Course project: Implementation and Documentation

Overview

Make progress on the implementation of your project and solidify the documentation for end-users and developers.

Set up

Work with your project group to (1) make progress towards the final release, (2) provide a comprehensive user documentation, and (3) provide a comprehensive developer documentation.

Instructions

1. Make progress in your public GitHub repository

* Each group member must contribute to the code base.
* Each group member must demonstrate proper use of version control and CI.
* Each contribution (commit/pull request) must be tested, commented, and code reviewed.

Review and follow the [Git conflict resolution tutorial](https://rawgit.com/mernst/git-conflict-tutorial/master/git-conflict-resolution.html). When merge conflicts come up later in the quarter, you will be happy that you did.

**(1) Call the library and initial setup YuXiao Yao**

**(2) Circulation---gamerover function JunWei Zhang**

**(3) Snake and raspberry(The theme of the game, the snake and snake berry**

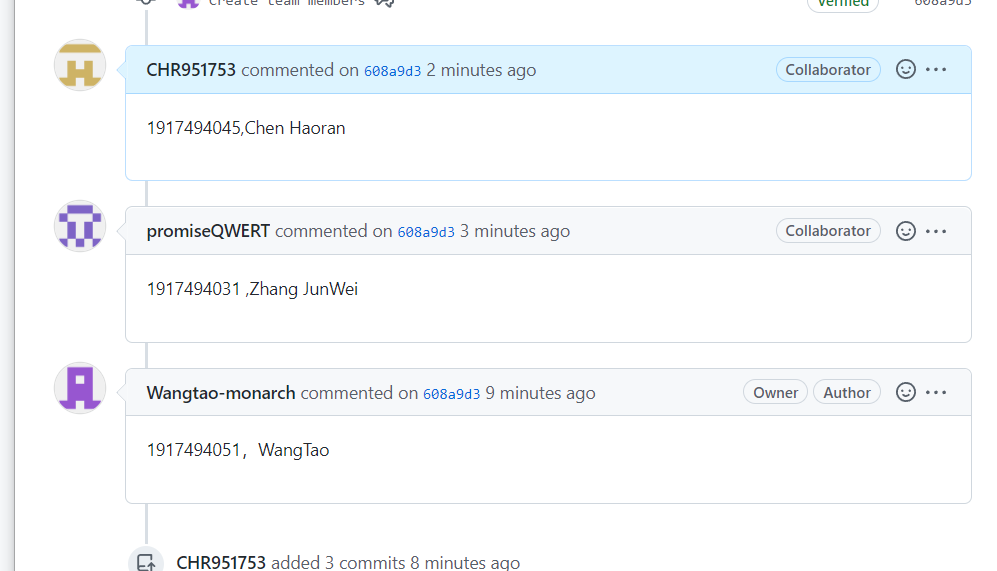
**display and movement) HaoRan Chen—Tao Wang—PengCheng Yang**

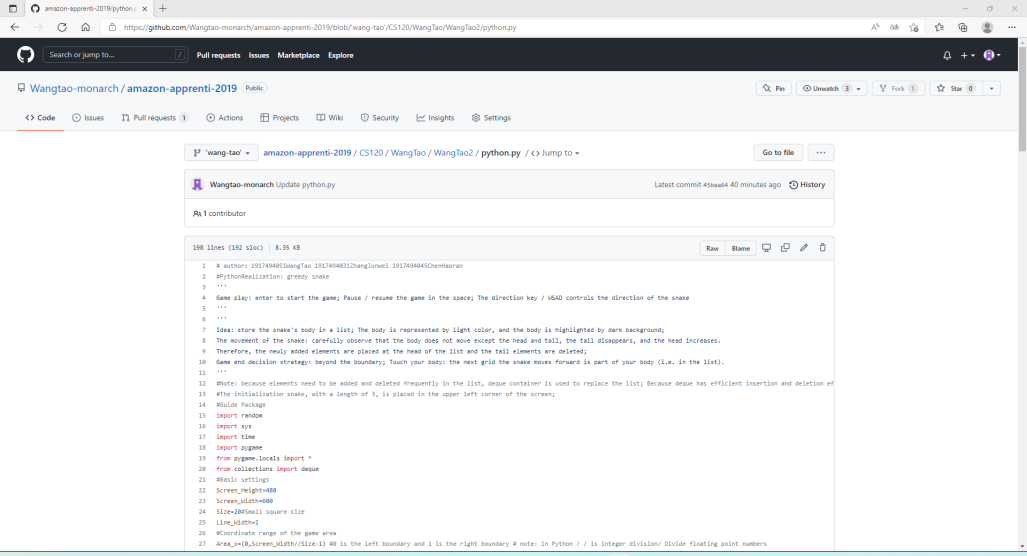
1. **commit and pull request are tested, commented, and code reviewed.-**

