|  |  |
| --- | --- |
|  | **AGH – UNIVERSITY OF SCIENCE AND TECHNOLOGY** |

Project documentation for

**“The Snake Game”**

**Design Laboratory**

Electronics and Telecommunication EN, III year

*Patrycja Nazim*

coordinator: Agnieszka Dąbrowska-Boruch

28.12.2024

# Project description

The project was created with the purpose of broadening the knowledge about microcontrollers and testing how they work. With the usage of ESP8266 microcontroller the Snake Game was created, its main purpose is to provide fun, and the satisfaction of completing the game. Player can win the game when he/she scores at least 100 points, which are the apples eaten by the snake.

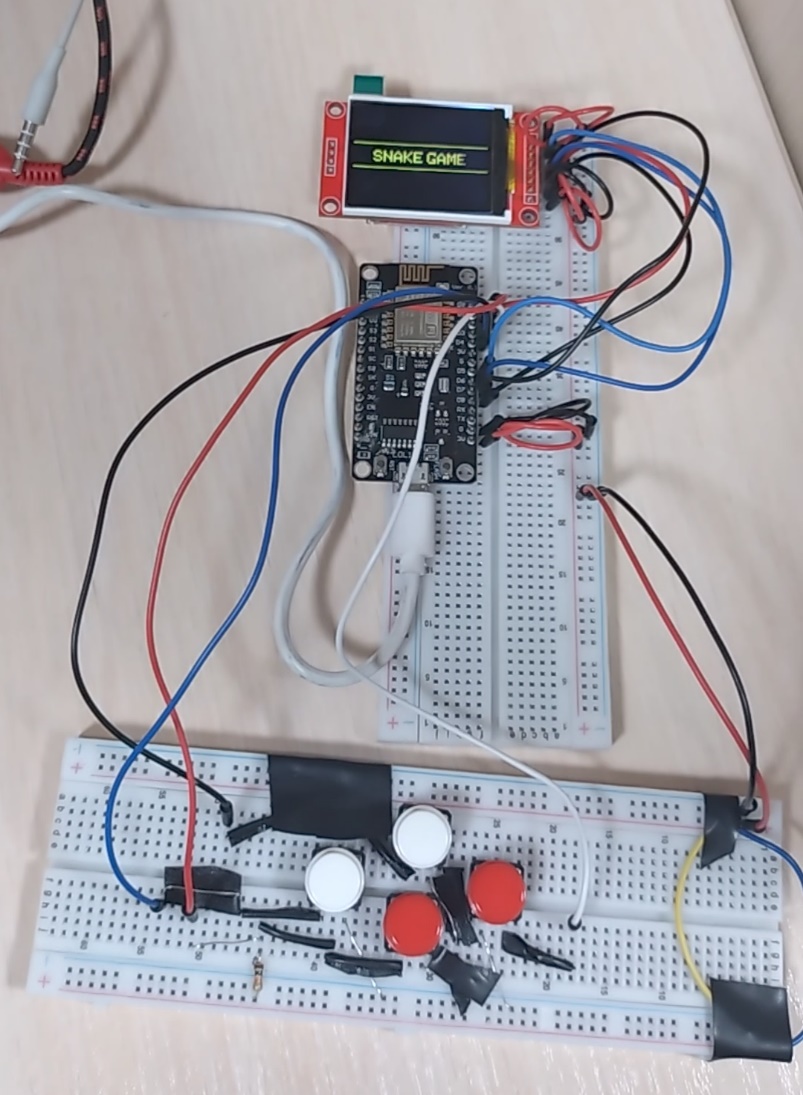


Figure : The project



Figure : The win screen

Video of the win screen:

[20241228\_202921~3.mp4](https://aghedupl-my.sharepoint.com/:v:/g/personal/nazim_student_agh_edu_pl/EcL9DlghiKZHv_Hlo0fvO8kBe92ahzFoDU5MOiSHa8dhtg?e=Mcv64O&nav=eyJyZWZlcnJhbEluZm8iOnsicmVmZXJyYWxBcHAiOiJTdHJlYW1XZWJBcHAiLCJyZWZlcnJhbFZpZXciOiJTaGFyZURpYWxvZy1MaW5rIiwicmVmZXJyYWxBcHBQbGF0Zm9ybSI6IldlYiIsInJlZmVycmFsTW9kZSI6InZpZXcifX0%3D)



Figure : the loss screen

Video of the loss screen:

[20241228\_205216~3.mp4](https://aghedupl-my.sharepoint.com/:v:/g/personal/nazim_student_agh_edu_pl/Eeho4dg1qj1AiuzunqC0G3QBQHQrlVaEm0z3fNXXQvZEZg?e=PL1N0m&nav=eyJyZWZlcnJhbEluZm8iOnsicmVmZXJyYWxBcHAiOiJTdHJlYW1XZWJBcHAiLCJyZWZlcnJhbFZpZXciOiJTaGFyZURpYWxvZy1MaW5rIiwicmVmZXJyYWxBcHBQbGF0Zm9ybSI6IldlYiIsInJlZmVycmFsTW9kZSI6InZpZXcifX0%3D)

# User’s manual

To play the game the user can use the buttons connected on the board in similar way to how other consols are built.

