CIS 269 – Software Maintenance – Spring 2025

W5 Lab: Advanced Version Control

(For the due date, please refer to this lab’s posting on Canvas)

Purpose/knowledge/skills: There’s more to do with version control than we finished last week. We’ll also learn to integrate with GitHub, a popular on-line project management and source control site.

Version control in Visual Studio is under constant revision. Your workflow may be slightly different depending on the version of Visual Studio and underlying support pieces. Once accounts are all configured and connected, things work in a more predictable way. Keep in touch as we work through this!

Task 1: Register with GitHub (5 points)

1. Register with GitHub at <https://github.com/join>.

The work you do this week will be publicly visible to anyone on the internet, connected to the email address you use to register.

I recommending using a GitHub account with your @spscc.edu address for lab work. Later, on your own time, create a personal account for professional portfolios and project participation outside of CIS 269.

1. Take note of your GitHub account log-in and password.
2. Send a Canvas message to the professor with your GitHub account name.   
   Please don’t send your password.
3. The professor will add your name to the collaborator list for the repository for the   
   “CLONE A REPOSITORY” section of the lab.
4. When you’re registered and accept the invitation to the repository, take a screenshot/snip showing GitHub with your account logged in.

Rubric:

* GitHub registration: 5 points

Please paste your screenshot here:

|  |
| --- |
|  |

Task 2: Clone a github repository (5 points)

Let's clone a GitHub repository (<https://github.com/ashraf-spscc/CIS269SP25>) into Visual Studio.

You can use these steps to clone any public-access GitHub project!

Azure repos can be cloned the same way, although it may be more efficient to click the “clone” URL in the Azure Projects web interface

1. Open Visual Studio.
2. Click “Continue without code” if needed.
3. In the menu bar, click Git -> Clone Repository

|  |
| --- |
|  |

1. Paste the URL of the repository (<https://github.com/ashraf-spscc/CIS269SP25>) into the repository location.
2. Click Clone. Visual studio will work for a minute getting the files.
3. Depending on previous step
   1. If the solution explorer opens, skip to step 7.
   2. If needed, right-click the local git repository for CIS269SP25 and click “open”.

|  |
| --- |
|  |

* 1. If you are prompted to install the git command-line tools, do so by clicking “Install” and following the prompts. Accept all defaults on the git installation.

|  |
| --- |
|  |

* 1. If Visual Studio switches to the “Team Explorer” view, click “Show Folder View” to see the files.

|  |
| --- |
|  |

1. Visual Studio may switch to the solution view. It should like this, although there may be additional files.

|  |
| --- |
|  |

1. Click back to the Team Explorer Pane.
2. Right-click the repository and select "Open in Folder File Explorer".

|  |
| --- |
|  |

1. The contents of the folder should look like the screenshots that follow. There may be extra files based on the work other students have already done.

|  |
| --- |
|  |

Or

|  |
| --- |
|  |

1. Back in Visual Studio, click the Solution Explorer.
2. Paste a screenshot of the solution explorer showing the cloned project below the rubric.

Rubric:

* Screenshot showing cloned files: 5 points

Please paste your screenshot here:

|  |
| --- |
|  |

Task 3: Add and change files in the local repository (5 points)

Let's add some files to this Visual Studio project.

1. Save a G-rated image of an apple, pear, or banana to the folder you opened in the previous problem.
2. Please add your initials to the file name: for example, AppleJSM.jpg .

|  |
| --- |
|  |

1. The file should show up automatically in the Solution Explorer. The small “+” sign means the file is new since the last commit/push.

|  |
| --- |
|  |

1. Add your initials to the text in Overview.txt. If other students have their initials there, add your initials below them. The Solution Explorer shows a red check mark to indicate the file has changed.

|  |
| --- |
|  |
|  |

1. Let's add the files to the local repository. Either:

* In the menu bar, click Git -> Commit or Stash . . .
* In the team explorer pane, click “Git Changes”

Your change list will look different than mine!

|  |
| --- |
|  |

1. Type a commit message in the box, something like "JSM added an apple."
2. Click "Commit All".
3. Do not sync with the server yet, but please leave the window open.
4. Paste a screenshot/snip showing a successful local commit below the rubric.

Rubric:

* Screenshot of local commit: 5 points

Please paste your screenshot here:

|  |
| --- |
|  |

Task 4: Sync with GitHub (10 points)

1. If needed, save all files.
2. Try to sync your local repository with the remote one. In the menu bar, click Git -> Push.

