Laboratorium z przedmiotu Systemy wbudowane (SW)

Karta projektu

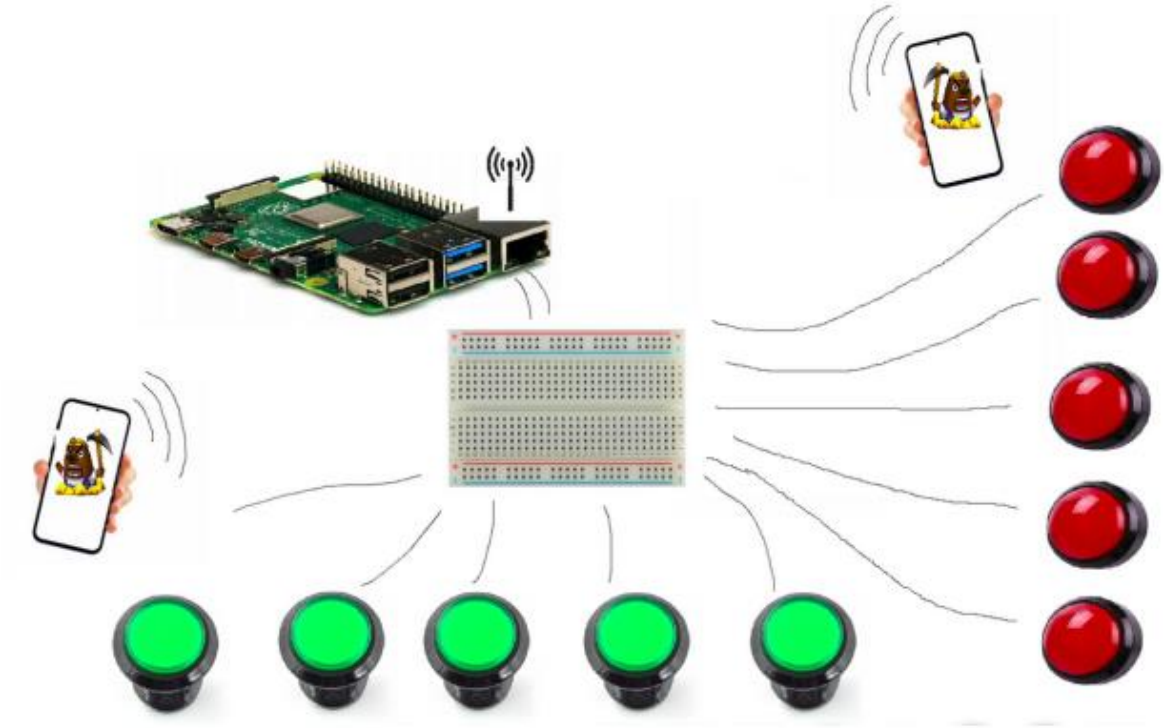
Whack-a-mole++ Multiplayer - Definitive Edition

Prowadzący:	*Autorzy:*	*Grupa dziekańska:*	***I2.2***
mgr inż. Ariel Antonowicz	**144226 145383** **145283 143137**	***Ocena:***	

Cel projektu:

Gra w stylu whack a mole, polegająca na jak najszybszym wciskaniu podświetlających się przycisków. Gra będzie urozmaicona o tryb wieloosobowy (dwóch graczy) oraz o synchronizację z aplikacją mobilną, która zapisuje wyniki, przechowuje rankingi i śledzi przebieg rozgrywki.

Schemat:



