Functions and Decisions/Selections Assn

Use the examples we’ve done as a starting point, and come up with your own game that uses the Micro:bit and functions! You are welcome to combine the turtle module as well. At a minimum, your game must:

* **use at least one fruitful function (that you have created)**. You are welcome to use the fruitful functions shown in our demos, but you need to create at least one of your own!
* **use at least one non-fruitful function (that you have created)**.

Some ideas of things you might want to try creating include:

* “guess the time” game (in which a number of seconds is displayed on the Micro:bit, and the user needs to try to press a button as close to the correct amount of time as possible)
* bop-it game (display an A or B, and user needs to press the correct button. User gets more points the faster they can press the correct button. The game ends if the user presses the wrong button, or after a certain number of clicks.)
* have an “object” (led) fall from the top of the screen. The user needs to catch the object, and can move left/right by using the buttons or accelerometer. If they do not catch the object, the game ends.
* create a morse code visualizer, in which the user can type in a message, and the Micro:bit will display the message by flashing the LEDs using morse code. Another option would be to have the Micro:bit flash AND play beeps on your headphones ([see how to connect your headphones to the Micro:bit](https://makecode.microbit.org/projects/hack-your-headphones/make))
* dice rolling simulator, in which a random number is displayed on the Micro:bit, and used in the program you have running in Thonny
* anything else you can dream up!

**Note:** Please be sure to include a docstring in your function (this allows others to easily understand what your function is intended to do). Ask me in class (or Google it) if you aren't sure what a docstring is.

**Extra for Experts**

Use the [Python Docs turtle reference](http://docs.python.org/dev/library/turtle.html) to do the following:

* include time delays in your program as required, using time.time()
* use the micro:bit for both input and output
* explore Python lists, and use one in your program