

CS304

Software Engineering

TAN, Shin Hwei

陈馨慧

Instructor: 胡春风, 王大兴

Southern University of Science and Technology

Welcome to the first lab!

Objective today:

- Get to know your classmates for the class project
- Learn about roles in software development teams
- Play a game on Flame of Open Source (useful intro to the project)

Let's play a game

- Many roles exist in a software development team
- How to play?
 1. Learn about the responsibilities for each role
 2. Decide which role you want to play
 3. Answer interview question about each role
 1. **Do not search online for answer!**
 4. Decide which role you want to play based on your suitability

Which roles do you want to be?

- Software Engineer
- Software Engineer in Test
- Technical Project Manager

What do Software Engineer do?

- Responsibilities:
 - Designing and coding tools.
 - A team player who works well in a collaborative environment with peers in other development disciplines, Quality and Program Management.

What do Software Engineer in Test do?

- Responsibilities:
 - Build advanced automated test suites to exercise our applications.
 - Work with the development and test engineering teams to automate testing.
 - Conduct research on emerging technologies.
 - Analyze and decompose a complicated software system and design a strategy to test this system.

What do Technical Project Manager do?

- Responsibilities:
 - Lead project as technical architect in initiation phase to conclude right approach with several team
 - Work as project manager to make project through collaboration with development manager and product manager
 - Involve in multiple project executions simultaneously

Which role you want to be?

Discuss with your classmates sitting around you

- Why you want to be ...?
- Do your classmate think that you are suitable?

Discuss around ~5 minutes

Which role you want to be?

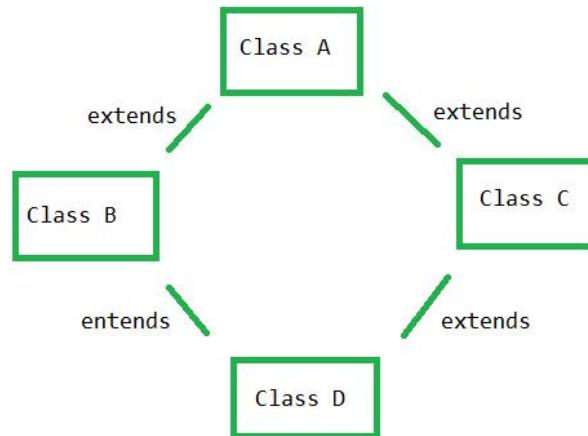
- How many of you want to be a Software Engineer?
- How many of you want to be a Software Engineer in Test?
- How many of you want to be a Technical Project Manager?

Now try answering interview questions for all roles..

Software Engineer

A real interview quick question I get when I was a student:

- Does Java support Multiple inheritance?



Discuss around
~2 minutes

Software Engineer In Test

Interview Question:

- Write test cases for adobe reader

Discuss around
~10 minutes

Technical Project Manager

Interview Question:

- Choose a Google/Microsoft/Tencent/Apple product and talk about it, what do you like about it, what would you improve.

Discuss around
~10 minutes

Which role you want to be now?

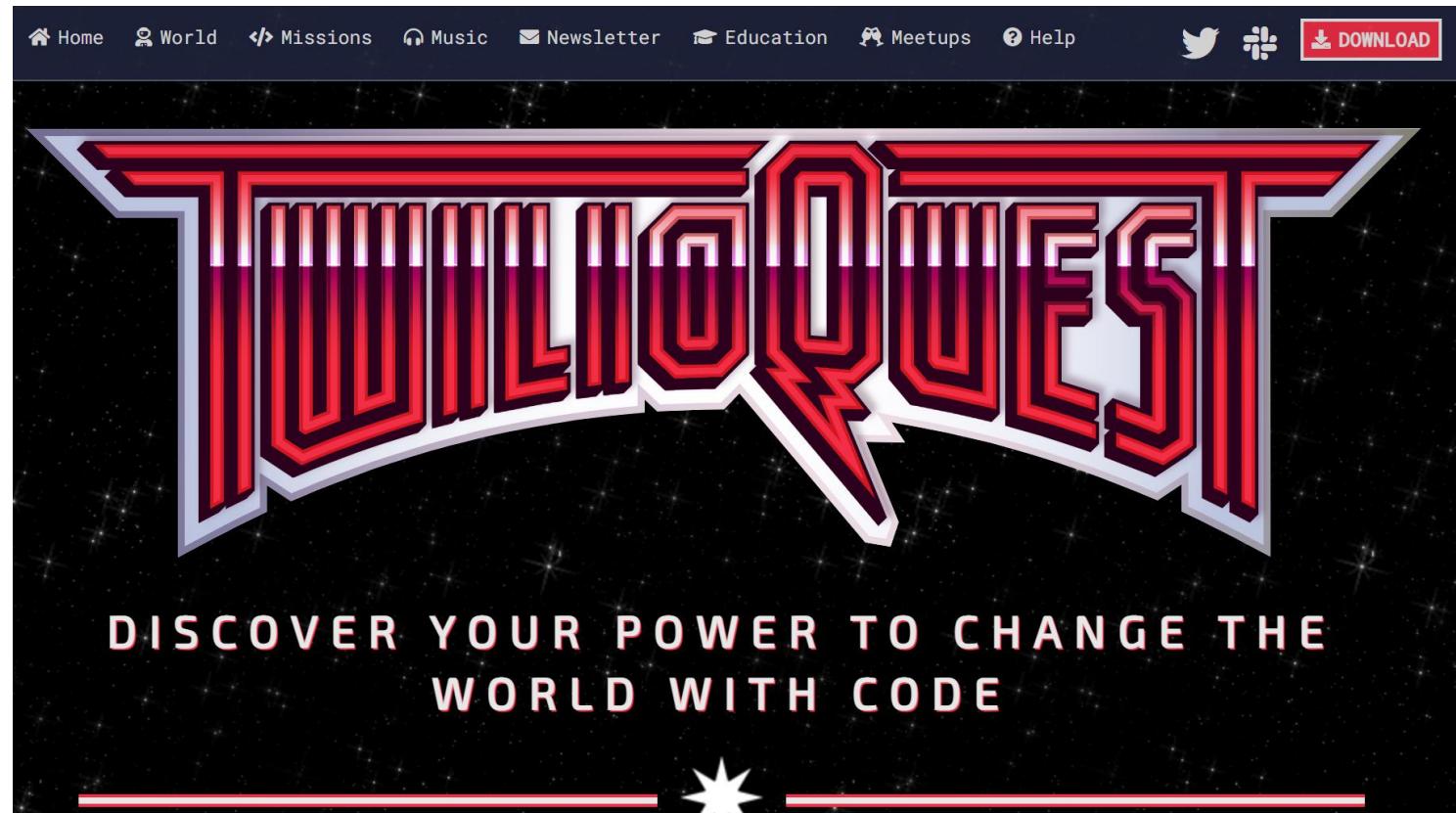
Discuss with your classmates sitting around you

- How many of you want to be a Software Engineer?
- How many of you want to be a Software Engineer in Test?
- How many of you want to be a Technical Project Manager?