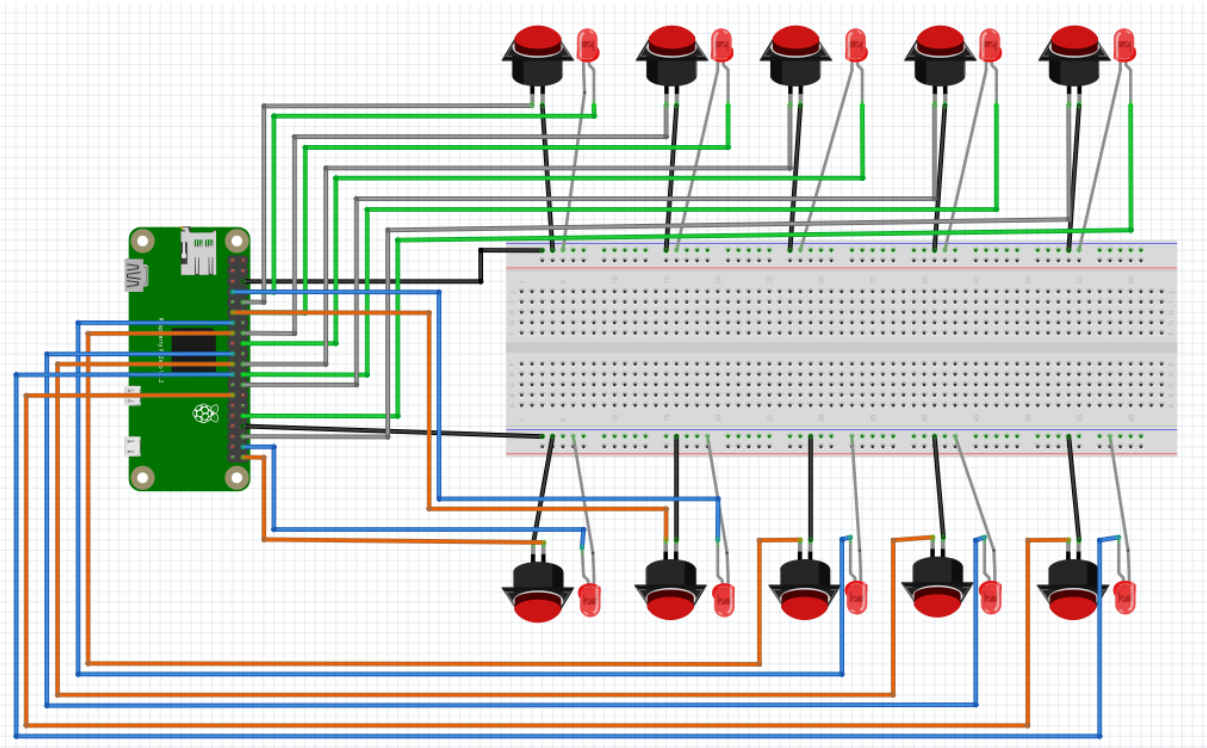
Wykorzystana platforma sprzętowa, czujniki pomiarowe, elementy wykonawcze: Raspberry Pi, Arcade Button, Telefon

Cele:

1. Stworzenie gry polegającej na reakcji użytkownika, aby wcisnął odpowieni guzik w oknie czasowym, gdzie warunkiem porażki jest nie wciśnięcie guzika określoną ilość razy.
2. Rozwinięcie gry poprzez dodanie trybu multiplayer do obsługi dwóch użytkowników, rywalizujących ze sobą. Warunkiem wygranej jest jak najszybsze otrzymanie wymaganej liczby punktów przed oponentem.
3. Stworzenie serwera i aplikacji internetowej obsługującego rozgrywki użytkowników chętnych do gry.
4. Dodanie bazy danych obsługującej dane graczy tj. nazwy i historię rozgrywek.

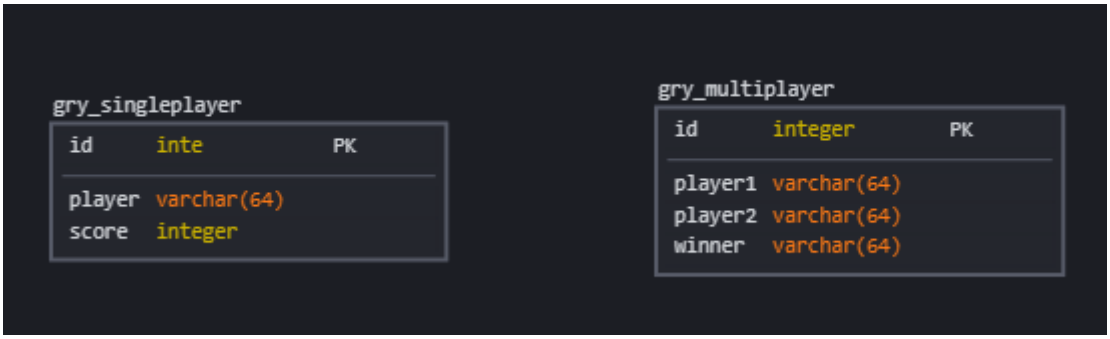
Najważniejszym zadaniem projektu było stworzenie logiki i zaprogramowanie guzików aby odpowiednio reagowały na sygnały wysyłane przez Rasberry jak interakcje z użytkownikiem.