**Tao Wang,Haoran Chen,Junwei Zhang.**

**We have uploaded the code to the GitHub folder, and the link is as follows:**

**<https://github.com/Wangtao-monarch/amazon-apprenti-2019/blob/'wang-tao'/CS120/WangTao/WangTao2/python.py>**

****



2. Write user documentation

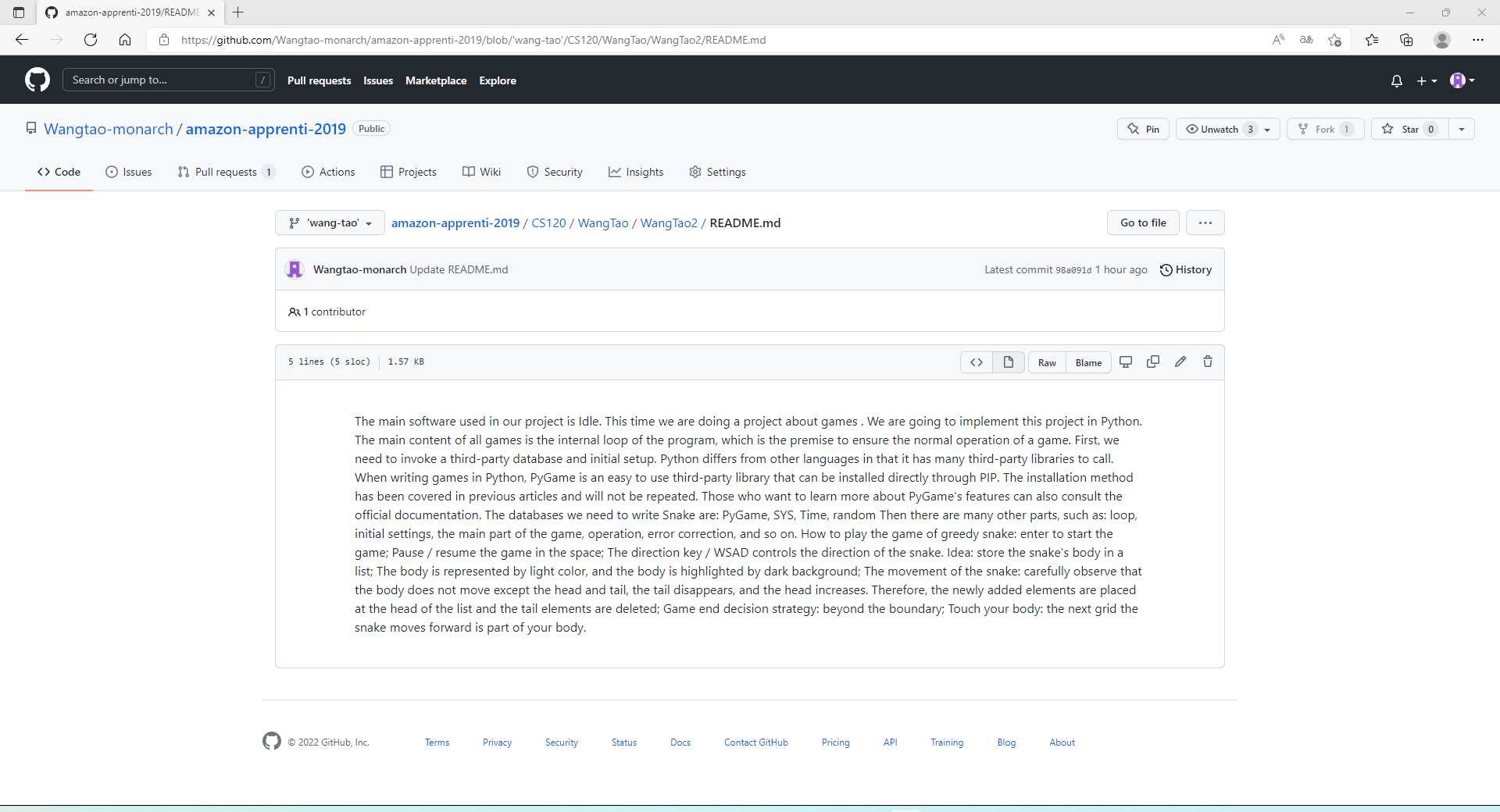
Your public repository must contain a complete user manual. Anyone looking at your repository should be able to easily find the user manual. **The user manual is focused solely on people who want to use your project**.

