

SWE20001 – Development Project 1: Tools and Practices

Pass Task <2.1P>

Duy Phuong Nguyen (101204984)

Lab: Friday 2:30pm

Tutor: Huai Liu

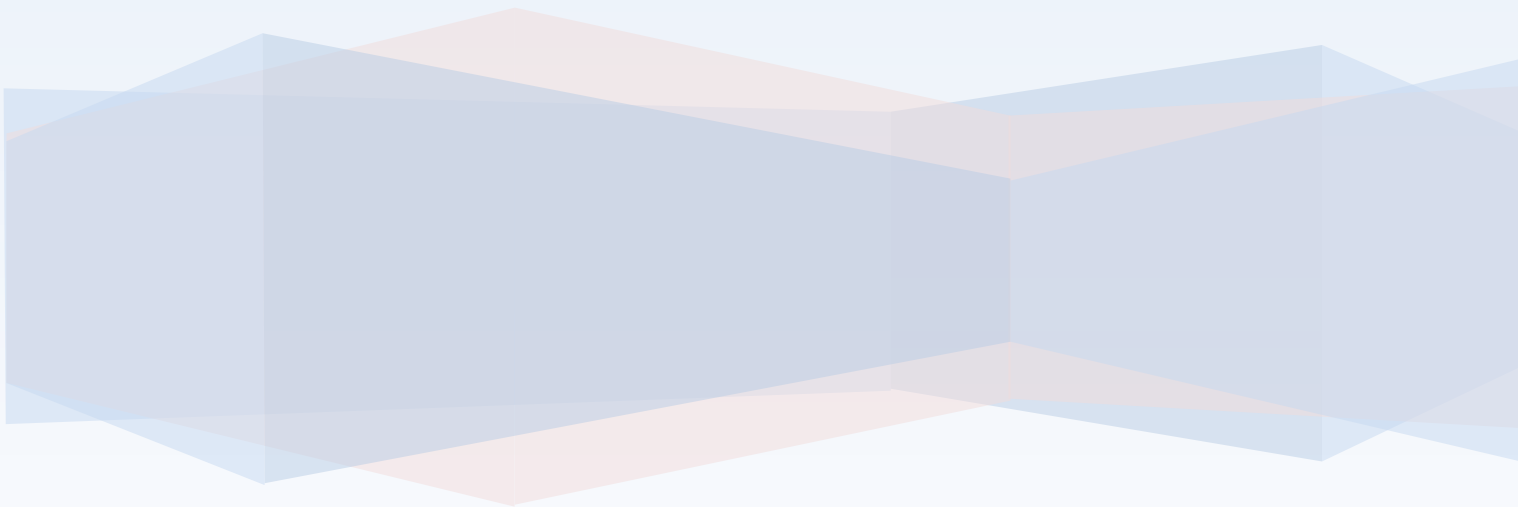
Team number: 3

Henry Tran 102075482

Edward Knight 102700043

Duy Phuong Nguyen 101204984

Christian Ilott 102169762



This report summarises my contributions to the teamwork in Week <2> as well as the tasks I have performed during this week:

Activity 2.1. Team formation and To Do process:

During this team activity, I have done the following to contribute to the To Do process our team carried out:

- First, sign up and install Git to my computer
- Then setup the git by checking version, input username and email address