Schemat:



Schemat przedstawia sposób podłączenia warstwy sprzętowej do urządzenia kontrolującego. Jeden guziki zabiera dwa piny GPIO, jeden obsługujący wyświetlanie diody LED drugi zarządzający sygnałem po wciśnięciu guzika.

Schemat bazy danych:



PROJEKT A REALIZACJA:

Wstępne założenia projektu udało się wykonać w całości. Gra polega na wciśnięciu guzika w okienku czasowym aby zdobyć punkt. Obsługuje tryb dla pojedynczego jak i dwóch graczy. Interfejsem jest aplikacja internetowa dostępna z poziomu komputera lub telefonu, której hostem jest nasze Rasberry PI. Pamięcią rozgrywek jest baza danych zapisująca historię rozgrywek graczy lub gracza. Nie udało się wykonać pokazu przebiegu rozgrywki w czasie rzeczywistym na ekranie aplikacji (Wynik rozgrywki jest wyświetlany dopiero po zakończeniu partii), co można dodać jako plany przyszłościowe. Zawsze można powiększyć pulę graczy do grania, zmniejszając liczbę guzików do kliknięcia, lub dodając nowe (jedynym ograniczeniem tego pomysłu jest liczba wejść i wyjść na urządzeniu sterującym).

OPIS KODU:

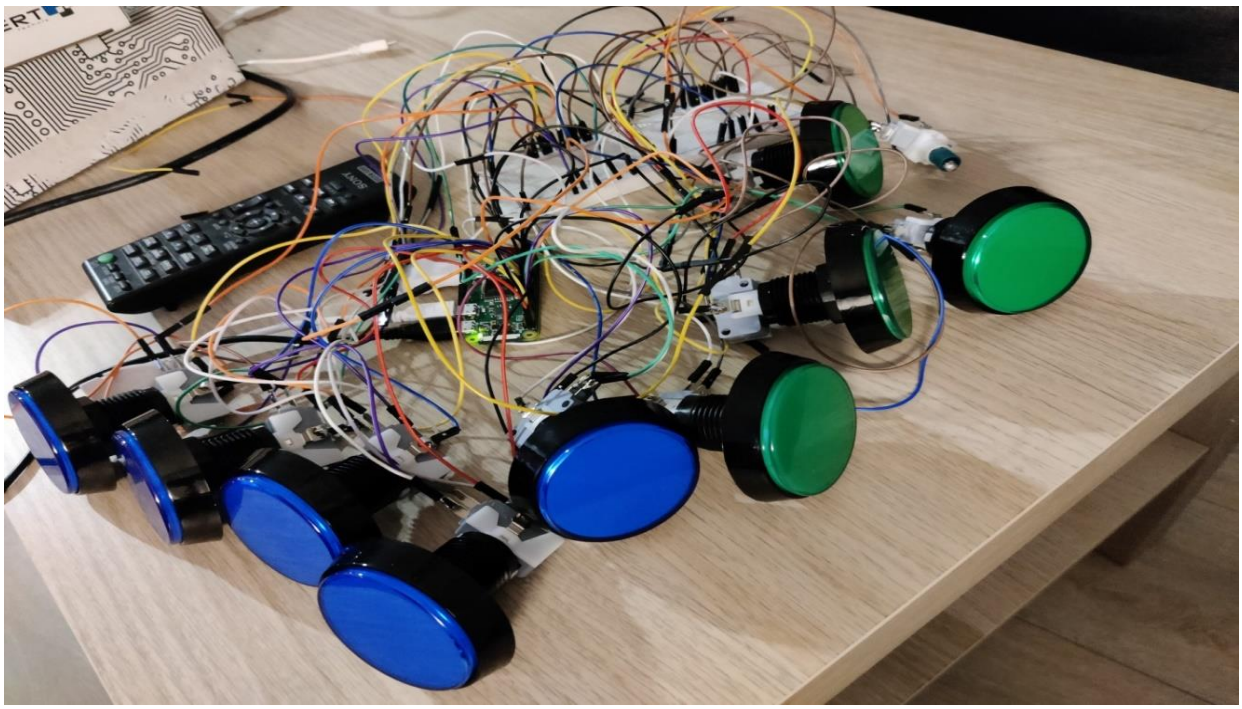
Najważniejszy fragment kodu. Zdefiniowanie klasy Button, która odpowiada za przygotowanie guzika do działania, oraz wywołanie, aby wykonać test refleksu. Dioda LED zaświeca się, kiedy zostanie wylosowana liczba odpowiadająca diodzie i jej guzikowi, w przypadku naciśnięcia guzika dioda LED zostaje wyłączona i zostaje naliczony punkt oraz następuje wyjście z pętli, co praktycznie powoduje zakończenie działania na danym guziku. Skuchy naliczane są w taki sposób, aby w przypadku poprawnego naciśnięcia się zerowały, a w razie przegapienia okienka czasowego na pewno się odejmowały.