The user manual should describe the functionality of your project as you expect it to be at the end of the quarter. For this assignment, **indicate missing functionality as work in progress**.

The user documentation should include at least the following information:

This is a casual game that allows users to feel relaxed.

How to play the game of greedy snake: enter to start the game; Pause / resume the game in the space; The direction key / WSAD controls the direction of the snake. Idea: store the snake's body in a list; The body is represented by light color, and the body is highlighted by dark background; The movement of the snake: carefully observe that the body does not move except the head and tail, the tail disappears, and the head increases. Therefore, the newly added elements are placed at the head of the list and the tail elements are deleted; Game end decision strategy: beyond the boundary; Touch your body: the next grid the snake moves forward is part of your body.



https://github.com/Wangtao-monarch/amazon-apprenti-2019/blob/'wang-tao'/CS120/WangTao/WangTao2/README.md

* **A high-level description.** What does the system do and why would a user want to use it.

How to play the game of greedy snake: enter to start the game; Pause / resume the game in the space; The direction key / WSAD controls the direction of the snake. Idea: store the snake's body in a list; The body is represented by light color, and the body is highlighted by dark background; The movement of the snake: carefully observe that the body does not move except the head and tail, the tail disappears, and the head increases. Therefore, the newly added elements are placed at the head of the list and the tail elements are deleted; Game end decision strategy: beyond the boundary; Touch your body: the next grid the snake moves forward is part of your body.

* **How to install the software.** If your system has prerequisites (e.g., tools, libraries, emulators, third-party applications, etc.), your instructions should list all of them and indicate how to install and configure them. Make sure to indicate what specific version requirements these prerequisites must satisfy. If running the system requires the installation of, e.g., a virtual machine, a database, or an emulator, make sure to provide clear step-by-step instructions.

We need to install idle and open python.py.

* **How to run the software.** How to start up the system?

The main running software we need is Idle.

* **How to use the software.** You can assume that your user is familiar with your particular platform (e.g., use of a Web browser, desktop applications, or mobile applications). For missing functionality, your documentation should simply indicate that this functionality is work in progress.

We need to write or import code in idle.Then we need to compile and run

* **How to report a bug.** This should include not just the mechanics (a pointer to your issue tracker), but also what information is needed. You can set up a bug-report template in your issue tracker, or you can reference a resource about how to write a good bug report. Here is an [example for bug reporting guidelines](https://developer.mozilla.org/en-US/docs/Mozilla/QA/Bug_writing_guidelines).

