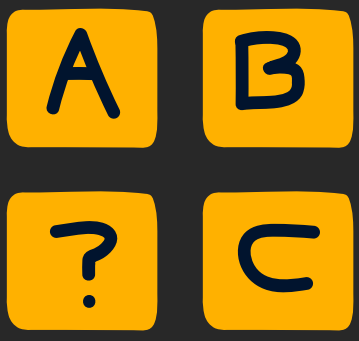
micro:bit Python Games



Created by Mr. Lovell. Find @mrlovellcomp

Micro:bit Tamagotchi Guessing Game

In this activity, you will create the Tamagotchi guessing game on the micro:bit. The player will try to outsmart the Tamagotchi by picking the right button!

Import the random library & set the Tamagotchi and Player scores to zero

Use selection inside a conditional while loop to find a winner.

Use selection to display who wins the game

Tamagotchi Guessing Game

```
# Imports go at the top
from microbit import *
import random
tamagotchiScore = 0
playerScore = 0

# while True loop enables the game to loop back through the Tamagotchi logic
while True:
    #Display Guess - your tamagotchi wants to play a game with you!
    display.scroll('Guess')
    #the tamagotchi pet uses the random library to guess
    guess = random.randint(1,2)
    if button_a.was_pressed():
        display.scroll('A')
        #simple check to see who wins the round!
        if guess == 1:
            playerScore = playerScore + 1
            display.scroll(':(')
        else:
            tamagotchiScore = tamagotchiScore + 1
            display.scroll(':)')
    if button_b.was_pressed():
        display.scroll('B')
        #simple check to see who wins the round
        if guess == 1:
            playerScore = playerScore + 1
            display.scroll(':(')
        else:
            tamagotchiScore = tamagotchiScore + 1
            display.scroll(':)')
    #Display a message explaining who has won
    if playerScore == 3:
        display.scroll('You Win!')
    if tamagotchiScore == 3:
        display.scroll('I win!')

```

Explorer Task Ideas

- Add a start up logo to make the player know they are playing with a pet
- Add sound effects, your Tamagotchi wants to have fun!
- Improve the code and make playing the Tamagotchi game fun!

Created by Mr. Lovell. Find @mrlovellcomp

Micro:bit Simon Says Game

In this activity, you will create a Simon Says game on the micro:bit. The micro:bit will describe a random pattern that the player reproduces using the micro:bit's buttons.

Import the random library, set variables to zero and create empty arrays to store the patterns

Use selection inside a conditional while loop to create random patterns and to play the game

Use selection and iteration to compare the contents of the arrays

```
# Imports go at the top
from microbit import *
import random
microGo = []
playerGo = []
guessCount = 0
guessCorrect = 0
randomNum = 0

# Code in a 'while True:' loop repeats forever
while True:
    #If A is pressed, collect 1 and add to playerGo array
    if button_a.was_pressed():
        display.scroll('1')
        playerGo.append(1)
        #Increase guessCount variable
        guessCount = guessCount + 1
    #If B is pressed, collect 2 and add to playerGo array
    elif button_b.was_pressed():
        display.scroll('2')
        playerGo.append(2)
        #Increase guessCount variable
        guessCount = guessCount + 1
    #If A+B is pressed, collect 3 and add to playerGo array
    elif button_a.is_pressed() and button_b.is_pressed():
        display.scroll('3')
        playerGo.append(3)
        #Increase guessCount variable
        guessCount = guessCount + 1
    #If shaken, have micro:bit generate a random pattern
    #and show on the LED display
    if accelerometer.was_gesture('shake'):
        for i in range(3):
            randomNum = random.randint(1,3)
            microGo.append(randomNum)
        for i in range(3):
            display.scroll(microGo[i])
    #If three guesses have been made, run the array check
    #and display well done, or try again, based on the result.
    if guessCount == 3:
        for i in range(3):
            if microGo[i] == playerGo[i]:
                guessCorrect = guessCorrect + 1
        if guessCorrect == 3:
            display.scroll("Well Done!")
        else:
            display.scroll("Try again!")
        #Now reset the variables and clear the arrays for another go
        guessCount = 0
        guessCorrect = 0
        microGo.clear()
        playerGo.clear()
```

Explorer Task Ideas

- Add a start up logo to make the player know they are playing a guessing game
- Add sounds and gestures that the player has to repeat
- Improve the code and make playing the game fun

Created by Mr. Lovell. Find @mrlovellcomp

Number Guessing Game

In this activity, you will create a number guessing game on the micro:bit. The micro:bit will randomly display a number between 1 and 10 and the player has to choose higher or lower using the A and B buttons.

Import the radio library and set player and micro:bit scores to zero.

Show the random number generated by the micro:bit and use selection to see who has one.