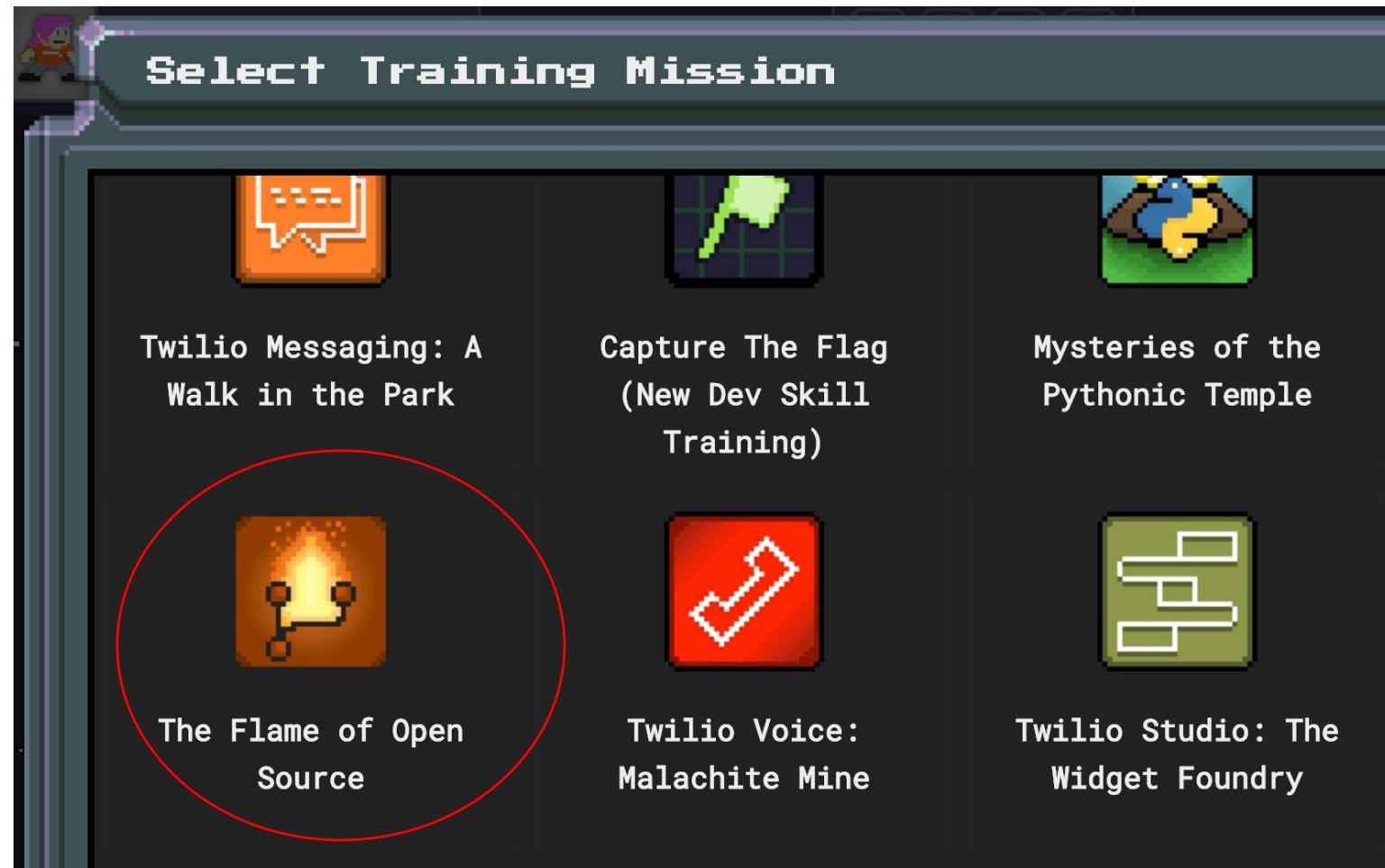
Part 2 – Flame of Open Source Game

TwilioQuest Game

- Download the twilioQuest game! (Need VPN)
- <https://www.twilio.com/quest>



Select The Flame of Open Source