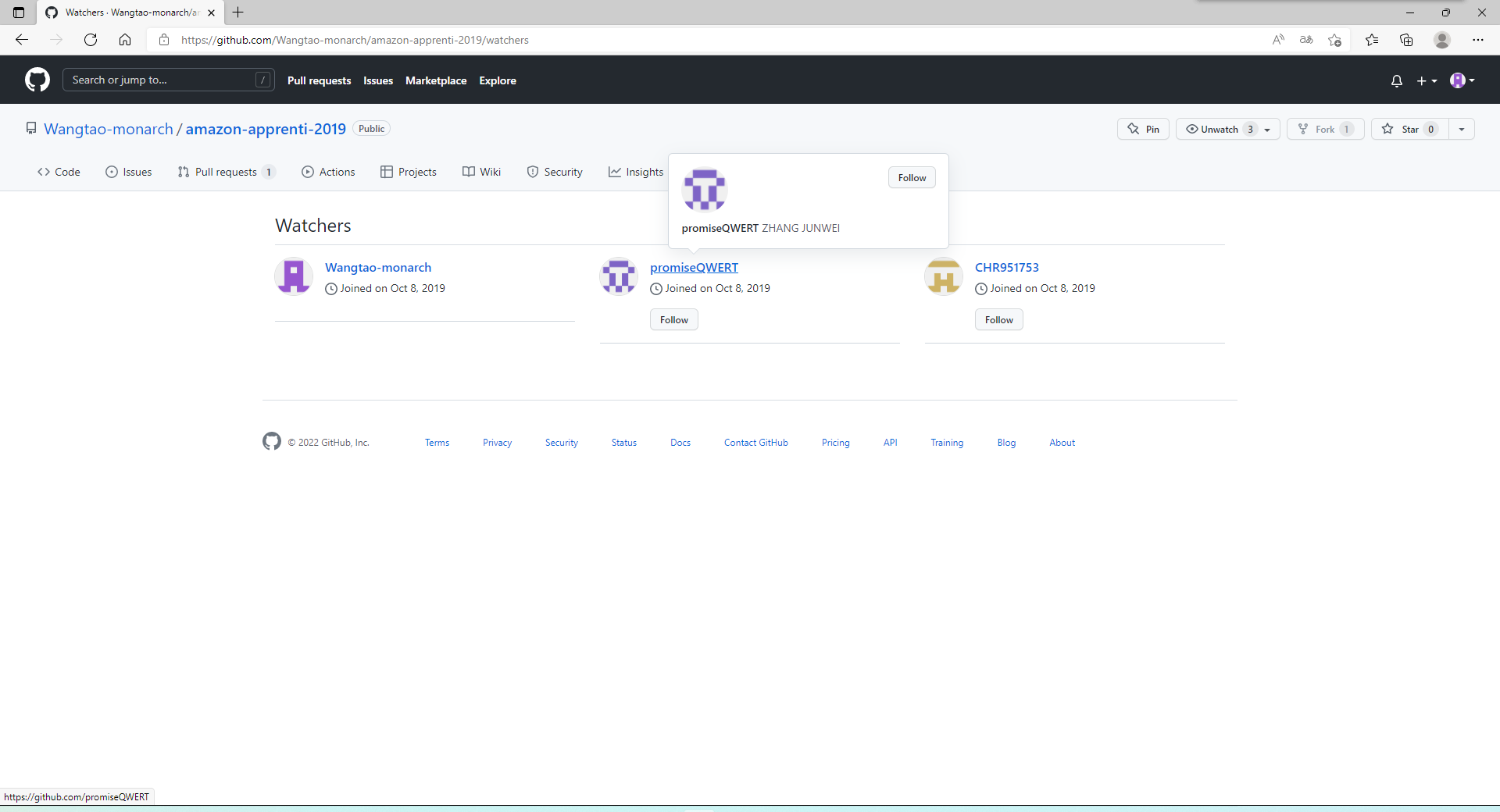
Bug fix: If the snake is moving to the right, quickly click to apply down and left commands respectively, the down command will be overwritten, only the left command is accepted, direct GameOver.

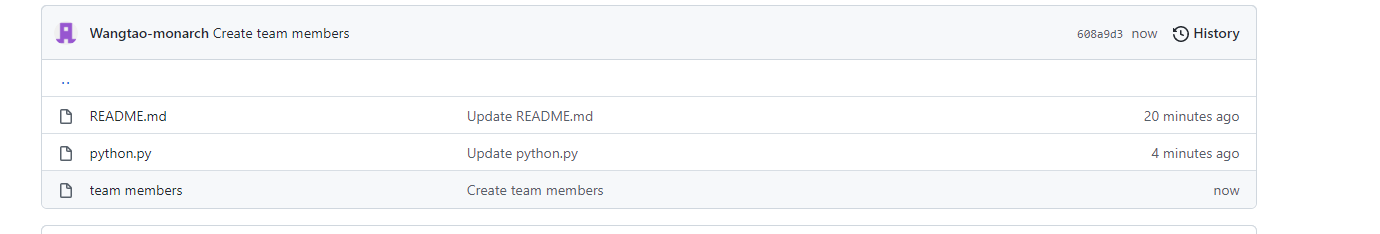
* **Known bugs.** Known bugs or limitations should be documented in the bug tracker. A user testing the implemented use case(s) should not encounter trivial bugs (e.g., NPEs) or a large number of bugs that are unlisted in your bug tracker.

If the snake is moving to the right, quickly click to apply down and left commands respectively, the down command will be overwritten, only the left command is accepted, direct GameOver.

3. Developer documentation

Your public repository must contain developer guidelines. Anyone looking at your repository should be able to easily find these guidelines. **The developer guidelines are focused solely on people who want to contribute to your project**.