(If you get an error here, that’s fine. See step 4.4.)

|  |
| --- |
|  |

1. If a GitHub login box appears, log in using your GitHub account created at the start of the lab.
2. If other students happen to work at the same time as you, GitHub may give you an error because the remote files have changed since you cloned the repository. That is fine. We talk more about merging and branching later. For now:
   1. Click “Git -> Pull” to get the latest files from the server.
   2. Click “Git -> Push”.
   3. It may take a few repetitions to do this at a time when other students aren’t actively changing the repository.
3. Open your local copy of the repository in the file manager. Confirm that your file has been added. Files added by other students may appear also.

|  |
| --- |
|  |

1. In the future, you can click "Git -> Sync" to do an automatic Pull-then-Push in a single step.

|  |
| --- |
|  |

1. Take a screen shot of your local repository in the Windows file manager. The repository should show at least the text file, your image file, and possibly image files from other students. Paste your screen shot below the rubric.

Rubric:

* Screenshot showing local repo with your image file and possibly others: 10 points

Please paste your screenshot here:

|  |
| --- |
|  |

Task 5: Explore branch and merge for version control

1. Create a new branch in Visual Studio by following these steps.
2. In the menu bar, click “Git -> Manage Branches” to see the current branches.

|  |
| --- |
|  |

1. Right-click the Main branch, select “New Local Branch From . . .”

|  |
| --- |
|  |

1. Enter a branch name with your initials. For the instructor, this is JSMBranch. Your will have different initials.

|  |
| --- |
|  |

1. Click “Create”. A new branch is created.

|  |
| --- |
|  |

1. Go back to the solution explorer.
2. Add another image file (apple, pear, or banana) to the project.

|  |
| --- |
|  |

1. Commit the branch change locally.

You may need to go to Git -> Commit or Team Explorer -> Git Changes to do this.

1. Merge the new branch with “Main” to make the changes permanent.

Git -> Manage Branches.

1. Double-click main.
2. Right-click your branch and select “merge [your branch” into Main”

|  |
| --- |
|  |

1. Merge from the branch you created into the current branch main.
2. Click “Merge”.
3. View the solution explorer. You should see the additional file added by merging your branch with the main branch.

|  |
| --- |
|  |

1. Please paste a screenshot below the rubric, showing main with your second image file added.
2. Delete your branch by right-clicking the branch in the team explorer and choosing “Delete”.
3. You do not need to sync back with GitHub at this time.

Rubric:

* Screenshot showing second image in the repo: 10 points

Please paste your screenshot here:

|  |
| --- |
|  |

Task 6: Work with blame in version control (5 points)

1. In the solution explorer, right-click “Overview.txt” and select “Git -> View History.”
2. Page a screenshot of the history pane here.

|  |
| --- |
|  |

1. In the solution explorer, right-click “Overview.txt” and select “Git -> Blame (Annotate)”.
2. Page a screenshot of the blame/annotations pane here.

|  |
| --- |
|  |

Rubric:

* Two screenshots: 5 points total, 2.5 points each

Task 7: Answer questions using history/blame in version control (5 points)

1. Who was the last user to edit line 1? The GitHub user name reported in the Annotations pane is all you need here.

|  |
| --- |
|  |

1. Who was the last user to edit the last line of the file?

(Hint: is this in Annotations or History?)

|  |
| --- |
|  |

1. Who was the last user to edit the file? (Hint: is this in Annotations or History?)

|  |
| --- |
|  |

1. Who has edited the file the most times? (Hint: is this in Annotations or History?)

|  |
| --- |
|  |

1. Who has edited most of the lines in Overview.txt? (Hint: is this in Annotations or History?)

|  |
| --- |
|  |

Rubric:

* Questions: 5 points total, 1 point each

Task 8: Perform program maintenance (10 points)

Space War (Space Demo) is a JavaScript game you can play by opening "Index.html" in a web browser. Press the left and right arrow keys to move, and space to shoot.

You’ll fix some bugs in this program.

**Use the local Commit feature of source control to make backup copies.**

Please fix the following bugs:

1. The player ship moves faster in one direction than the other.

Fix it so it moves the same speed both ways.

1. Choose one of the following.
   1. Adjust the update() function so the player can wrap around the screen instead of stopping at the left/right edge

OR

* 1. Add an explosion graphic to the enemies

1. Zip the project and this word document together, and submit the zip file through Canvas.

Rubric:

* Ship speed fix: 5 points
* Update() function or Explosion graphic: 5 points

Task 9: Document source control from program maintenance (5 points)

For the Space War project in the previous task . . .

1. Paste a screenshot showing your commit history for the project under Git -> View Branch History. The screenshot must show multiple commits to earn credit.
2. Go to the Solution Explorer.
3. Select a file you have changed several times.
4. Right-click on the file and select “Git -> View History.”
5. Right-click on a commit in the history and click “Compare with previous”.
6. Paste a screenshot showing diff or merge views of two versions of the file.

Example submission

|  |
| --- |
|  |
|  |

Rubric:

* Two screenshots: 5 points total, 2.5 points each

Screenshot #1:

|  |
| --- |
|  |

Screenshot #2:

|  |
| --- |
|  |