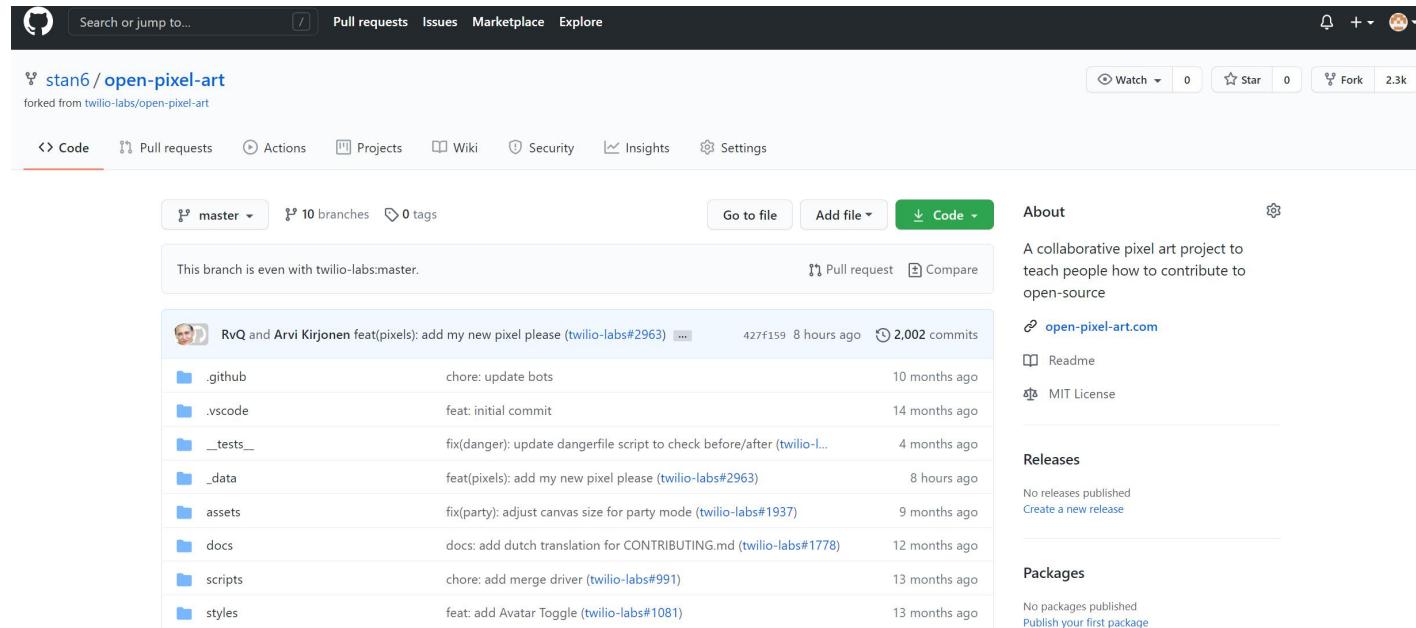
First Step

- Create a GitHub account



Second Step

- Create a fork

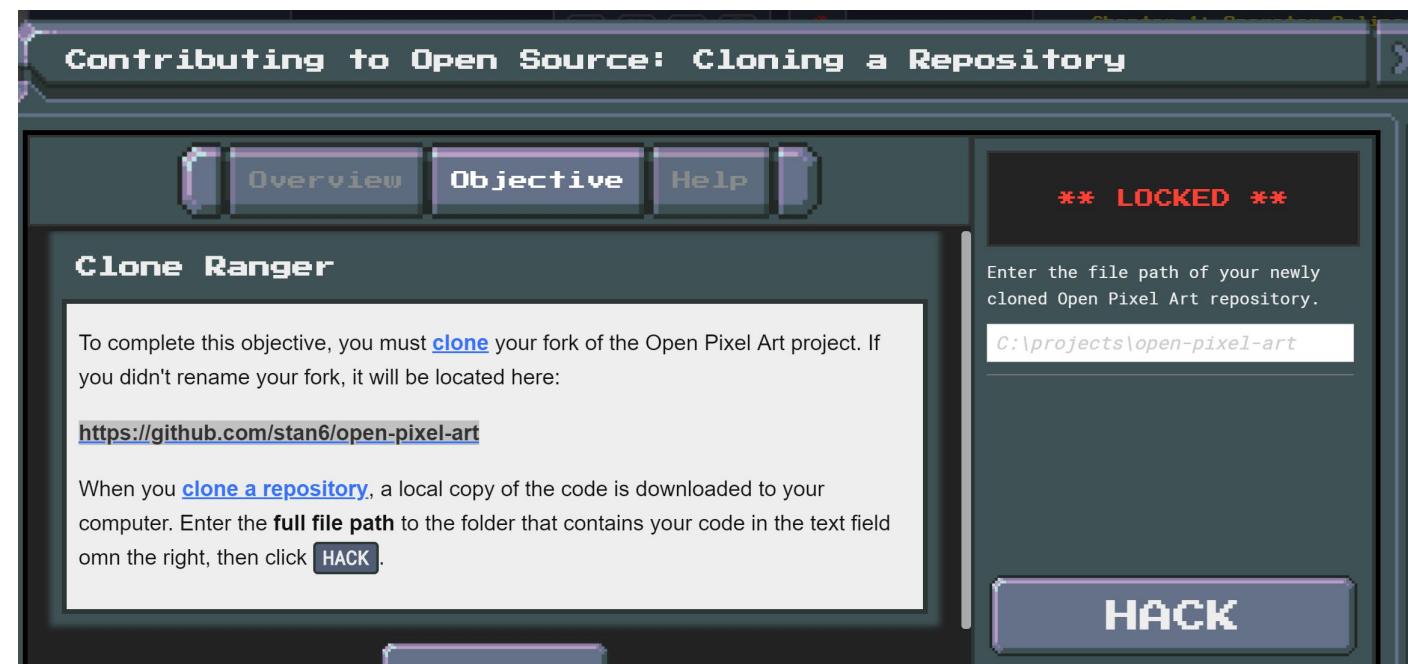


Third Step

- Install git
 - <https://git-scm.com/book/en/v2/Getting-Started-Installing-Git/>