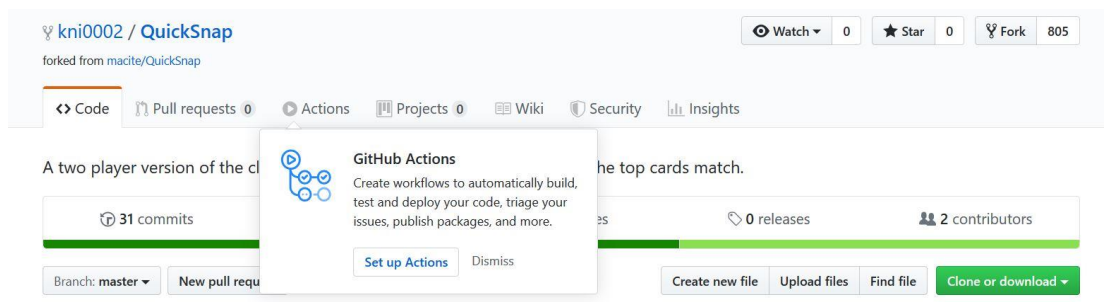
```
mngdu@DESKTOP-M9QOI14 MINGW64 ~
$ git version
git version 2.25.1.windows.1

mngdu@DESKTOP-M9QOI14 MINGW64 ~
$ git config --global user.name "mngduyphuong"

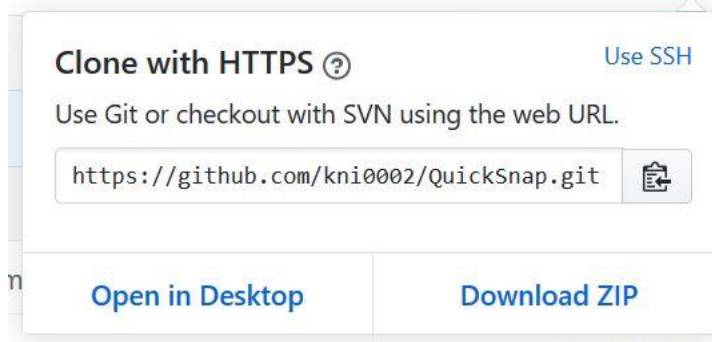
mngdu@DESKTOP-M9QOI14 MINGW64 ~
$ git config --global user.email "mngduyphuong@gmail.com"

mngdu@DESKTOP-M9QOI14 MINGW64 ~
$
```

- Our team had a little discussion and assigned roles for everyone:
 - The team's leader is Edward Knight
 - UI Coder is Henry Tran
 - Game Developer 1 is Duy Phuong Nguyen
 - Game Developer 2 is Christian Ilot
- The team's leader added me to the project so I can see the QuickSnap repo



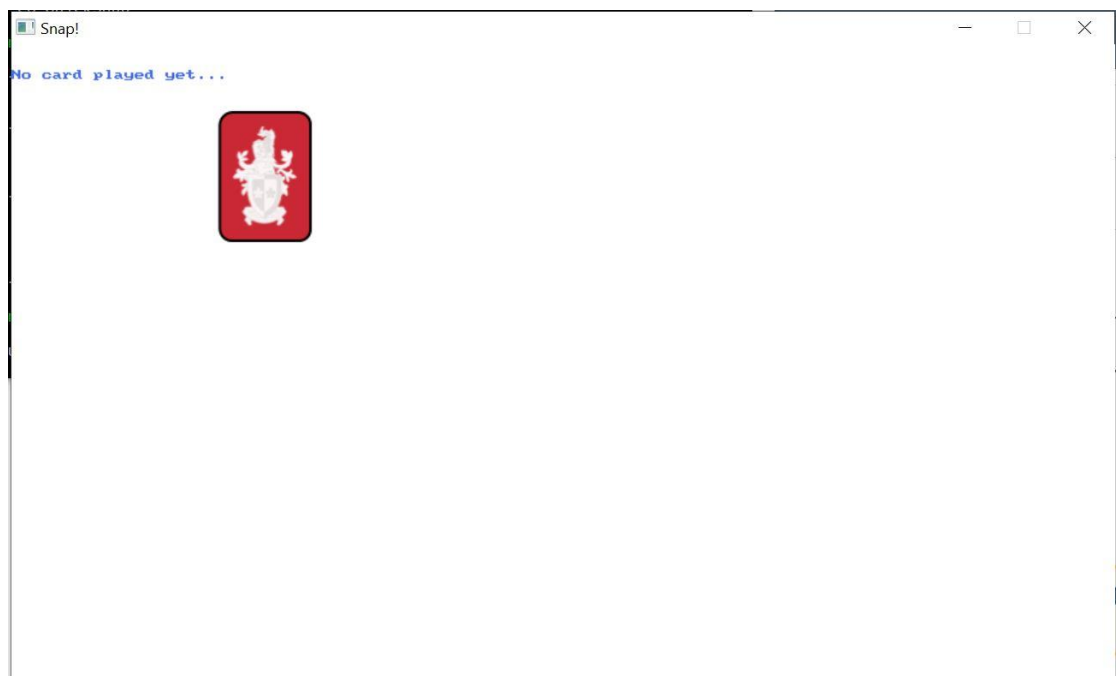
- Added the HTTPS clone url from GitHub to my terminal



```
mngdu@DESKTOP-M9QOI14 MINGW64 ~  
$ cd /c/users/mngdu  
  
mngdu@DESKTOP-M9QOI14 MINGW64 /c/users/mngdu  
$ https://github.com/kni0002/QuickSnap.git  
bash: https://github.com/kni0002/QuickSnap.git: No such file or directory  
  
mngdu@DESKTOP-M9QOI14 MINGW64 /c/users/mngdu  
$ git clone https://github.com/kni0002/QuickSnap.git  
Cloning into 'QuickSnap'...  
remote: Enumerating objects: 1, done.  
remote: Counting objects: 100% (1/1), done.  
remote: Total 221 (delta 0), reused 0 (delta 0), pack-reused 220  
Receiving objects: 100% (221/221), 7.74 MiB | 2.75 MiB/s, done.  
Resolving deltas: 100% (65/65), done.
```