A circuit board with red and white buttons

Description automatically generated

Figure : the user interface

Top button pressed – snake turns up

Right button pressed – snake turns right

Left button pressed- snake turns left

Down button pressed – snake turns down

The game runs in a loop so no users action is needed to play again after one game ends.

# Source files

Software of the project was created in the Visual Studio Code, with the Platform IO VSCode extension, and the usage of external libraries.

The project consists of the following source files:

* *main.cpp* – file inclusion directives, declaration and implementation of all the classes, variables, functions, graphical interface, definitions
* platform.ini – software testing method by which individual units of source code, sets of one or more MCU program modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use

# Dependencies

The following external libraries are used in the project:

* Adafruit\_GFX.h 1.11.11 – This is the core graphics library for the display.

<https://github.com/adafruit/Adafruit-GFX-Library>

* Adafruit\_ST7735.h - This is a library for display based on ST77\* drivers.

<https://github.com/adafruit/Adafruit-ST7735-Library>

* Arduino.h - gives the programmer access to a large group of standard C libraries, and also libraries written especially for Arduino.

<https://github.com/arduino/ArduinoCore-avr/blob/master/cores/arduino/Arduino.h>

* SPI.h - allows for the communication with SPI devices, with the Arduino board as the controller device.

<https://docs.arduino.cc/language-reference/en/functions/communication/SPI/>

* Wire.h -  allows for the communication with I2C devices

<https://docs.arduino.cc/language-reference/en/functions/communication/wire/>

* functional – provides a set of predefined class templates for function objects, including operations for arithmetic, comparisons, and logic.

<https://en.cppreference.com/w/cpp/header/functional>

# Structs description

In the project the following structs were created:

1. Point – represents a point with coordinates x,y. Used to store the position of the elements on the game field.
2. Snake – represents a snake in the game. It manages the snake’s position, size and direction.
   * 1. Members:

* Point head\_, tail\_: Tracks the position of the snake's head and tail.
* Direction direction\_: Stores the current direction of movement (e.g., Up, Left).
* uint8\_t size\_: The initial size of the snake.
  + 1. Methods:
* set\_direction: Changes the direction of the snake, ensuring it doesn't reverse direction.
* get\_head and get\_tail: Provides access to the head and tail coordinates.
* get\_direction: Retrieves the current movement direction.

1. FieldElement - represents an individual pixel in the game field.

3. Members:

* uint8\_t pixel\_ : stores the type of pixel (e.g., Empty, Wall, Apple).
* bool has\_food\_ : indicates whether the cell contains food.

4. Methods:

* get\_pixel: returns the pixel type.
* has\_food: returns whether the cell has food.

1. template <size\_t Width, size\_t Height> struct Game - manages the game logic and field.

5. Members:

* Snake &snake\_: reference to the snake instance.
* FieldElement field\_[Width][Height]: an array representing the game field.
* bool is\_over\_: tracks whether the game is over.
* std::function<void(uint8\_t, uint8\_t)> on\_food\_added\_: callback for when food is added to the field.

6. Key Methods:

* start\_game: initializes the game, placing the snake, walls, and apples.
* move\_snake: moves the snake and handles collisions or eating food.
* add\_snake, add\_wall, add\_Apple: adds initial game elements (snake, walls, apples).
* move\_head, move\_tail: handles movement logic for the snake's head and tail.
* Pixel Enumeration**:** defines types of pixels in the game field, like Empty, Wall, Apple, or directional arrows (Up, Left).

# Future development

This project could be further developed and upgraded. The user interface could be improved by using Bluetooth communication between the buttons and the microcontroller, so the user can grab the buttons more freely. Other type of cables could be used to maintain more cleaner look. The screen used could be bigger in size, so that the game is better visible. The ranking system could be implemented where user could save his record.

# GitHub link

The code used in this project can be found the GitHub platform:

[**https://github.com/Coffee-deve/Design\_lab\_project/tree/Snake\_game**](https://github.com/Coffee-deve/Design_lab_project/tree/Snake_game)