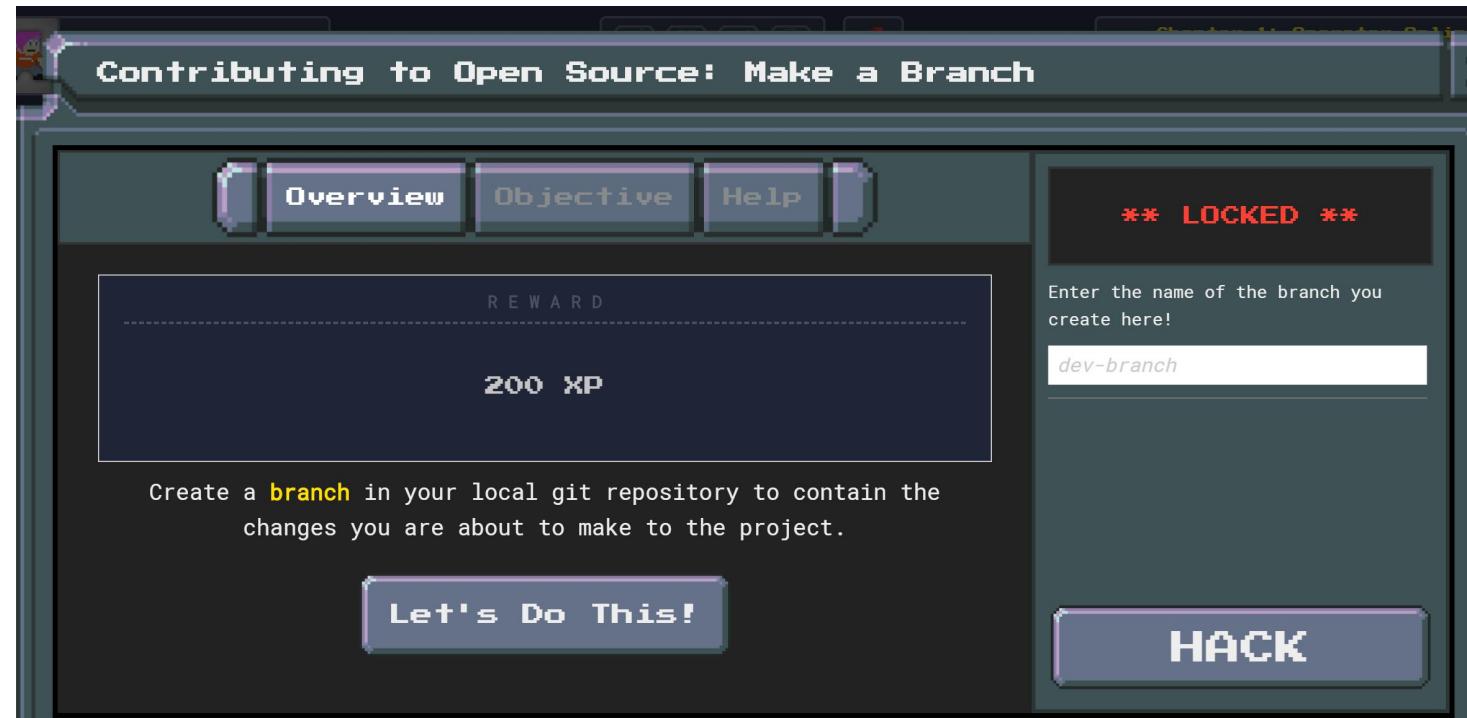
4th Step

- Clone the open-pixel-art repository according to the game



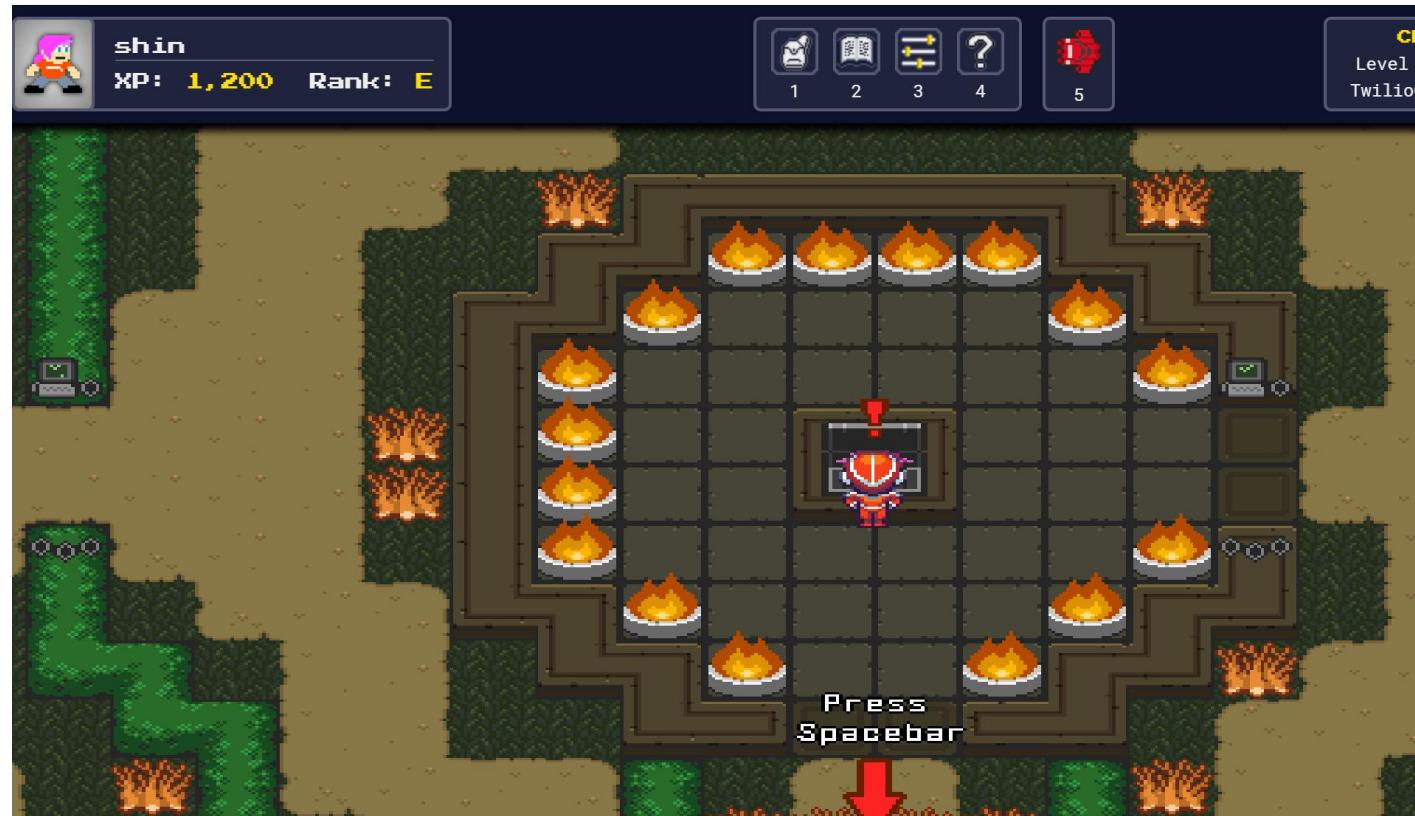
Fifth Step

- Make a branch according to the game



Sixth Step

- Make your first contribution!