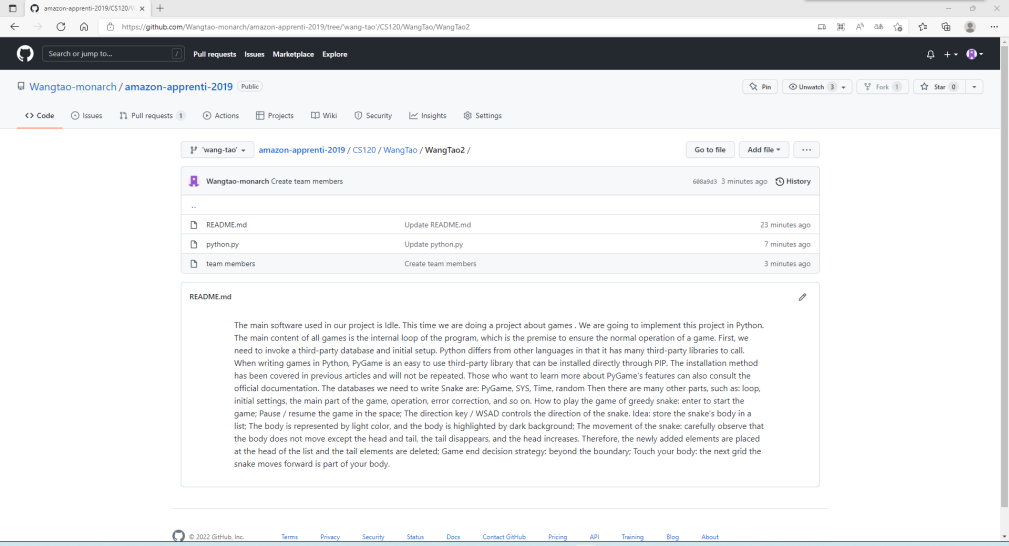


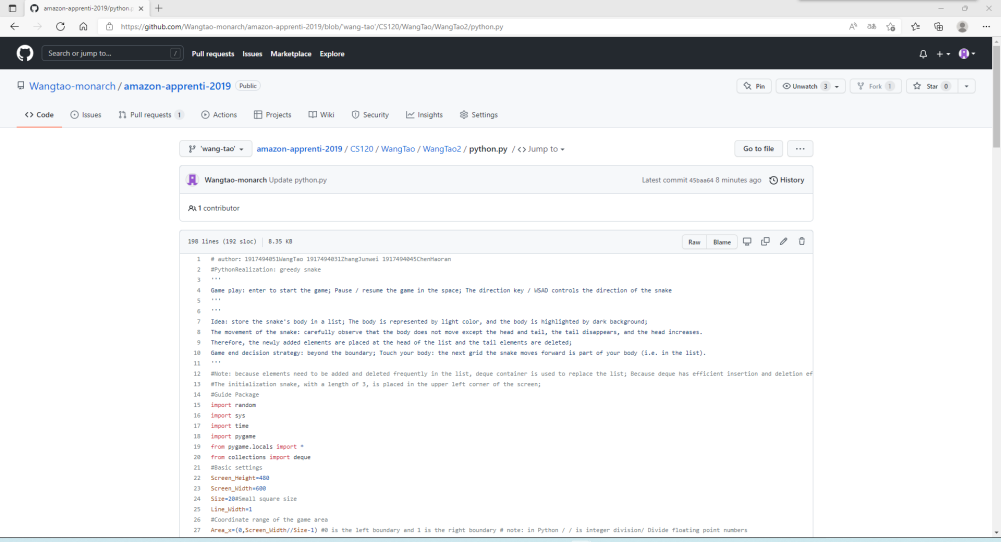


The developer documentation should include at least the following information:

* **How to obtain the source code.** If your system uses multiple repositories or submodules, provide clear instructions for how to obtain all relevant sources.

