Major Pathway

Fanghui He, Yining Hua, Qiaqia Ji, Yanwan Zhu

Introduction

When transitioning to colleges, students are expected to find all related information at the department website on their own. Although the website is a good resource, some further explanations are necessary to provide students with a big picture of their education.

With the goal of informing students in an interactive way, we want to build an RPG game to help students better understand the paths through Computer Science majors and minors at Smith. Ideally, our players are expected to have a basic idea about what the CS Department at Smith is like.

Discussion of the implementation

We divided our game into four major different classes: the Character class, the Professor class, the Computer class, the Objects class which has a subclass a similar class - the NPC class. Each class handles an important part of the game.

In the Character class, we store basic information about the main character, such as the image, the position coordinate, the major declaration information,etc. We have implemented the move method, the switchIMG method which lets the character to move rather than to drift. Other methods in the Character class are used to check or update specific information about the character.

In the Professor class, we have a constructor that creates professor instances of different professors and have functions that handle the interactions with every professor, such as chatting, advising, declaring major, etc.

In the Computer class, we process the academic information of the character, such as course catalog, class schedule, academic year, semester, and practice problems to raise GPA. Specifically, the users can create the practice problems themselves. Simply create a new txt, and add the name of the .txt to the Questions.txt under the course key.

The Objects class only contains a constructor, since some objects only need a constructor because the scene logic involving them is implemented directly in the interact() function. The NPC class is a class similar to the Objects class but without argument name, and it has a chat function. The door class is a subclass of the Objects class in which we have an openDoor() method which shifts the scene to offices.

The interact function checks what the current scene is if the character is going to approach something in the particular scene, and update the scene before it changes the scene. We also have a bunch of helper functions for the interact() function like drawScene(), checkElevator(), resetScene() and etc.

Description of Programming Techniques

We chose to use a combination of object-oriented programming and functional programming because of the nature of the game. We used a global variable “scene” to monitor where the main character is going. We used dictionary to handle the course catalog and course schedule. This data structure might not be super efficient, but was very intuitive. For efficiency purposes, we might want to choose HashSet or TreeSet, but since the course catalog is not very large (the difference between efficiencies is not that terrible), and we wanted the course schedule to be displayed in the order we register the class, we finally chose to use the dictionary.

System's performance

Since our project is a game, we’ve tried to optimize the user experience from user preferences. For example, our game allows users to select their preferred keyboard control system. In addition, we have implemented staircase and elevator to mimic the real Ford. We have smooth animations and scene shifts, as we avoid creating tons of objects and manage to make good use of instances that we have created.

Back to the Real-world Context

We have implemented functions such as chatting with professors, chatting with other students(NPCs), advising, checking course information, and practice courses in order to graduate. So the game structure is basically completed and functions are all realized, but definitely more information about courses and professors are needed.

Shortcomings and Future Improvement

1. Every time when the character switches from one scene to another, there'll be a shift of the position, namely that the character will stay at the same position on the previous scene, and will go back to the right position only after you press a key.
2. The character can only take a maximum of four classes every semester. After the character registers the classes, and takes all the classes he or she registered in the semester, the semester variable will increase. But the character cannot retake any courses he registered in the previous semesters. Otherwise the game will update the semester.
3. For the GPA of the character, we want to develop a more precise method to calculate the GPA. Since now we only have one question for a course, when the character answers the question wrong, GPA becomes 2.7, and it becomes 4.0 if the character answers the question correctly.

Reference:

Tkinter:

https://wiki.python.org/moin/TkInter

https://stackoverflow.com/questions/38745738/return-value-using-button

https://stackoverflow.com/questions/35003476/opencv-python-how-to-detect-if-a-window-is-closed

Read in text as dictionary:

https://stackoverflow.com/questions/4803999/how-to-convert-a-file-into-a-dictionary

https://kaijento.github.io/2017/04/09/python-convert-a-text-file-into-a-dict/

https://stackoverflow.com/questions/9314824/python-create-dictionary-from-text-file-thats-in-dictionary-format

Contributions:

Ning:

- Art
- Scene logics and helper functions
- Conversations

Fanghui:

- The Professor class
- The Computer class

Yanwan:

- The Character class
- Readme and change log

Qiaqia:

- The Computer class