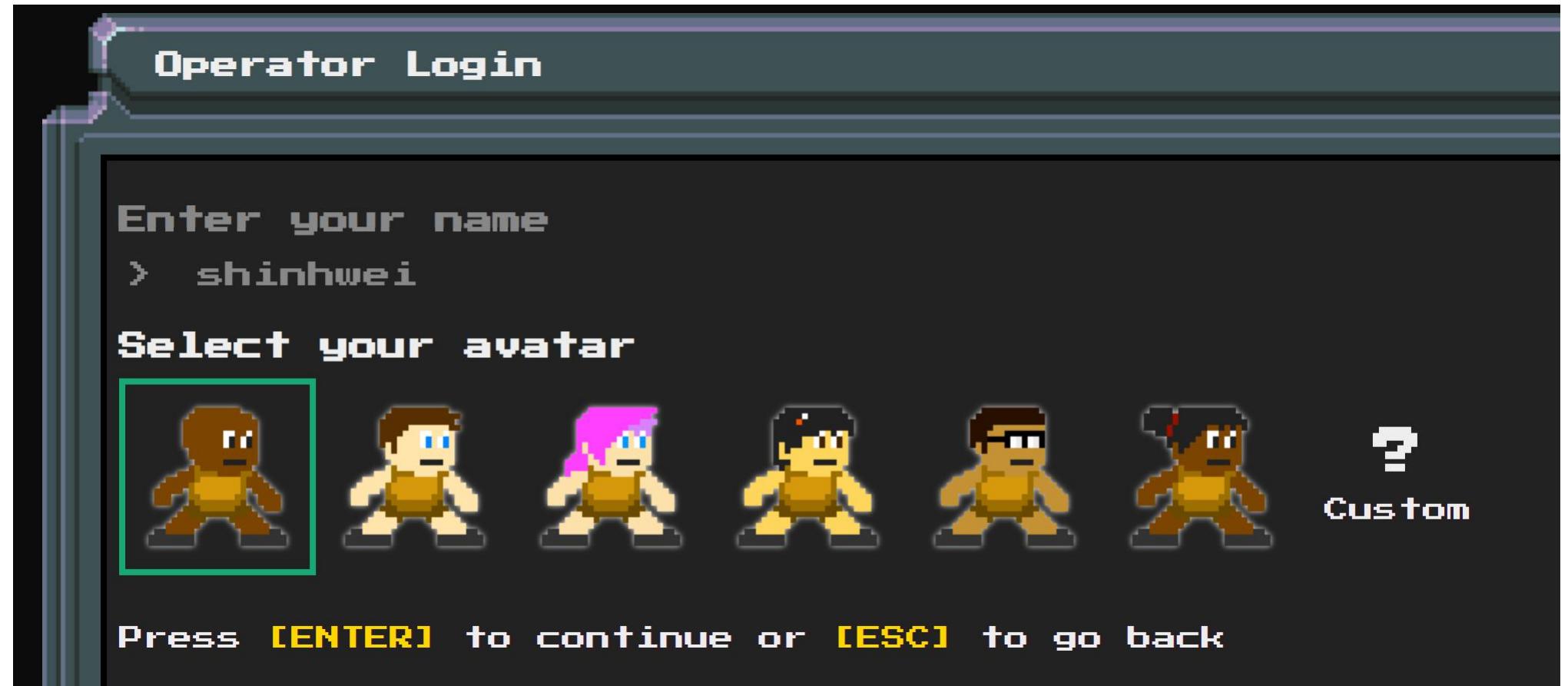


Without VPN

Two ways

1. Use this link: <http://10.20.71.79:10086/lab1/>
2. Or follow the offline instruction in the following slides

Screen 1



Screen 2



Screen 3

Select Training Mission

 Twilio Messaging: A Walk in the Park	 Capture The Flag (New Dev Skill Training)	 Mysteries of the Pythonic Temple	 The Flame of Open Source 0% Complete Rank: None
 The Flame of Open Source	 Twilio Voice: Malachite Mine	 Twilio Studio: The Widget Foundry	In a desperate attempt to discourage new developers from contributing to open source projects, the Legacy Systems have hidden the Flame of Open Source in the center of a forest maze. Learn how to use git and GitHub to contribute to an open source project, and use the Flame of Open Source to clear a path for others to do the same!

Screen 4

Select Training Mission

 Twilio Messaging: A Walk in the Park	 Capture The Flag (New Dev Skill Training)	 Mysteries of the Pythonic Temple	 The Flame of Open Source 0% Complete Rank: None
 The Flame of Open Source	 Twilio Voice: Malachite Mine	 Twilio Studio: The Widget Foundry	In a desperate attempt to discourage new developers from contributing to open source projects, the Legacy Systems have hidden the Flame of Open Source in the center of a forest maze. Learn how to use git and GitHub to contribute to an open source project, and use the Flame of Open Source to clear a path for others to do the same!

Screen 5



Screen 6

The screenshot shows a game interface with a dark, futuristic aesthetic. At the top, a banner reads "Contributing to Open Source: Join GitHub". Below it is a navigation bar with three tabs: "Overview" (selected), "Objective", and "Help".

The main content area is titled "Git Started" and contains the following text:

Many open source projects (and many software companies) use [git](#) for [version control](#) and collaboration. [GitHub](#) is a popular platform that many open source projects use to host their [git repositories](#).

The challenges to follow will take you through contributing to an open source project hosted on GitHub. The first step on that journey is [creating a free GitHub account](#).

Once you've made your free GitHub account, **enter the username you chose** into the field on the right and press **HACK**.

To the right of the main content is a "Hacking" interface with a title "** HACKING **". It features a text input field containing "stan6" and a large blue button labeled "HACK".

Screen 7

The screenshot shows a game interface with a dark blue header bar. The title "Contributing to Open Source: Join GitHub" is displayed in white text. Below the header is a navigation bar with three tabs: "Overview" (selected), "Objective", and "Help".

The main content area has a dark grey background and contains the following sections:

- Lets Make a GitHub Account**
- What is GitHub?**

GitHub is a website that hosts open source code on the internet so that developers around the world can collaborate on it remotely. It runs on a technology for code sharing called **git** that we will learn a little about in later objectives.
- GitHub Signup Process**

Follow the GitHub sign up process to [create a new user account](#). You will only need a free plan to complete this objective, and as long as you're working on open source software you'll only ever need the free plan!