Dla modelu multiplayer wykorzystano funkcję pozwalającą na wywołanie dwóch przycisków na raz. Punkt otrzymuje gracz z szybszą rekacją wciskając swój guzik. Kolejność indeksów przycisków jest identyczny, aby dać graczom wyrównane szansę.

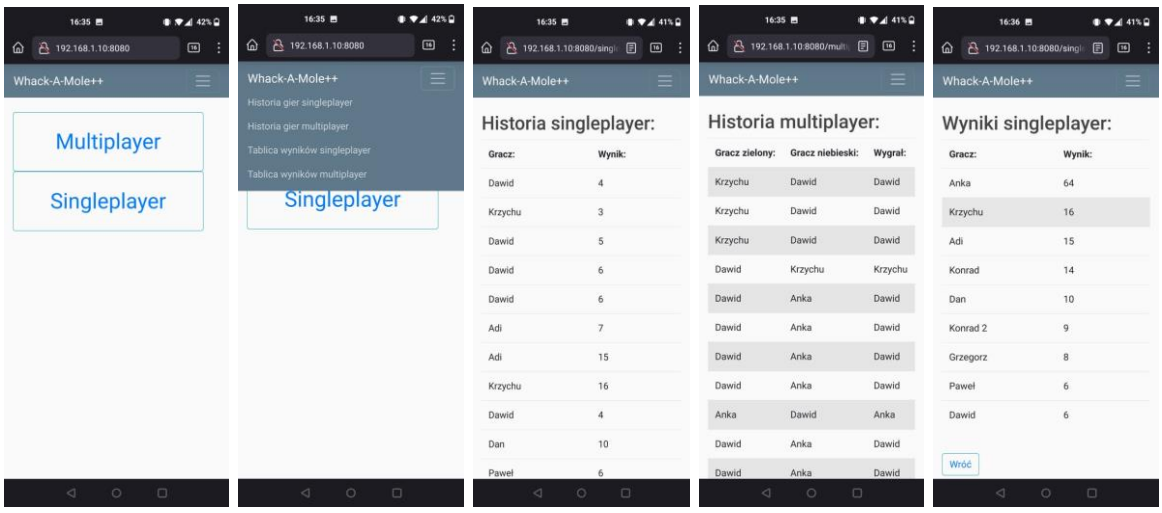
```python
def button_click_multi(buttonA, buttonB, timer):
    global punktyA
    global punktyB
    global skuchyA
    global skuchyB
    timer = int(timer * 1000)
    for i in range(timer):
        sleep(0.001)
        if GPIO.input(buttonA.BUTTON_PIN):
            GPIO.output(buttonA.LED_PIN, 0)
            GPIO.output(buttonB.LED_PIN, 0)
            sleep(1.5)
            punktyA += 1
            break
        elif GPIO.input(buttonB.BUTTON_PIN):
            GPIO.output(buttonA.LED_PIN, 0)
            GPIO.output(buttonB.LED_PIN, 0)
            sleep(1.5)
            punktyB += 1
            break
        else:
            GPIO.output(buttonA.LED_PIN, 1)
            GPIO.output(buttonB.LED_PIN, 1)
```