Shake the micro:bit to check your score

```
#imports including random
from microbit import *
import random
mbNumber = 0
score = 0


# Game will repeat. Aim to have have several players, playing several rounds
# Player with the highest score wins
while True:
    # Randomly generate a number that the microbit holds
    mbNumber = random.randint(1,10)
    # Randomly generate a number for the player
    playerNumber = random.randint(1,10)
    display.scroll(mbNumber)
    sleep(2000)
    # Simple message asking the player to guess higher or lower
    display.scroll("?")
    # Nested selection to check whether the micro:bit number is higher or lower than the player'
    if button_a.was_pressed():
        if playerNumber > mbNumber:
            display.scroll('W')
            score = score + 1
        else:
            display.scroll('L')
    if button_b.was_pressed():
        if playerNumber < mbNumber:
            display.scroll('W')
            score = score + 1
        else:
            display.scroll('L')
    # Shake the micro:bit to show the player score.
    if accelerometer.was_gesture('shake'):
        display.scroll(score)
```

Explorer Task Ideas

- Add a start up logo to make the player know they are playing a number guessing game game
- Show animations when a player reaches a score threshold
- Play a sound when player reaches a score

Created by Mr. Lovell. Find @mrlovellcomp

Micro:bit Reaction Game

In this activity, you will create a reaction game on the micro:bit. This is a two-player game, where each player is assigned to the A or B button. The players listen for a sound and first to press their button wins!

Import the random and music libraries

Use selection and randomness inside a conditional loop to make the players react to the sound

Use 'is pressed' to check who pressed their button first!

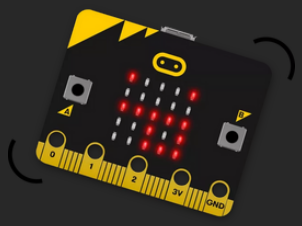
```
# Imports go at the top
from microbit import *
import music
import random


# Code in a 'while True:' loop repeats forever
while True:
    sleep(1000)
    #Show a target on the display
    display.show(Image.TARGET)
    sleep(400)
    #Shake to start
    if accelerometer.was_gesture('shake'):
        #show a '3,2,1' countdown
        display.show(3)
        sleep(1000)
        display.show(2)
        sleep(1000)
        display.show(1)
        sleep(1000)
        #Delay the buzzer for a random moment of time
        wait = random.randint(1, 3)
        if wait == 1:
            sleep(1000)
        elif wait == 2:
            sleep(2000)
        else:
            sleep(500)
        #Increase sound volume to the loudest
        set_volume(255)
        #Play a starter note
        music.play(['c'])
        #Use 'is pressed' to detect who presses the button first
        if button_a.is_pressed():
            display.scroll('A!')
        if button_b.is_pressed():
            display.scroll('B!')
```

Explorer Task Ideas

- Add a start up logo to make the player know they are playing a reaction game
- Add something other than sound that the players have to react to
- Redraft the code to make the game a best-of-3

Created by Mr. Lovell. Find @mrlovellcomp



Micro:bit Just Dance

In this activity, you will create a JustDance game on the micro:bit! This is a multi-player dancing game that a whole class can join in with. The player that most closely matches the coach wins!

Import the radio library and set a score variable to zero.

Use selection to broadcast a different string depending on the gesture.

Shake the micro:bit at the end of the game to see who has won!

```
from microbit import *
import radio
radio.on()
#Set radio power to 1, we are all dancing in the same room!
#Set radio channel to be the same on all micro:bits
radio.config(group=23, power=2)
#Set score to zero
score = 0

# Code in a 'while True:' loop repeats forever
while True:
    # add a small pause before checking for a message
    sleep(500)
    # store the string received by the radio into a variable named 'message'
    message = radio.receive()
    # If the micro:bit is shaken, display the score
    if accelerometer.was_gesture('shake'):
        display.show(score)
        sleep(2000)
        display.clear()
    # now broadcast on radio depending on micro:bit gesture
    # nested if will only increase score by 1 if both the broadcasting micro:bit
    # held by the JustDance coach is being held with the same gesture as the player
    if accelerometer.was_gesture('left'):
        radio.send('left')
        if message == 'left':
            score = score + 1
    if accelerometer.was_gesture('right'):
        radio.send('right')
        if message == 'right':
            score = score + 1
    if accelerometer.was_gesture('face up'):
        radio.send('face up')
        if message == 'face up':
            score = score + 1
    if accelerometer.was_gesture('face down'):
        radio.send('face down')
        if message == 'face down':
            score = score + 1
```

Explorer Task Ideas

- Add a start up logo to make the player know they are playing a dancing game
- Show animations or stars when a player reaches a score threshold, just like JustDance!

Created by Mr. Lovell. Find @mrlovellcomp