**Lab 06: Bop-It!**

# **Objectives:**

* Develop problem-solving skills
* Develop skills using multiple loops
* Practice writing whole programs from the ground up

# **Starting Point:**

* [**lab06.c**](https://drive.google.com/open?id=1kdIa7Khhwd5S4nMuIw-twi8MWs3jdJy6)

# **Turn-In:**

* **Upload one .pdf file** containing the following before the start of lab next week.

1. Source code that you have written.
2. Answers to the questions at the bottom of this document.
3. **Upload one .zip** file of your **lab06** folder. Title this **firstname\_lastname\_lab06.zip**. Before zipping, delete all generated **.exe** files (not the **ds4rd.exe**).

**Process:**

In this lab, you will be implementing a simple “Bop-it!”-like game. Your program will run for a single game. Start with implementing a “start menu”, output a line asking the user to push a button to start and wait for the user to push a button.

The game will involve printing a line telling the user which button to press (chosen randomly by the program) and will wait a certain time for a response. After each successful action by the player the time the program will wait for a response will reduce. The game ends when the player does the wrong action or time runs out, (finding time passed may not be as easy as you think, **hint**: you may want to keep track of a previous time value and compare that to the newly scanned-in time).

**Note:** You should ensure that button presses are only registered once per press.

**Requirements**

* Program prints message telling user to push a button to start game.
* Once in game, output tells player which button to push. The button to press should be determined randomly.
* Game runs continuously until user reaches lose condition.
  + A wrong button is pressed
  + Time runs out
* Program waits for user action for less time after each iteration.
* Program outputs how many successful actions player executed at end.

## **Questions:**

1. How did you randomize the buttons that needed to be pressed?
2. What game states, if any, did you keep track of?
3. What mechanism did you use to make sure extraneous button presses were not registered?

## **Ending the Lab Session:**

1. Be sure you will be able to have access to your code and data when you work on your lab report.
2. Leave your source code on your U: drive.
3. **Plug in the DS4 into its charging station!**
4. Log off before leaving your workstation in the lab!

## **Sample Output:**

**Round with wrong button pressed:**

This is a Bop-It Game!

Please press the Circle Button to begin!

Press the triangle button!

You have 2500 milliseconds to respond!

Press the triangle button!

You have 2400 milliseconds to respond!

Press the square button!

You have 2300 milliseconds to respond!

Press the cross button!

You have 2200 milliseconds to respond!

Press the circle button!

You have 2100 milliseconds to respond!

Press the triangle button!

You have 2000 milliseconds to respond!

Press the square button!

You have 1900 milliseconds to respond!

Press the circle button!

You have 1800 milliseconds to respond!

Press the circle button!

You have 1700 milliseconds to respond!

Wrong button! :(

You lose!

You made it through 7 rounds!

**Round with a timeout:**

This is a Bop-It Game!

Please press the Circle Button to begin!

Press the cross button!

You have 2500 milliseconds to respond!

Press the triangle button!

You have 2400 milliseconds to respond!

Press the cross button!

You have 2300 milliseconds to respond!

Press the circle button!

You have 2200 milliseconds to respond!

Press the cross button!

You have 2100 milliseconds to respond!

Press the square button!

You have 2000 milliseconds to respond!

Press the square button!

You have 1900 milliseconds to respond!

Press the square button!

You have 1800 milliseconds to respond!

Press the circle button!

You have 1700 milliseconds to respond!

Press the triangle button!

You have 1600 milliseconds to respond!

Press the cross button!

You have 1500 milliseconds to respond!

Press the cross button!

You have 1400 milliseconds to respond!

Press the square button!

You have 1300 milliseconds to respond!

Press the cross button!

You have 1200 milliseconds to respond!

Press the square button!

You have 1100 milliseconds to respond!

Press the cross button!

You have 1000 milliseconds to respond!

You ran out of time. :(

Thanks for playing!

You made it through 15 rounds!

## 