



Name: _____

Assessment Frog Frenzy

- 1) How does the controller communicate which buttons are pressed to the Pico?
 - a) It sends an analog signal directly to the Pico
 - b) It uses a shift register to send bits representing the button states
 - c) It stores the button states in memory and sends them all at once
 - d) It uses wireless communication to send the button states

- 2) What are the three pins used to connect the controller to the Pico?
 - a) Power, ground, signal
 - b) Input, output, control
 - c) Read, write, execute
 - d) Data, latch, clock

- 3) What is the purpose of the CLOCK pulse in the controller's communication process?
 - a) It resets the shift register
 - b) It powers the shift register
 - c) It initializes the controller connection
 - d) It sends the button states to the Pico in order, one bit at a time

- 4) How do you ensure the player's character stays within the console boundaries when moving up?
 - a) By using a delay block
 - b) By changing the controller's buttons
 - c) By using a constrain block
 - d) By printing the score

- 5) Which variable determines the horizontal position of the player character?
 - a) player x
 - b) food x
 - c) enemy x
 - d) player y

- 6) Imagine you are designing a different game using the same controller and Pico. Describe the basic idea of your game and how you would use the controller to interact with the game. Include at least two ways you would use the controller.



Name: _____

7) Think about how you can apply the controller and Pico setup you learned to a real-world situation outside of gaming. Describe a potential application and how it would work.

8) Look at the Python code you have created for your game. Currently, the enemy moves at a constant speed of 0.5 units per game loop. Can you change the code so that the enemy's speed increases by 0.1 units each time the player collects the food? Explain what changes you made to the code and why.



Name: _____

Answer Key Frog Frenzy

- 1) B - It uses a shift register to send bits representing the button states
- 2) D - Data, latch, clock
- 3) D - It sends the button states to the Pico one bit at a time
- 4) C - By using a constrain block
- 5) A - player x
- 6) *Example:* For my game, I would design a space adventure where the player controls a spaceship navigating through an asteroid field. The controller would move the spaceship left, right, up, and down to avoid incoming asteroids. Pressing the 'A' button on the controller would allow the spaceship to shoot lasers to destroy the asteroids. The game's goal would be to survive as long as possible and score points by destroying asteroids.
- 7) *Example:* One potential real-world application could be using the controller and Pico to create a remote-controlled robot for search and rescue missions. The controller would allow an operator to navigate the robot through dangerous or hard-to-reach areas. Buttons on the controller could be programmed to operate different functions of the robot, such as extending an arm to pick up objects, turning on a camera to send live video feed, or activating sensors to detect hazards. This setup would enable rescuers to safely explore and perform tasks in hazardous environments without putting themselves at risk.
- 8) *Example:*

```
import board
from piper_blockly import *
import time
import random

# Initialize variables
player_x = None
enemy_x = None
player_y = None
score = None
enemy_y = None
food_x = None
food_y = None
enemy_speed = 0.5 # Added variable for enemy speed
```



Name: _____

```
# Function to read controller input
def read_controller():
    global player_x, enemy_x, player_y, score, enemy_y, food_x, food_y
    buttons_pressed_value = piper_controller.readButtons()
    if piper_controller.wasPressed(BUTTON_2):
        player_y = min(max(player_y - 1, 0), 9)
    if piper_controller.wasPressed(BUTTON_4):
        player_y = min(max(player_y + 1, 0), 9)
    if piper_controller.wasPressed(BUTTON_1):
        player_x = min(max(player_x - 1, 0), 15)
    if piper_controller.wasPressed(BUTTON_3):
        player_x = min(max(player_x + 1, 0), 15)

piper_controller = piperControllerPins(board.GP4, "GP4", board.GP2, "GP2",
board.GP3, "GP3")

try:
    set_digital_view(True)
except:
    pass

# Function to move the enemy
def move_enemy():
    global player_x, enemy_x, player_y, score, enemy_y, food_x, food_y, enemy_speed
# Added enemy_speed
    if enemy_x < player_x:
        enemy_x = isNumber(enemy_x) + enemy_speed
    elif enemy_x > player_x:
        enemy_x = isNumber(enemy_x) + -enemy_speed
    if enemy_y < player_y:
        enemy_y = isNumber(enemy_y) + enemy_speed
    elif enemy_y > player_y:
        enemy_y = isNumber(enemy_y) + -enemy_speed

# Function to print the game state
def print_game():
    global player_x, enemy_x, player_y, score, enemy_y, food_x, food_y
    consoleClear()
    consolePosition(player_x, player_y)
    print('O', end="")
    consolePosition(food_x, food_y)
    print('*', end="")
    consolePosition((int(enemy_x)), (int(enemy_y)))
    print('X', end="")
    consolePosition(18, 5)
    print('Score: ', end="")
    print(score, end="")

# Function to set up the game
def set_up_game():
    global player_x, enemy_x, player_y, score, enemy_y, food_x, food_y, enemy_speed
```



Name: _____

```
# Added enemy_speed
player_x = 0
player_y = 0
enemy_x = 15
enemy_y = 9
food_x = random.randint(0, 15)
food_y = random.randint(0, 9)

# Function to check if player has reached the food
def check_food():
    global player_x, enemy_x, player_y, score, enemy_y, food_x, food_y, enemy_speed
    # Added enemy_speed
    if player_x == food_x and player_y == food_y:
        score = isNumber(score) + 1
        enemy_speed += 0.1 # Increase enemy speed
        set_up_game()

# Function to check if enemy has caught the player
def check_enemy():
    global player_x, enemy_x, player_y, score, enemy_y, food_x, food_y
    if player_x == (round(enemy_x)) and player_y == (round(enemy_y)):
        score = isNumber(score) + -1
        set_up_game()

# Game loop
set_up_game()
score = 0
while True:
    read_controller()
    move_enemy()
    check_enemy()
    check_food()
    print_game()

    time.sleep(0.5)
```

Explanation:

1. I added a new variable `enemy_speed` initialized to `0.5`.
2. I modified the `move_enemy` function to use `enemy_speed` instead of a constant `0.5`.
3. I updated the `check_food` function to increase `enemy_speed` by `0.1` each time the player collects the food.