Defuse the Bomb

A CSC 102 Project

Team: Rohan Khanal, Ricardo Almeida, Murat Yildiz

BOMB DEFUSAL MANUAL

Version 1

Verification Code: <enter your code>

The Game

This project is based on the game **Keep Talking and Nobody Explodes**[1], a cooperative bomb defusing party game. As the game designers put it, “You’re alone in a room with a bomb. Your friends, the 'Experts', have the manual needed to defuse it. But there’s a catch: the Experts can’t see the bomb, so everyone will need to talk it out - fast! Put your puzzle-solving and communication skills to the test as you and your friends race to defuse bombs quickly before time runs out!”

Their version is a software game. Our version takes the idea and realizes it as a physical device with buttons, switches, and more! Although our version can be played just like theirs, players can interact with both the bomb and this document at the same time (i.e., players can both defuse the bomb and serve as the “Experts”, using this document to help disarm the phases).

The backend of our version of the game is a Raspberry Pi[2] computer that combines a typical computer with the ability to interact with the outside world through sensors. The underlying software is written in Python[3] and is the result of a final group-based project in CSC 102 (The Science of Computing II) in the Computer Science Program at the University of Tampa.

Defusing Bombs

The bomb will “explode” when its countdown reaches 0:00 or when too many strikes have occurred. You defuse the bomb by disarming all of its “phases” before the countdown expires.

[1] https://keeptalkinggame.com/

[2] https://www.raspberrypi.com/

[3] https://www.python.org/

Phases

The bomb has four phases, each of which must be disarmed to defuse the bomb. The phases can be disarmed in any order. Once a phase is disarmed, it becomes inactive, and changing it doesn't affect the bomb. Instructions for disarming the phases are provided in this document.

Strikes

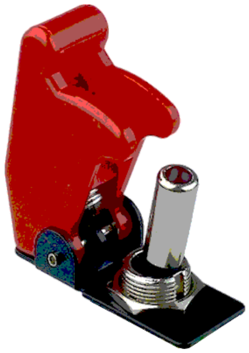
A mistake in disarming a phase results in a strike. Get too many strikes, and the bomb “explodes. A sound will be played each time the user makes a mistake and the available strike counter is decreased by 1. The user fails if he exhausts all the 5 strikes and sound will be played for failure

Information

A different version of the bomb is randomly presented each time it is “booted”. There are 6,720 unique versions of the bomb with a whopping 1,176,000 possible variations!

Disarming some phases will require specific information about the bomb. Pay close attention to the “bootup” text on the bomb's screen.

Regarding the Toggles



It's so tempting to just toggle the switches over and over with those bright red LEDs and cool switch covers that you can flip. But one wrong toggle gets you one step closer to...BOOM!

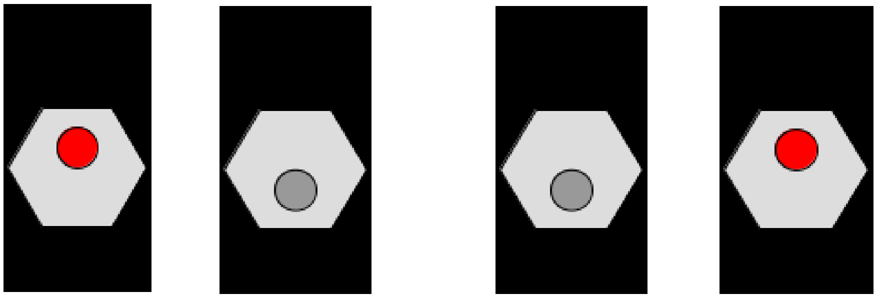
The correct state of each toggle switch is based on the bomb's serial number.

You must first add the numeric digits in the serial number together to obtain a target value. Convert this value to a 4-digit binary number. Toggle the switches to represent the binary number.

Converting a number to binary (base 2) can be done by placing a 1 in the appropriate powers of two represented by the columns of the table below that, when added together, sum to the value. A 0 is placed in the remaining columns. The leftmost digit of the binary number is known as the MSB (most significant bit), while the right-most digit is known as the LSB (least significant bit).

2^3	2^2	2^1	2^0
8	4	2	1

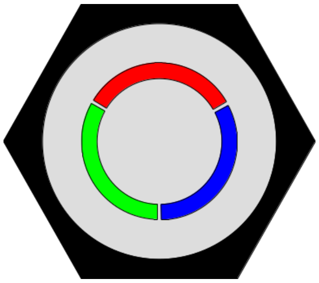
The left-most toggle switch represents the MSB, and the right-most toggle switch represents the LSB. The LED on a toggle switch lights up to represent a binary 1. Use the diagram below to assist you (which, by the way, represents the value 9).



