0512-4494 Spring 2025
Project #4

Yehonatan Kohavi - kohaviy@tauex.tau.ac.il

In this project, you will gain hands-on experience by simulating the launch and control of a **Missile** as it pursues a target. You will also let the user set the configuration of the simulation via a web interface.

Submission Guidelines

- Presentation date is **08/06/2025**
- Moodle due date (code freeze & submission) is **07/06/2025**
- Submission file is: **project_4_ID1_ID2.zip**

  The zip should include the following files:

  - presentation.ppt
  - diagram.j
  - libraries.txt
  - **your relevant code (sources + headers)**

  E.g. for a pair of students with IDs 123456789 and 987654321 the zip file should be named:

  project_4_123456789_987654321.zip

- Do not forget to read the **General Submission Guidelines document** & the suggested workflow before starting to work on this project.

Project Description

In this project you will make a simulation of a missile launch and control.
Basically you will display on a screen a target, aim a missile and launch it towards the target, complying with the laws of motion.

- The simulation will support ballistic missile, and a powered missile (with user control).
- The simulation will support 3 levels of **route difficulties** - no obstacles, fixed obstacles, random obstacles.
- The simulation will support 3 levels of **target difficulties** - static target, slow moving target, target moving randomly.
- The simulation will support a very basic simulation menu configuration via a **web** interface.

Item List (Hardware Specifications)

The supported hardware list of WOKWI: supported-hardware and interface.

In this project **you will use:**

- 1x ESP32 Microcontroller
- 1x ILI9341 2.8" Touch Screen LCD -
  Note: The link provided refers to the regular LCD version – not the touchscreen. You will use the touchscreen version.
- 1x Push Button - for missile launch.
  Note: you don't have to use external component, the ESP32 has one onboard.
- 1x LED.
  Note: you don't have to use external component, the ESP32 has one onboard.
- 1x Buzzer - for Hit/Miss target sounds.
- Wires (for connections)
- Resistors - in case needed.

This item list is merely a **recommendation**. You may want to change, replace, or add different items. As long as you comply with the Software Specs, it's fine; however, you should mention any changes in the presentation.

Software Specifications

textbfImportant:

- There are numerous **example projects available online** that you can use to your advantage - some feature games and simulations similar to this project, while others demonstrate the use of HTTP servers and MQTT, as required. Take the time to explore and learn from them before starting your own implementation.
- This will require **vscode** with WOKWI extension to run the HTTP server locally.

You can use the wokwi libraries as you wish.
Your project (code) should comply with the following **SW specifications**:

1. **Web Menu Interface**
   You will connect the ESP32 to the WiFi, run a HTTP server **locally** and create a very simple configuration menu with **HTML**.
   In the web server you will prompt a title, and have the following buttons/fill boxes/sliders (at your choice):

   (a) Missile Type - Ballistic/Powered.

   (b) Missile Launch Speed.

   (c) Missile Angle.

   (d) Route Difficulty:

   i. No obstacles.

   ii. Fixed obstacles.

   iii. Random obstacles.

   (e) Target Difficulty:

   i. Static Target.

   ii. Slow Moving Target (on a single vertical line).

   iii. Randomly Moving Target (contained in a small area).

   (f) "Upload Simulation Configuration" – locks the configuration and turns on the LED.

   At your convenience:

   - ESP32 - Guide
   - ESP32 - HTTP server example
   - HTML Guide

2. **Screens**
   Use the **320x240** resolution (rotate the screen on the side), and show the following screens:

   (a) Start Up Screen - Prompt message: "Pending Simulation Configuration from WEB".

   (b) Simulation Over Screen - will be shown for a few seconds - prompting to the user: "Target Hit / Miss" after the simulation is over. Then - returning to Start up screen.

   (c) Simulation Screen - The missile will be in the lower left corner, the target somewhere in vertical axis of the right corner of the screen.
   Note: You shall switch to this screen only after the simulation configuration has been uploaded from the WEB.

   The missile side shall be a reasonable (a few pixels), and the target size shall be bigger than the missile size but not so big (the sizes and shapes are at your choice).

