Team FooBar()

Report

Version 2.0

What have you and your team learned from this experience?

As a team we have learned the following from this experience:

- We learned the Python programming language, an interpreted, object-oriented, and extensible programming language.
- We learned how to better use integrated development environments and other useful tools to develop the software and construct documentation.
- We learned, through trial and error, how to correctly use version control technologies.
- We learned how to use non-standard libraries, such as utilizing PyGame to build the graphical user interface for the game.
- We learned how to obtain the solutions to problems by reading information available through online resources, such as questions and answers from stack overflow and articles posted on Red Blog Games, and applying them to our own project.
- We learned how to utilize assets from online sources, such as OpenGameArt.com, to incorporate sound, music, and art into the game.
- We learned how to use the strengths and weaknesses of each team member to assign tasks that complement that team member's skills.
- We learned how to distribute tasks in a way that no one felt over- or under-worked.
- We learned that being a team leader assumes responsibility for all members of the team: adapting roles as necessary (sound designer, art designer, game designer, developer, tester, etc.) in order to make any improvement or fill any deficiencies in the product.

How do you feel this will help you out in the industry?

We now have the experience of working with one of the most popular computer programming languages in the world, the knowledge of implementing new features in a software product by using libraries written by other people, the ability to work cohesively and effectively as a team with multiple members that have varying degrees of skill and wisdom, and the capabilities of explaining the work we have completed in a clear, concise, and precise manner. This provides us with everything necessary to obtain a job in the industry and allows us to become valuable assets to any company or organization that may hire us.

What was the hardest and what was the easiest part of making version 2.0?

The hardest part was testing the game to ensure that there were as few noticeable defects as possible for the presentation. Each time we made a change, we had to restart the testing process. This is time consuming and problematic due to the fact that it was necessary to determine whether or not the change we made actually fixed the problem and if new problems were created in the process. The easiest part was listening to music and sound effects to determine what sounded right in the game and looking at art to determine what looked right in the game. The only difficult part of that was stopping ourselves.

What encryption algorithms did you use?

For encryption, we chose to use Caesar Cipher for the simple algorithm and AES (Advanced Encryption Standard) for the complex encryption. For the third algorithm, we simply hard-coded a value to represent another value, which represented which algorithm was used for encryption.

Did you have weekly meetings? If so, on what day(s)?

Meetings were held as we saw fit, mostly on Tuesday's or Thursday's at 3:30pm. We did meet once a week, with the exception of one or two weeks.

Team FooBar()

Report

Version 1.0

What language did your team use?

FooBar() is using the Python programming language. We used PyGame for the GUI implementation.

What unit testing framework did your team use?

We used PyUnit for our unit testing.

What IDE's and what development OS's did your team use?

Our team used Microsoft Visual Studio as well as PyCharm on Windows and macOS.

What have you learned from this experience so far?

We have learned the following from this experience:

- The A* algorithm is very difficult to implement
- Python is a sweet language
- Debugging takes longer than initial development
- We should not make assumptions with requirements or how we think our code is working
- Programming for hours and hours straight is not an ideal situation; it makes you want to drown yourself in a bath tub…this project made me want to drown myself in a bath tub filled with orange Gatorade.
- Cooperation is very vital to success
- Effective communication is necessary for cooperation
- We must be anticipating change and be able to adapt to changing requirements
- Agile development

How do you feel this will help you out in the industry?

This project will definitely help us with working on a team and communicating effectively. The project has also helped us understand assigning tasks. By evaluating each individual's skills and taking into account each other's workloads, we were able to do a better job of assigning tasks. With these individual tasks, it taught us the individual responsibilities and accountability of working on a team. For the management side, this project will help in the industry if we were to ever receive a management position. Being in the shoes of the developers and also having to do a little managing, we will be better prepared. Deadlines are another big thing that this will help with in the industry. With Python being one of the leading languages in the industry, learning this language has increased our value as employees and will potentially lead to more job opportunities.

Did this assignment help link the course material to real world application?

This assignment was a good learning experience that helped us apply what we have learned in the course to a real-world application of developing a game.

What was the hardest and what was the easiest part?

The hardest part of this project was refactoring the code for optimization and implementing the A* algorithm. The easiest part of this project was implementing the random (dumb) enemy.

What types of maze randomizer algorithms did your team implement?

We used the Binary Search Tree and Sidewinder maze randomizers.

What type of shortest path algorithm did your team implement for the enemy?

For our smart enemy, we used the A* algorithm.

How did you accomplish each of these applicable guidelines for the project? If they are not applicable, why are they not applicable?

Limit the visibility of information in a program

Python programming language does not have private/public variables or functions.

Check all inputs for validity

We only executed code if the user input matched the predefined string for the corresponding command. If the user inputted a string that did not match, an error message was printed, informing the user that the command could not be recognized. For the variable-length go command, we checked each sub-string for validty, using the isalpha() and isdigit() functions to determine if the sub-string contained only characters or digits, before executing the proper command.

Provide a handler for all exceptions

We manually handled exceptions rather than using predefined try and except functions.

Minimize the use of error-prone constructs

We successfully minimized the use of error-prone constructs by not using any error-prone constructs.
Provide restart capabilities

There is not a need to provide restart capabilities for this iteration because it was not a specification.

Check array bounds

We made sure that no object could be rendered beyond the screen (bounds of the grid).

Include timeouts when calling external components

This is not applicable for this iteration because there are not external components being communicated with.