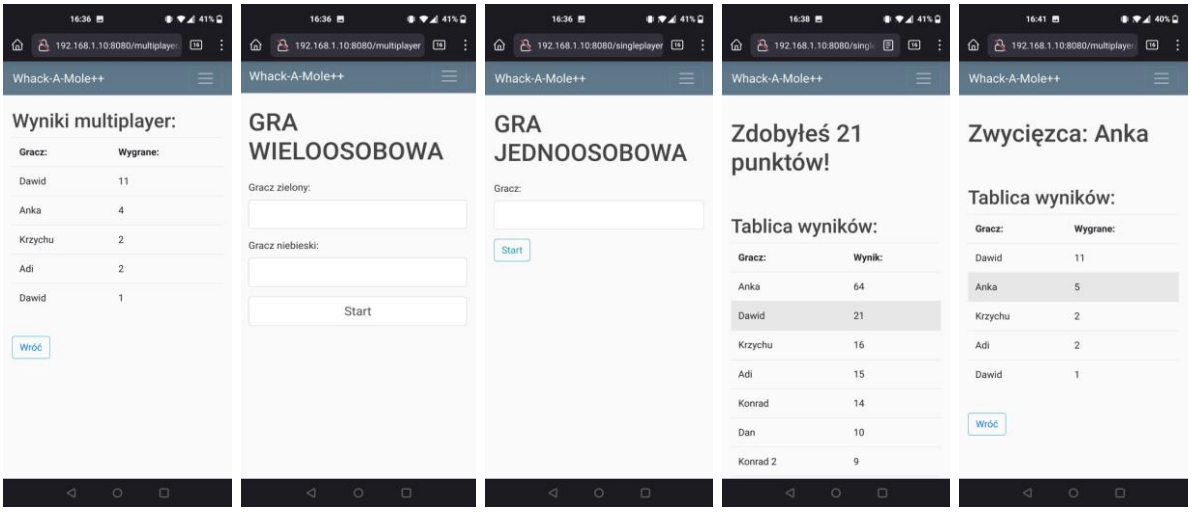
```python
class Button:
    def __init__(self, BUTTON_PIN, LED_PIN):
        self.BUTTON_PIN = BUTTON_PIN
        self.LED_PIN = LED_PIN

    def set_up(self):
        GPIO.setup(self.BUTTON_PIN, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
        GPIO.setup(self.LED_PIN, GPIO.OUT)

    def button_click(self, timer):
        timer = int(timer * 1000)
        global punkty
        global skuchy
        for i in range(timer):
            sleep(0.001)
            if GPIO.input(self.BUTTON_PIN):
                GPIO.output(self.LED_PIN, 0)
                punkty += 1
                skuchy += 1
                break
            else:
                GPIO.output(self.LED_PIN, 1)
        skuchy -= 1
        print('Punkty', punkty)
        print('Skuchy', skuchy)
        return
```

Rzeczywiste zdjęcia projektu:



Aplikacja mobilna:





Podsumowania i wnioski:

Całe założenie projektu obejmuje wykorzystanie każdej nauki wynoszonej na obecnym semestrze studiów. Posiadamy urządzenie sterujące, które jest zarządzane przez stronę internetową, której aplikacja również posiada bazę danych. Projekt uznajemy jako angażujący i wymagający odpowiedniego przygotowania teoretycznego, aby podjąć kroki przed praktycznym wykonaniem.

Film z pełnym działanie projektu:

https://www.youtube.com/watch?v=nS4xYHBNqu8