainScene
- Add in Label Text for a suitable label
- Add in a simple script, to load the CubeScene after 1 second
- Test the unity project
- Save and close the unity project



Step 10 : Save the project and push in the updates

- Create a new branch > Version 2
- Commit the changes to Version 2 branch
- Push the contents to GitHub cloud

Step 10 : Save the project and push in the updates

- Create a new branch > Version 2
- Commit the changes to Version 2 branch. Add in a commit message.
- Push the contents to GitHub cloud
- Verify the contents in GitHub Cloud
- Inform your team member

Step 11 : Team member access the repository

- Access to your owner repository
- Access to the contents of version 2 branch
- Verify the unity project for the updates
- Merge the master branch and version 2 branch
- Test the unity project

Step 12 :Copy and paste URL (your repository) for submission

- https://github.com/Rxyyxxn/18_Rayyan_PA04

Step 13 :Submission

- Save this PPT slides as pdf
- Save it as IndexNumber_YourName_PA4_practice
- Upload to Myconnexion