Defuse the Bomb
A CSC 102 Project

Team: <Rohan Khanal, Ricardo Almeida, Murat Yildiz>

GitHub:
https://github.com/XNetoGaming/Group-repository.git

BOMB DEFUSAL WRITE-UP

Version 1

Verification Code: <enter your code>

Team individualization

What did you tweak to the design provided by your instructor that makes it different from the other teams? In other words, what did you do to make your version of the “bomb” unique?

The ways in which we made our bomb unique and different was the following:

- For the toggles, we made binary conversion problems for the user(s) to convert in which the answers could be from 1 to 15, as there are 4 toggles so the max would be 1111 or 15. These types of problems consist of what we learned in class with binary addition and multiplication.
- For the keypad, instead of using cryptography to decrypt words and messages, we chose to stick with quantity, so math and numbers. This tool will take users the most amount of time to figure out, so we made it so that multiple people can do it. We created a very complex binary multiplication equation so that when they find the answer in binary, they will have to convert it into a normal digit number and type it into the keypad. We made the # key be the delete key, for if they type in a wrong digit, and the * is the enter key, for when they are ready to submit their answer.
- Lastly, for the wires, we chose to do some multiple choice questions where the answers will be whatever wire the user(s) must “cut” or pull out. In other words, the answer choices will be A B C D E which is correlated to the wires from left to right; the leftmost wire is A and it goes in alphabetical order to the rightmost wire being E. We made most of the questions computer science related besides one which is for this university. So for this part, there will be a question displayed on the screen for the user. If they answer is wrong, the bomb will “blow up”, but if the answer is correct, then the machine will move on to the next question. This time, the next question will again have the same multiple choices but minus 1 because one of the wires is now gone. There will be a total of 3 questions for this part for the user to answer for them to successfully disarm the bomb.

The buttons were nearly impossible for us to figure out so we didn't do anything. We kept it the same as the original bomb game

Future development plans

If you were to continue working on this project, what would you do? Where could you go from here to make it better, more interesting, more fun? What could be done to increase the project's broader impact (e.g., to make it marketable)?

If we were to continue with this project, we would most likely use the keypad more for trivia questions, riddles, or any other types of questions that require a one-word response rather than just numbers and binary arithmetic operations. Either that, OR, we could just stick with the alphabetic substitution decipher and use encryption to decode a keyword and/or message. Another idea that we could do next time, is to use the buttons for True and False questions asking about anything in general, so trivia. We would have to set green for True, red for False, and then for blue, we could say "neither" or "it depends". And finally, if we were to make our project marketable and increase its broader impact, we could maybe charge people a fair amount every time to play our game and also maybe do more with the toggles and the buttons, just to add more detail and creativity to it.

Lessons learned

What did you learn by working on the project throughout the course? In your opinion, did it relate to *The Science of Computing* curriculum (and, if so, how)? How was the experience beneficial to problem solving in general? What did you learn that will benefit you in future courses in the Computer Science curriculum?

By working on this final project, we learned how to make a bomb and defuse it. We think this does relate to our class curriculum, because the source coding is much more complex and as a class, we really had to analyze the source code of the bomb and understand all the syntaxes and functions, what they do, and what parameters we need, etc. In 101, we were taught the basics of programming in Python and the vast terminologies of certain ideas and processing. Even learning it back then did have obstacles and hurdles here and there, but then we've realized that it's only going to get harder from here. This experience was very beneficial to problem solving because it requires us to really use our brains very carefully when trying to decode certain puzzles/algorithms.

In future computer science classes, I think what we learned that will benefit us in the future is probably going to be all the different coding styles that were unknown to us at first because this year, we've learned a lot of new python libraries and functions and the tasks that they do. We also learned that we should always be creative when working on projects like these and come up with our own ideas. Maybe also take some inspiration from other people, but still make it yours. I also may have learned that even the silliest ideas may even turn out to be really good ideas and so we should frequently say what's in our minds.