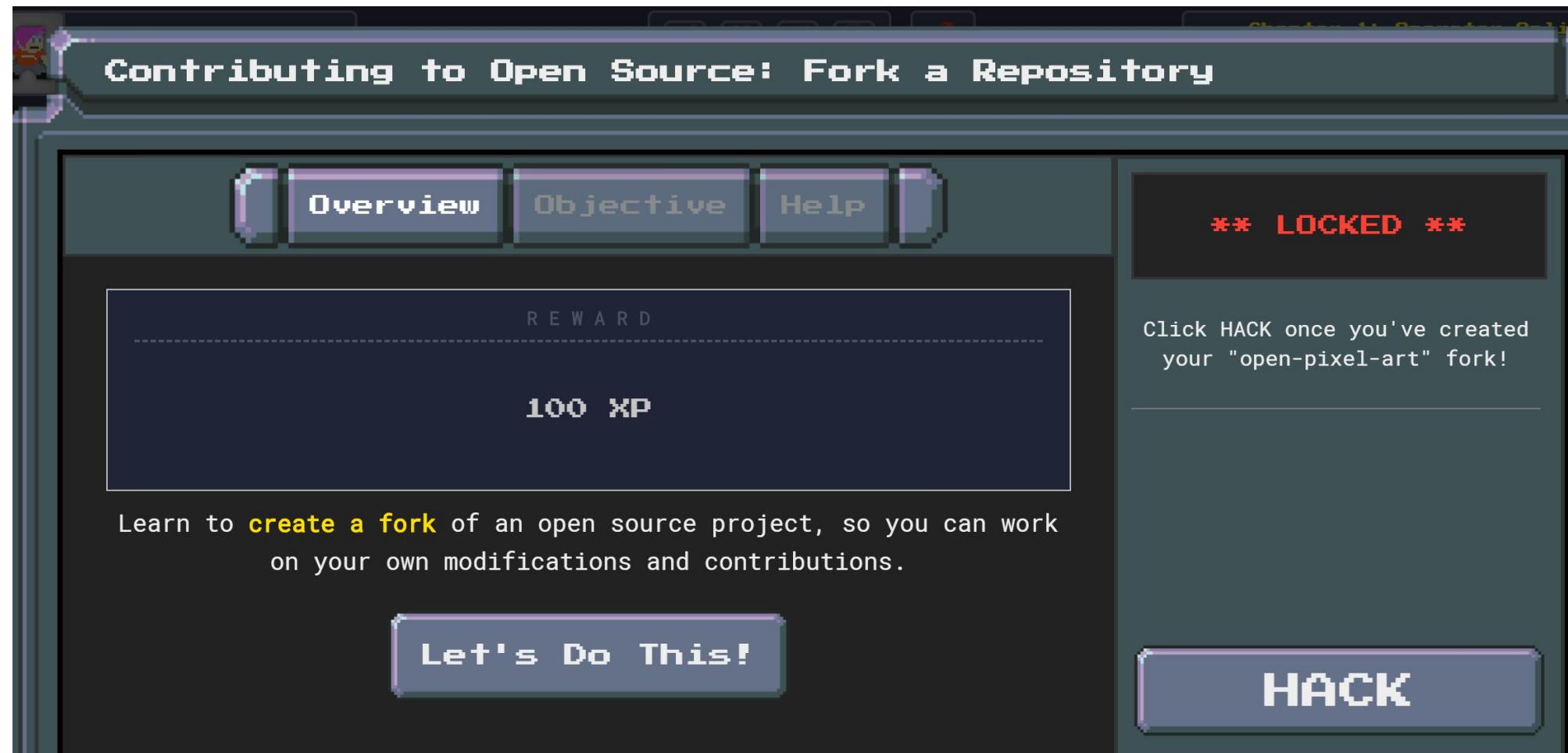
To the right of the main content is a sidebar with a dark grey background and a black header bar containing the text "**** HACKING ****". It includes a text input field labeled "Enter your GitHub username here." with the value "stan6" and a large blue button labeled "HACK".

<https://github.com/join>

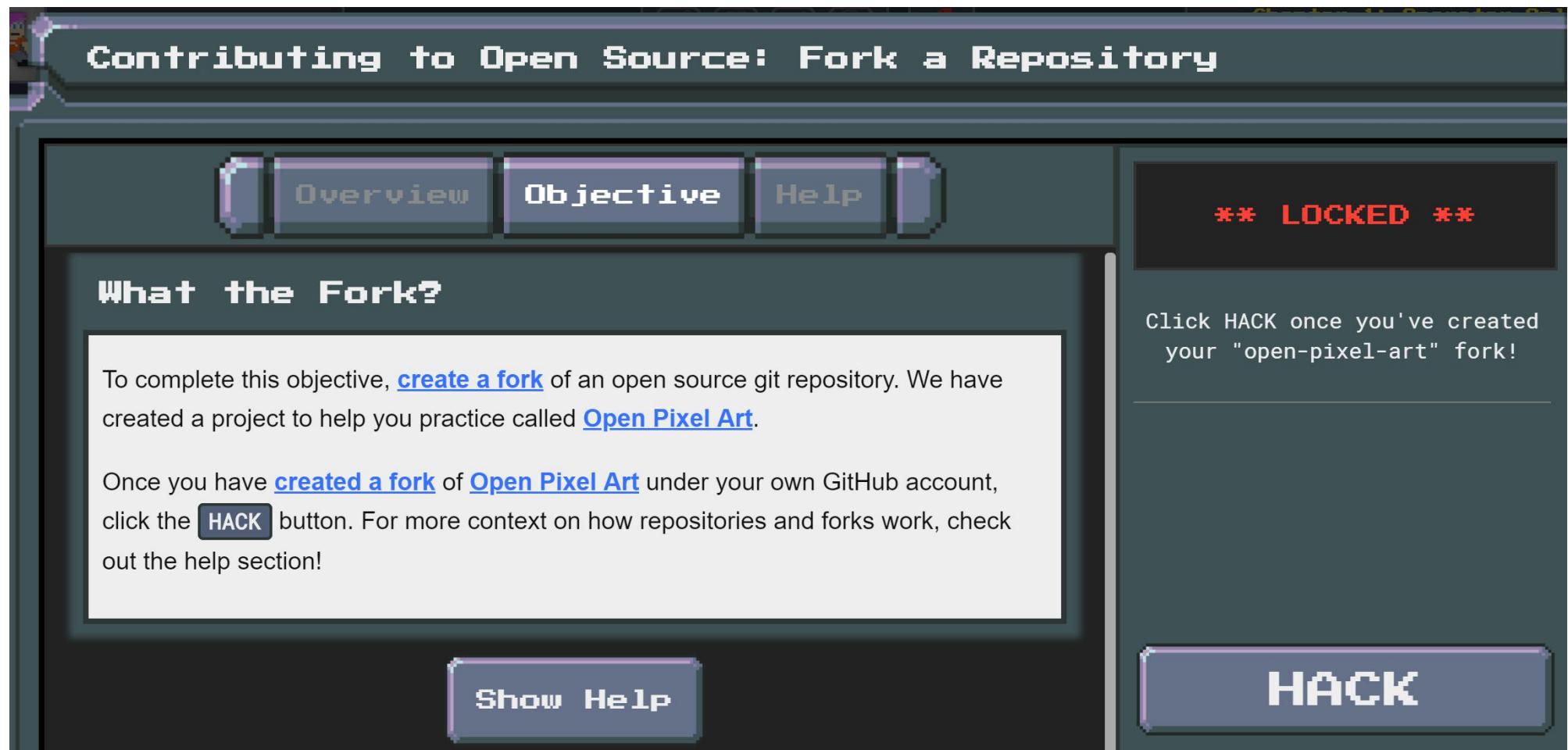
Screen 8



Screen 9



Screen 10



create a fork: <https://help.github.com/en/articles/fork-a-repo>
Open Pixel Art: <https://github.com/twilio-labs/open-pixel-art>

Screen 10

Contributing to Open Source: Fork a Repository



*** Objective Clear! ***

100 XP earned!

** SUCCESS **

We found your fork of the Open Pixel Art repository! Good job!