- Testing the project and trying to run the game

```
mngdu@DESKTOP-M9QOI14 MINGW64 /c/users/mngdu  
$ cd QuickSnap  
  
mngdu@DESKTOP-M9QOI14 MINGW64 /c/users/mngdu/QuickSnap (master)  
$ ./build.sh  
-----  
                Creating QuickSnap  
                for windows  
-----  
Running script from /c/users/mngdu/QuickSnap  
Saving output to ./bin/Debug  
Compiler flags -debug -define:DEBUG  
-----  
... Creating QuickSnap  
... Copying libraries  
... Copying Resources into QuickSnap  
Finished  
-----
```



- I started to work on my own part: Improving the game with auto-flipping cards. The first step is creating a new branch and name it add-automatic-card-flip

```
mngdu@DESKTOP-M9QOI14 MINGW64 /c/users/mngdu/QuickSnap (master)
$ git checkout -b add-automatic-card-flip
Switched to a new branch 'add-automatic-card-flip'
```

- Create Game Timer in Snap.cs file

```
public Snap ()
{
    _deck = new Deck ();
    _gameTimer = SwinGame.CreateTime();
}
```

- Update Start method

```
public void Start()
{
    if ( ! IsStarted )           // only start if not already started!
    {
        _started = true;
        _deck.Shuffle ();        // Return the cards and shuffle

        FlipNextCard ();        // Flip the first card...
        _gameTimer.Start();
    }
}
```

- Set up the Update method to check the time

```
public void Update()
{
    if(_gameTimer.Ticks>_flipTime)
    {
        _gameTimer.Reset();
        FlipNextCard();
    }
}
```

- Update HandleUserInput method in SnapGame.cs so user can press spacebar to start the game

```
private static void HandleUserInput(Snap myGame)
{
    //Fetch the next batch of UI interaction
    SwinGame.ProcessEvents();

    if (SwinGame.KeyTyped (KeyCode.vk_SPACE))
    {
        myGame.Start();
    }
}
```