The following model above shows an example of what 9 would look like with the toggles on in red representing as a 1 and the toggles that are switched off representing as 0.

Toggles are a binary so the user should know how to convert a normal digit number into its binary counter part, but how it works is the user is met with a math equation to solve which the answers could be from 1-15 as there are 4 toggles and the max is 1111 or 15 in binary so tell the users that when the toggles are off or down that represents a 0 and when the toggles are up that represents 1, for the description just tell them how to convert so if the answer is 10 you can teach them either the addition method or the division method whichever way you think that would be easier to the user

Regarding the Button



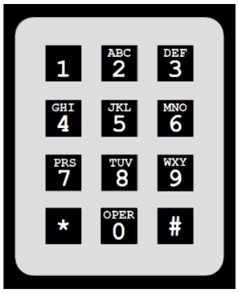
The button behaves in unpredictable ways. Follow the instructions below closely to avoid a strike!

At some point, you will need to press the button. However, releasing it is the hard part. The button has a lighted ring around it that can be red, green, or blue.

Release the button according to the following instructions:

Button color	**Release instructions**
Red	Release the button at any time.
Green	Release the button when the first numeric digit found in the bomb's serial number appears anywhere in the seconds of the countdown timer.
Blue	Release the button when the last numeric digit found in the bomb's serial number appears anywhere in the seconds of the countdown timer.

Regarding the Keypad



Ooooh, an encrypted phase! Press the correct keys on the keypad carefully to avoid a strike. Try to avoid calling the "operator".

Important information about the keypad is provided in the bomb's "bootup" text.

The correct combination can be determined by performing a complex binary multiplication equation and then converting that binary result into a normal digit number and typing it into the keypad. The # key is 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. Using this tool may require more than one person to do it because it will take up the most amount of time.

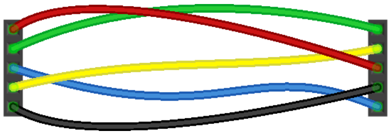
Binary numbers are base 2 and they only have a 0 and a 1 as bits. For example, the number 1 is… well it's just 1. 2 is 10 in binary, which is bit 1 for the 2^1, and bit 0 for the 2^0. You can refer to the table below that is listed of all the binary bases and their values; i.e. [1, 2, 4, 8…] etc. In order to add numbers, you would have to focus only on the bits that contain a 1. Ignore all the 0's. For example, if the user who is playing the bomb kit receives "1001101" as their answer, then they would convert that into a real number by going from right to left (or whichever way they'd prefer) and adding the 1 bits with the correct corresponding bases. So for 1001101, they would have to add 1 + 4 + 8 + 64 to get their answer as a normal digit and input it into the keypad phase.

Binary Base (2)	Binary Digit (8 bits)	Value
2^0	0000 0001	1
2^1	0000 0010	2
2^2	0000 0100	4
2^3	0000 1000	8
2^4	0001 0000	16
2^5	0010 0000	32

2^6	0100 0000	64
2^7	1000 0000	128

To enter the passphrase correctly, you must enter its numeric combination on the keypad. To do so, press each button on the keypad with the required letter only once.

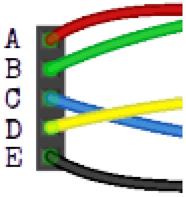
Regarding the Wires

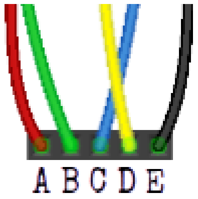


Which wires should you "cut"? One wrong "snip" leads you one step closer to an "explosion"!

There will be multiple choice questions that we have added to be displayed on the screen for the user to answer. The correct wire to "cut" or pull out is based on the correct answer choice to the question.

We have made the answer choices be the wires, which are letter-labeled as follows from left to right, depending on their orientation on your bomb:





Note that the actual color of the wires doesn't matter, and the color of the wires on your bomb may be different from those in this document.

"Cut" the wires based on the correct answer choice to each question according to the following instructions:

Answer Choice	**"Cut" instructions**
5	"Cut" the wire that is correlated to the right answer to the question that answer choice will not be able to be selected again.
4	"Cut" the wire that is correlated to the right answer to the question that answer choice will not be able to be selected again.
3	"Cut" the wire that is correlated to the right answer to the question that answer choice will not be able to be selected again.

2	“Cut” the wire that is correlated to the right answer to the question that answer choice IF WRONG BOOOMMMMMM!