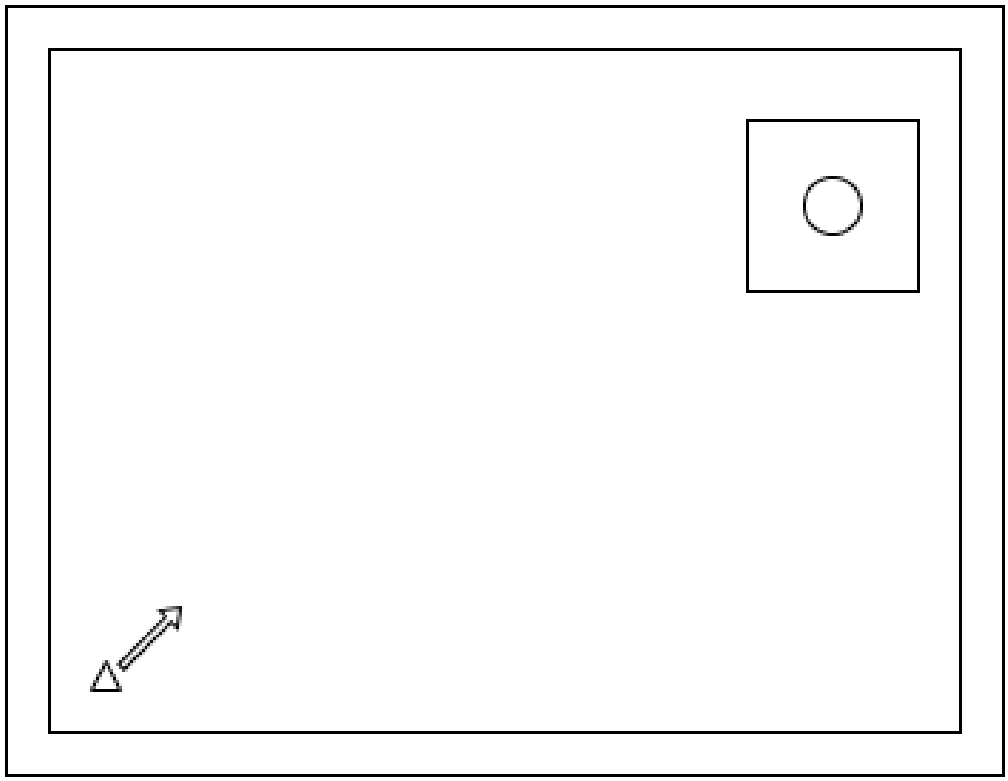

Figure 1: Simulation screen example of no obstacles, random moving target.
The missile is represented as a triangle, the target as a ball, and the target moves randomly within a bounding box—which is currently visible but should be hidden.

3. **Sounds**
   Make unique tones for the following events:

   - Missile Launch - once the launch button is pressed.
   - Missile Hit Obstacle.
   - Missile Hit Target.
   - Missile Miss Target.

   All of the sounds should be unique and different from one another, and the tones are left for you to choose.
   You can add sounds for other events as you wish.

4. **Configuration Values & Controllers**

   - You shall choose **Launch Speed** of the missile in such a way that the whole flight shall take around 20 seconds, and maximal 60 seconds (might be less).
   - You shall choose **Launch Angle** to be between 0-90 degrees.
   - **'Powered' Missile Control** - you can control its vertical movement by tapping the screen - tapping below the missile's current Y position will make it descend, while tapping above will make it ascend. The horizontal speed remains constant.

5. **Difficulty of Route**
   You should implement 3 route difficulty levels (mutually exclusive):

   (a) No Obstacles.

   (b) Fixed Obstacles - hard coded - at your choice.

   (c) Random Obstacles - will change for each simulation.

6. **Difficulty of Target**
   You should implement 3 route difficulty levels (mutually exclusive):

   (a) Static Target.

   (b) Slow Moving Target in a single dimension (up and down).

   (c) Randomly moving target - Should keep a designated box of a few pixels in which the target shall move in a random movement contained in that box making the aim hard to hit. Also the speed shall be not too high and not too low - at your choice.

   NOTE: the **route** difficulties and the **target** difficulties are **not** mutually exclusive, i.e. we can have any combination of them, e.g. fixed obstacles and randomly moving target.

7. **Single Simulation Sequence**

   (a) You will show the **Start-Up Screen**, and have the LED **turned off**.

   (b) The user will choose the simulation mode **via the web** - lock it.

   (c) You will switch to the **Simulation Screen** and turn the LED **on**.

   (d) Now the user will **launch** the missile only after the LED is on, via the **push button**. A sound shall be played.

   (e) The missile shall **fly according to the laws of motion** (and the missile type) until it would hit a target / reach the targets vertical axis - in which its either hit/miss. The LED shall be blinking while the missile is is flying.
   Note:
   - There is gravity to consider.
   - There is no air resistance.

   (f) The simulation screen will **freeze for a few seconds**, and a sound shall be played according to the hit/miss event.

   (g) Finally switching to the **Simulation Over screen** for a few seconds - letting the user know the outcome of the simulation.

   (h) **Repeat**

Bonus: Set up a **MQTT server**, and publish the missile's: **X,Y Coordinates** every 2 seconds from launch until the simulation is over.

- You can use HiveMQ as the MQTT broker.
- If you want (not mandatory for this bonus) integrate with Node-RED.

Three many examples are available online. Here is one.