[< Back to tutorial](#)

DONE

Screen 11

Contributing to Open Source: Install Git

Overview Objective Help

** LOCKED **

R E W A R D

200 XP

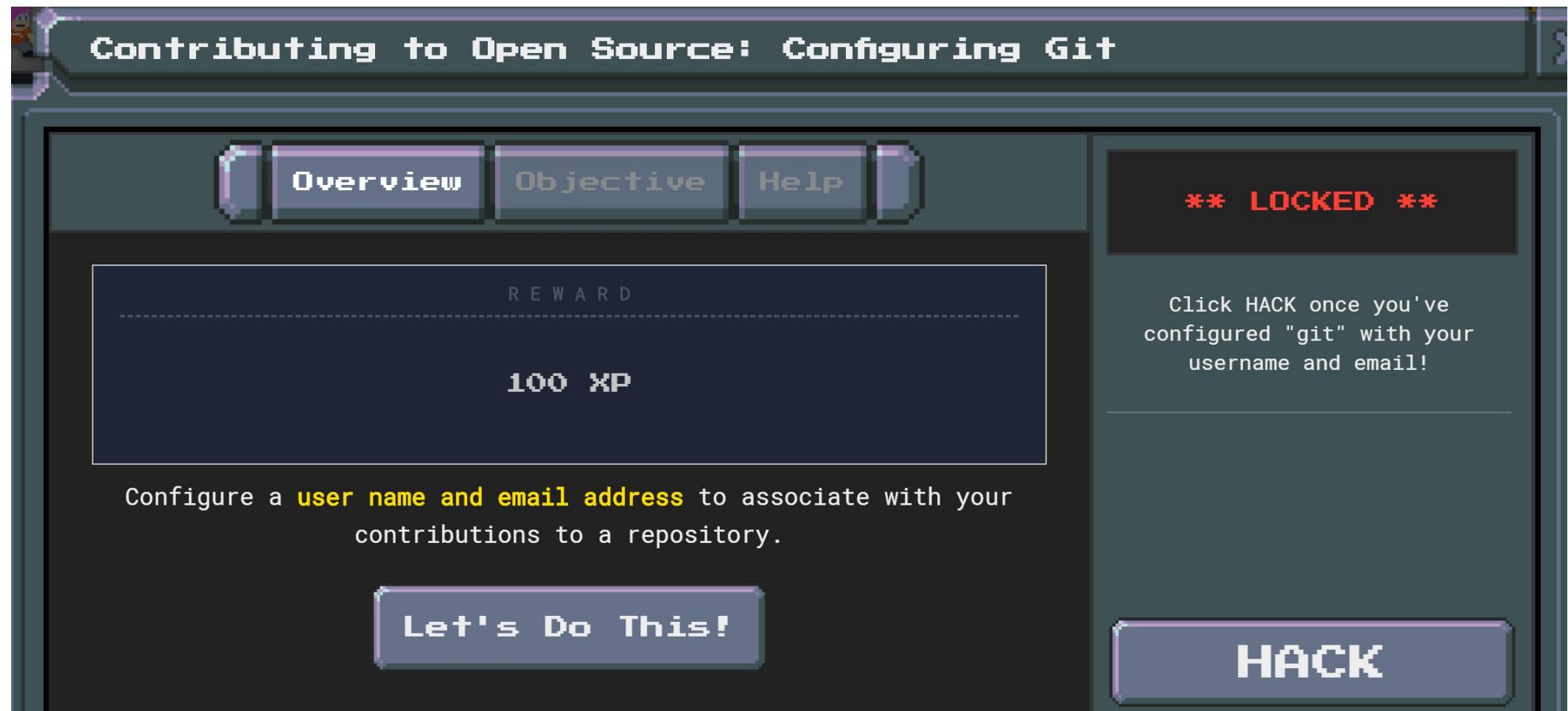
Install the **git command line tools** so you can work with git repositories on your computer.

Let's Do This!

HACK

The image shows a game-like interface for learning about contributing to open source. The title 'Contributing to Open Source: Install Git' is at the top. Below it is a navigation bar with three buttons: 'Overview', 'Objective', and 'Help'. To the right of the navigation bar is a dark box with the text '** LOCKED **' in red. In the center-left is a dark blue box labeled 'REWARD' with '200 XP' below it. Below the reward box is a text area with instructions: 'Install the **git command line tools** so you can work with git repositories on your computer.' At the bottom left is a button labeled 'Let's Do This!', and at the bottom right is a button labeled 'HACK'.