- Update PlayerHit method

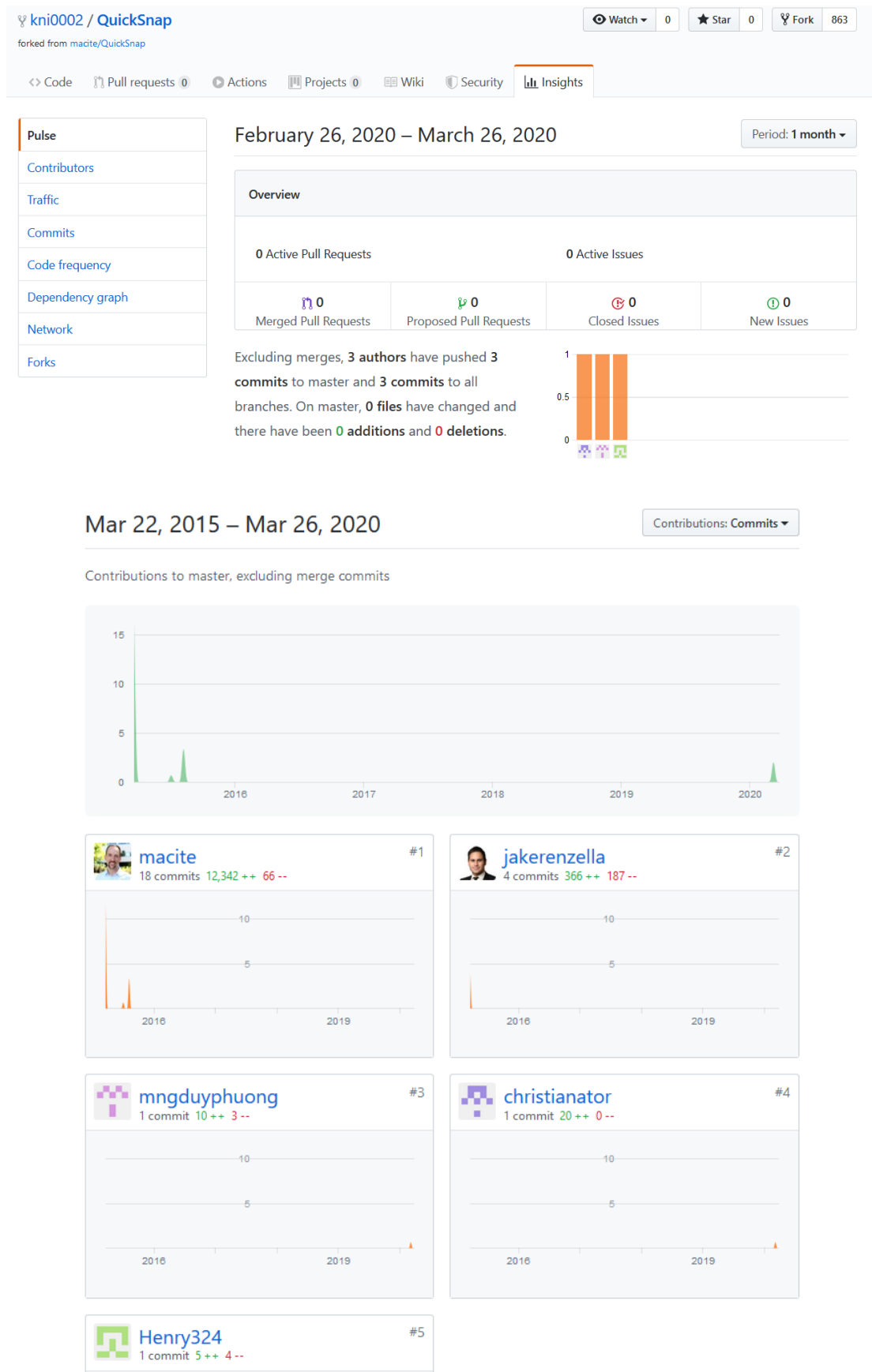
```
public void PlayerHit (int player)
{
    //TODO: consider deducting score for miss hits???
    if ( player >= 0 && player < _score.Length && // its a valid player
        IsStarted && // and the game is started
        _topCards [0] != null && _topCards [0].Rank == _topCards [1].Rank) // and its a match
    {
        _score[player]++;
        //TODO: consider playing a sound here...
    }

    // stop the game...
    _started = false;
    _gameTimer.Stop();
}
```

- All of my works then push to the sever by using the command in Git
git push -u origin add-automatic-card-flip

Below is the proof of my contributions and Network demonstration

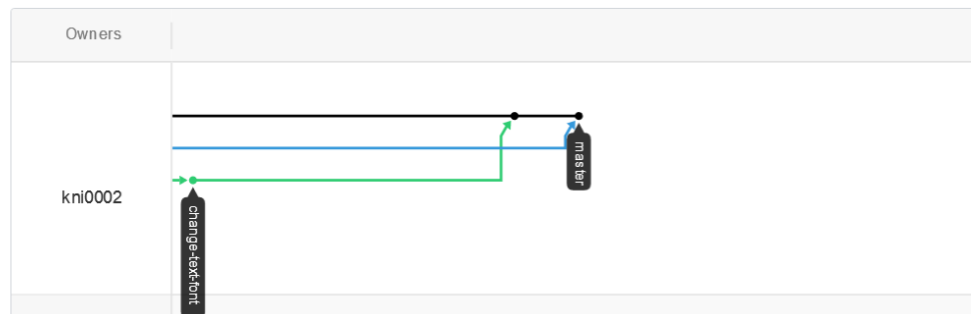
- The middle user is me with a purple profile picture



Network graph

Timeline of the most recent commits to this repository and its network ordered by most recently pushed to.