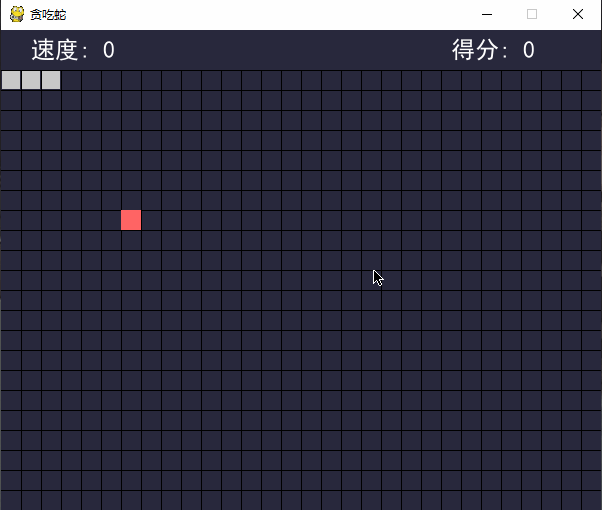
First enter this website https://github.com/Wangtao-monarch/amazonapprenti-2019/tree/'wang-tao'/CS120/WangTao/WangTao2, and then open python.py, you can get the code.





* **The layout of your directory structure.** What do the various directories (folders) contain, and where to find source files, tests, documentation, data files, etc.

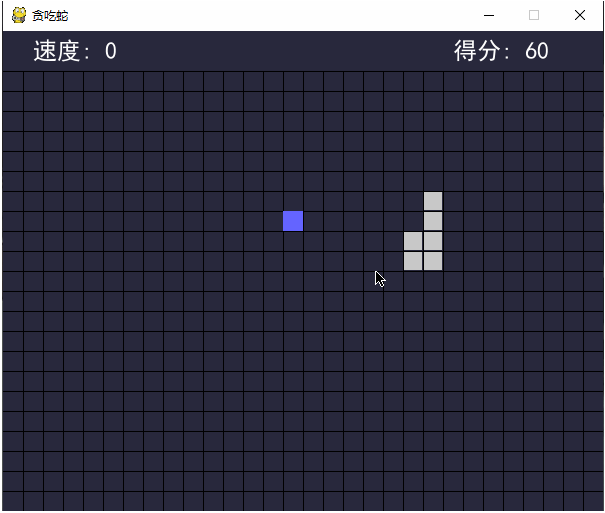
Read. Use me to tell users what software we want to do and some functions; python. Py is our source file; Team numbers is a member of our team

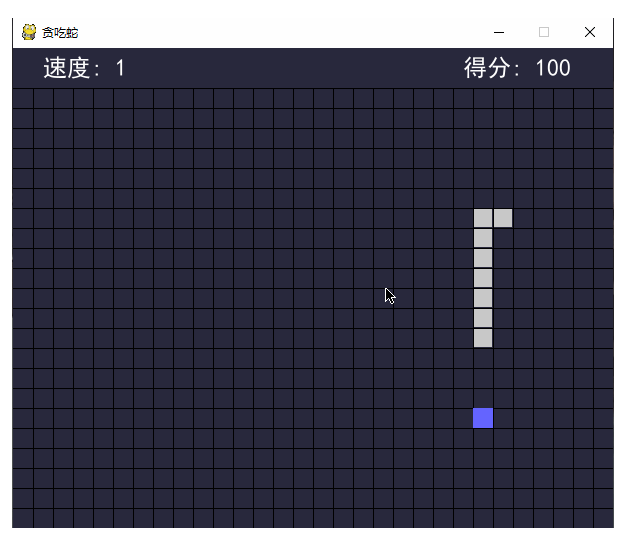


* **How to build the software.** Provide clear instructions for how to use your project’s build system to build all system components.

The main software used in our project is Idle. This time we are doing a project about games . We are going to implement this project in Python. The main content of all games is the internal loop of the program, which is the premise to ensure the normal operation of a game. First, we need to invoke a third-party database and initial setup. Python differs from other languages in that it has many third-party libraries to call. When writing games in Python, PyGame is an easy to use third-party library that can be installed directly through PIP. The installation method has been covered in previous articles and will not be repeated. Those who want to learn more about PyGame's features can also consult the official documentation. The databases we need to write Snake are: PyGame, SYS, Time, random Then there are many other parts, such as: loop, initial settings, the main part of the game, operation, error correction, and so on. How to play the game of greedy snake: enter to start the game; Pause / resume the game in the space; The direction key / WSAD controls the direction of the snake. Idea: store the snake's body in a list; The body is represented by light color, and the body is highlighted by dark background; The movement of the snake: carefully observe that the body does not move except the head and tail, the tail disappears, and the head increases. Therefore, the newly added elements are placed at the head of the list and the tail elements are deleted; Game end decision strategy: beyond the boundary; Touch your body: the next grid the snake moves forward is part of your body.