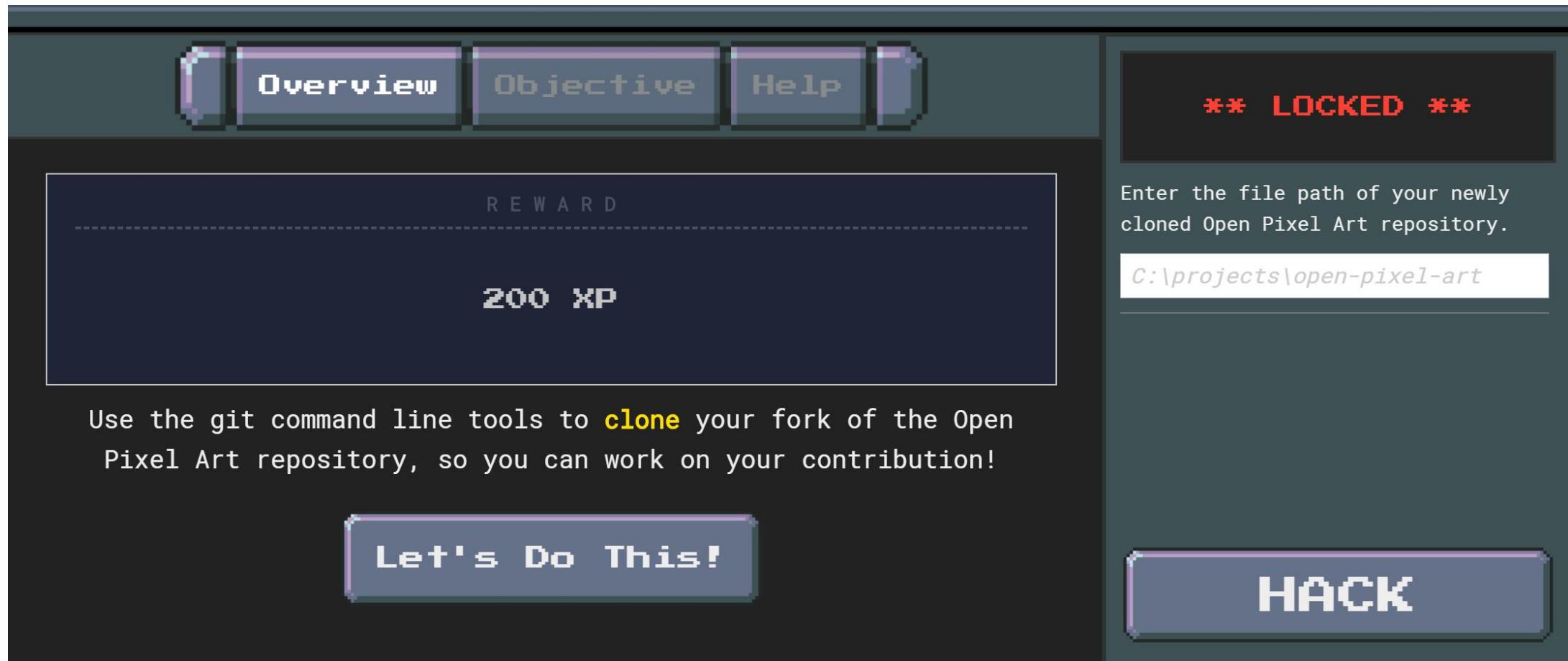
Screen 12



Screen 13



Screen 14



Screen 15



clone: <https://git-scm.com/docs/git-clone>

Screen 16



Screen 17



Screen 18

Contributing to Open Source: Make a Branch

Overview Objective Help

Branching Out

To complete this challenge, [create a branch](#) in your local git repository. You can use the `git branch` command for this purpose:

```
git branch my_branch_name  
git checkout my_branch_name
```

When you have created a branch, paste the name of your new branch into the text field on the right and click **HACK**.

Enter the name of the branch you create here!

dev-branch

HACK

Show Help

Create a branch: <https://git-scm.com/book/en/v2/Git-Branching-Basic-Branching-and-Merging>

Screen 19



Screen 20



Screen 21

Contributing to Open Source: Author Your Contribution

Overview **Objective** Help

Making Your Mark

Now, the fun part - making your changes to the project! To open this chest and claim the **Flame of Open Source**, you must edit source code in the [open-pixel-art](#) project and **commit your changes** on your development branch. See the "Help" section for instructions on how to make your change.

Once you have **committed your changes** to your development branch - [mybranch](#) - click the **HACK** button.

Show Help

**** FAILED ****

We didn't find a pixel in the `_data/pixels.json` file with your git username, stan6!

[**< Back to tutorial**](#)

Screen 22

Contributing to Open Source: Author Your Contribution

Pixel Time

Now that we have our project copied locally, and a development branch created, we are ready to make our contribution. But before that can happen, we need to set up the project and get it running! This way we can check out our changes on our own computer before pushing them back up to GitHub for everyone else to see.

Node.js and NPM

The Open Pixel Art project is build on top of Node.js and will use npm as well. This objective assumes you understand how npm works and have it installed.

If you're unfamiliar with these tools, follow this installation guide to get Node.js installed. After you have Node.js installed, you should be able to blindly copy and paste the npm commands in this objective.

Install the project dependencies

Now that you have Node.js and npm installed, in the main folder of your Open Pixel Art project run the following command:

```
npm install
```

This will download all of the code dependencies this project needs to run onto your computer. You should now see status messages as the dependenices install.

This process can take some time!

Screen 23

Boot it up

Now that our project is installed and configured, lets start it up locally. Run the following command in the root directory of your repository:

`npm start`

This will start a local web server on your computer that is running your own personal version of the Open Pixel Project. In the terminal, after the server has started you should see a URL printed that will usually be:

`http://localhost:8080`

Open this page in your web browser to see your own local Open Pixel Art project page live!

Pixels.json

Now that we can see our pixels locally, let's learn how to change them! Open the Open Pixel Art project in a text editor of your choice! Now, open up the file located at `_data/pixels.json`.

Screen 24

Inside Pixel.json file you'll find a JSON array of pixel objects that look like this:

The screenshot shows a terminal window with a dark theme. At the top, there are tabs labeled "Overview", "Objective", and "Help". The main area displays a JSON object:

```
"data": [    { "y": 1, "x": 1, "color": "#F22F46", "username": "<UNCLAIMED>" },    { "y": 1, "x": 2, "color": "#F22F46", "username": "<UNCLAIMED>" },    { "y": 1, "x": 3, "color": "#F22F46", "username": "<UNCLAIMED>" },    ...  ]}
```

Below this, the text "Pixel object properties" is followed by a dashed line. A bulleted list provides details about the properties:

- **x**: The x-coordinate of your pixel. 0 is the left-most column of pixels
- **y**: The y-coordinate of your pixel. 0 is the bottom-most row of pixels
- **color**: The color your pixel should have as a hex code (e.g. #ff0000 for red)
- **username**: The GitHub username you'll use to create the pull request

Screen 25

Commit

The screenshot shows a software window titled "Contributing to Open Source: Author Your Commit". The window has a dark theme with purple buttons labeled "Overview", "Objective", and "Help". A sidebar on the left contains the text "How do I commit?". The main content area explains the first step: adding files to be committed. It mentions adding the "_data/pixels.json" file and provides a command-line example:

```
git add _data/pixels.json
```

Below this, it says to verify the addition with the command:

```
git status
```

The final note states that the command will show which files are about to be committed in green, with a red note at the bottom: "The _data/pixels.json file should be the only one listed!"

Screen 26

```
git commit -m "feat(pixels): add my new pixel"
```

The screenshot shows a game interface with a title bar reading "Contributing to Open Source: Author Your Commit". Below the title is a navigation bar with buttons for "Overview", "Objective", and "Help". A terminal window is open, displaying the command "git status". The text within the terminal window reads:

This command will show you which files are about to be committed in green. The `_data/pixels.json` file should be the only one listed!

Now we need to use the `git commit` command to finalize our changes. We also add a `-m` flag to this command so that we can add a message that goes along with our commit.

The `feat(pixels) :` part before our message is added so that we the Open Pixel Art project can programmatically analyze commit messages. You can learn more about that by checking out the [Conventional Commits Standard](#).

At the bottom of the terminal window, the command `git commit -m "feat(pixels): add my new pixel"` is shown again.

Screen 26



Final Screen