The repository network shows the 100 most recently pushed forks. Do you need to see more forks? Please [give us feedback](#) on your usage of this feature.



Command	Description	Screenshot
<code>git --version</code>	Check the current version of gitbash	<pre>mngdu@DESKTOP-M9QOI14 MINGW64 ~ \$ git version git version 2.25.1.windows.1 mngdu@DESKTOP-M9QOI14 MINGW64 ~ \$ git config --global user.name "mngduyphuong" mngdu@DESKTOP-M9QOI14 MINGW64 ~ \$ git config --global user.email "mngduyphuong@gmail.com" mngdu@DESKTOP-M9QOI14 MINGW64 ~ \$ </pre>
<code>git config -global user.name</code> <code>git config --global user.email</code>	Setup Git username and email	<pre>mngdu@DESKTOP-M9QOI14 MINGW64 ~ \$ git version git version 2.25.1.windows.1 mngdu@DESKTOP-M9QOI14 MINGW64 ~ \$ git config --global user.name "mngduyphuong" mngdu@DESKTOP-M9QOI14 MINGW64 ~ \$ git config --global user.email "mngduyphuong@gmail.com" mngdu@DESKTOP-M9QOI14 MINGW64 ~ \$ </pre>
<code>git clone /path/.../...</code>	Clone the file from GitHub online sever to local computer	<pre>mngdu@DESKTOP-M9QOI14 MINGW64 ~ \$ cd /c/users/mngdu mngdu@DESKTOP-M9QOI14 MINGW64 /c/users/mngdu \$ https://github.com/kni0002/QuickSnap.git bash: https://github.com/kni0002/QuickSnap.git: No such file or directory mngdu@DESKTOP-M9QOI14 MINGW64 /c/users/mngdu \$ git clone https://github.com/kni0002/QuickSnap.git Cloning into 'QuickSnap'... remote: Enumerating objects: 1, done. remote: Counting objects: 100% (1/1), done. remote: Total 221 (delta 0), reused 0 (delta 0), pack-reused 220 Receiving objects: 100% (221/221), 7.74 MiB 2.75 MiB/s, done. Resolving deltas: 100% (65/65), done.</pre>
<code>git checkout -b <name></code>	Create a new branch	<pre>\$ git checkout -b testing Switched to a new branch 'testing'</pre>

./build.sh ./run.sh	Build and run the project	<pre> mngdu@DESKTOP-M9QOI14 MINGW64 /c/users/mngdu \$ cd QuickSnap mngdu@DESKTOP-M9QOI14 MINGW64 /c/users/mngdu/QuickSnap (master) \$./build.sh ----- Creating QuickSnap for Windows ----- Running script from /c/users/mngdu/QuickSnap Saving output to ./bin/Debug Compiler flags -debug -define:DEBUG ----- ... Creating QuickSnap ... Copying libraries ... Copying Resources into QuickSnap Finished ----- </pre>
git status git diff	Check the status and differences	<pre> mngdu@DESKTOP-M9QOI14 MINGW64 /c/users/mngdu/QuickSnap (add-automatic-card-flip) \$ git status On branch add-automatic-card-flip Your branch is up to date with 'origin/add-automatic-card-flip'. </pre>
Git push -u origin <name>	Push your changes to the sever	<pre> mngdu@DESKTOP-M9QOI14 MINGW64 /c/users/mngdu/QuickSnap (add-automatic-card-flip) \$ git push -u origin Enumerating objects: 15, done. Counting objects: 100% (15/15), done. Delta compression using up to 4 threads Compressing objects: 100% (8/8), done. Writing objects: 100% (14/14), 142.87 KiB 3.97 MiB/s, done. Total 14 (delta 1), reused 0 (delta 0) remote: Resolving deltas: 100% (1/1), completed with 1 local object. To https://github.com/kni0002/QuickSnap.git bb1a8dc,4e05dea add-automatic-card-flip -> add-automatic-card-flip Branch 'add-automatic-card-flip' set up to track remote branch 'add-automatic-card-flip' from 'origin'. </pre>

Diagram of the process