* **How to test the software.** Provide clear instructions for how to run the system’s test cases. In some cases, the instructions may need to include information such as how to access data sources or how to interact with external systems. You may reference the user documentation (e.g., prerequisites) to avoid duplication.

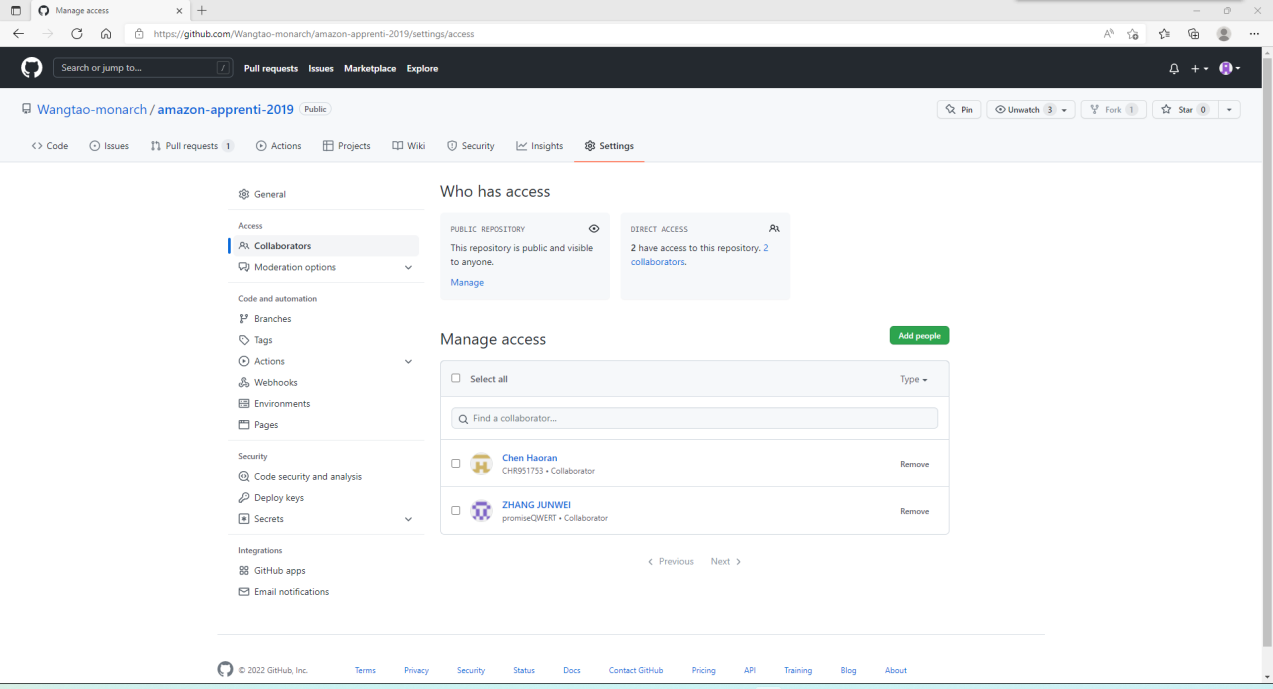






* **How to add new tests.** Are there any naming conventions/patterns to follow when naming test files? Is there a particular test harness to use?

I authorized the team members



COLLABORATORS:WANG TAO ;ZHANG JUNWEI ; CHEN HAORAN

* **How to build a release of the software.** Describe any tasks that are not automated. For example, should a developer update a version number (in code and documentation) prior to invoking the build system? Are there any sanity checks a developer should perform after building a release?

The developer should update the version number before calling the build system, and after building a distribution, the developer should perform any sanity checks.

Finally, our team members paid a lot of energy to implement this program. I hope this game can make more people feel relaxed in their busy life. I do believe this game will be better and better.

Assignment Submission:

Email link to your team GitHub Project Repository to our course TA.

The subject of the email should be: [Your StudentID, Assignment Name]

Clarifications

*What if info needed in my user or developer documentation already exists in my living document?*

The documentation in your repository should be largely self-contained, and anyone looking for documentation should find it, or a link to it, in the repository.

You may link to sections of your living document from your user/developer documentation, or you may move these sections to the repository and put a link to it in your living document.

Do not maintain two parallel versions of any documentation.