

Microcontroller mini-game "Defence Line"

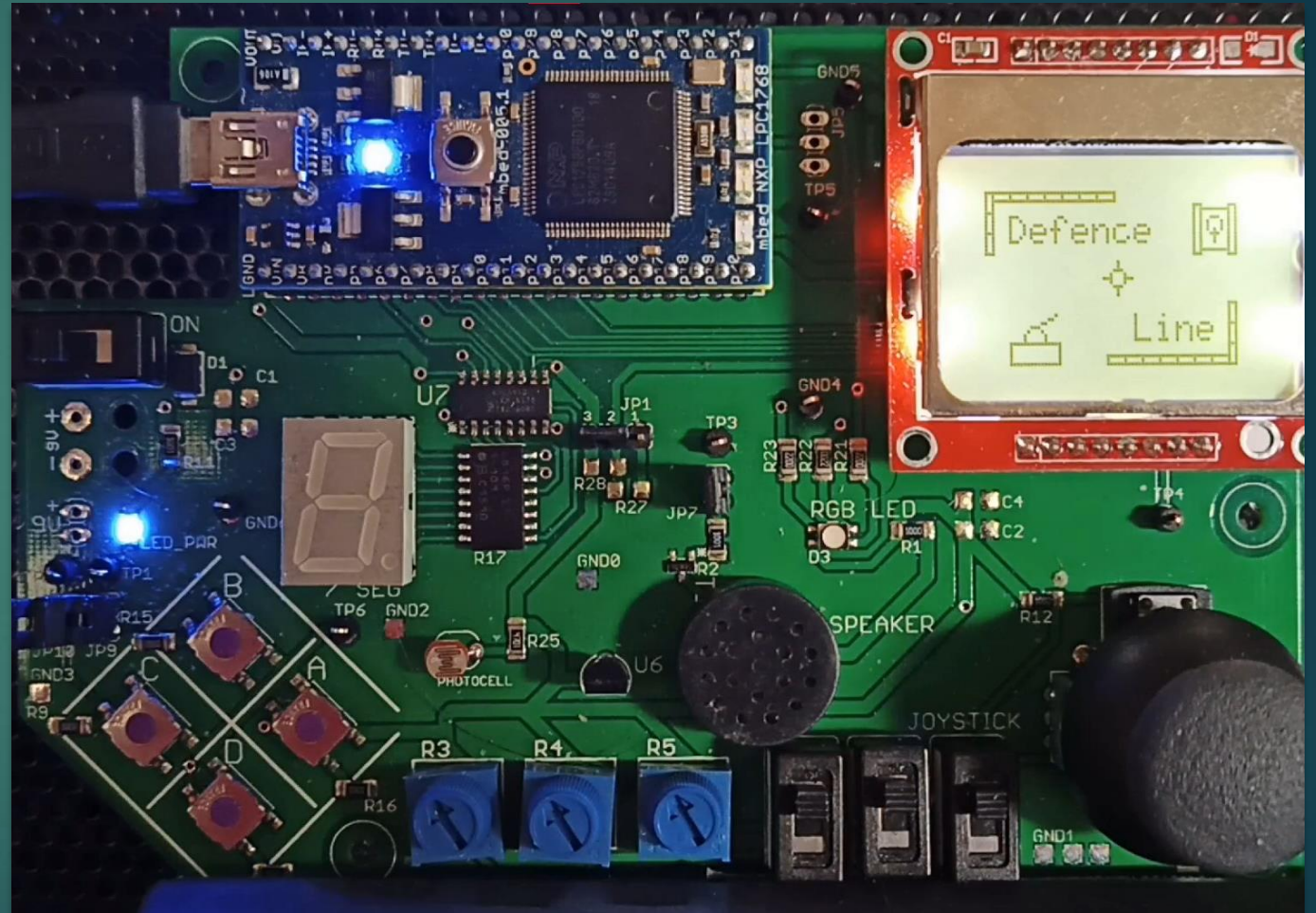
Made by Vitaly Okolelov

Email: okolelovvitaly@gmail.com

GitHub: <https://github.com/okolelovv>

LinkedIn:

<https://www.linkedin.com/in/vitaly-okolelov-715699194/>



Development board

Microcontroller: ARM Mbed LPC1768

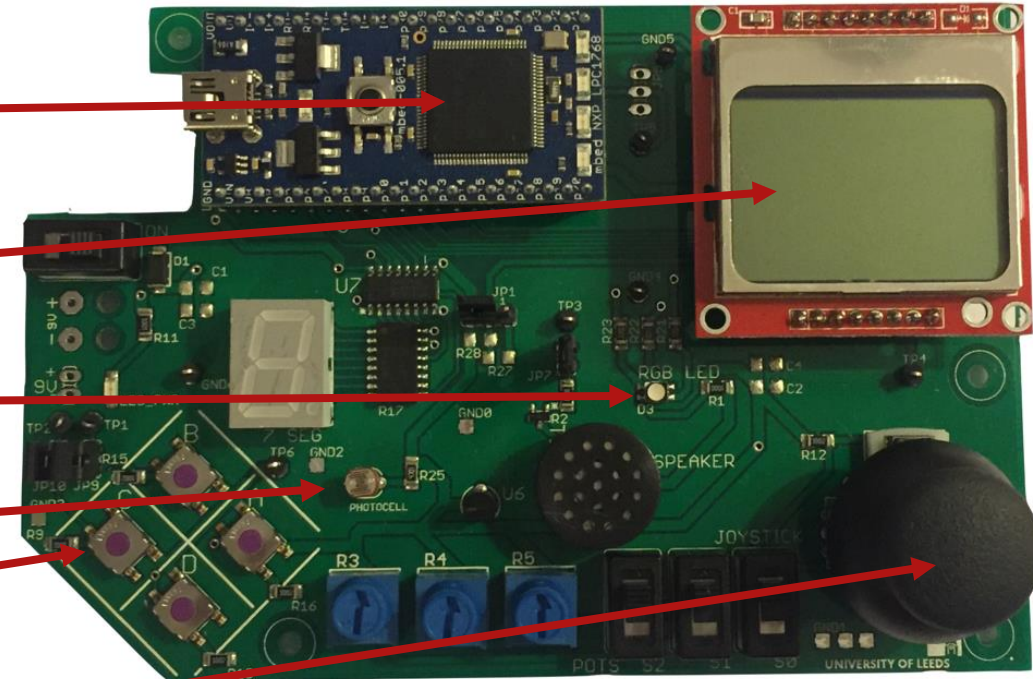
Nokia 5110 LCD display 84×48 pixels

RGB LED

Light-dependent resistor

4 push buttons

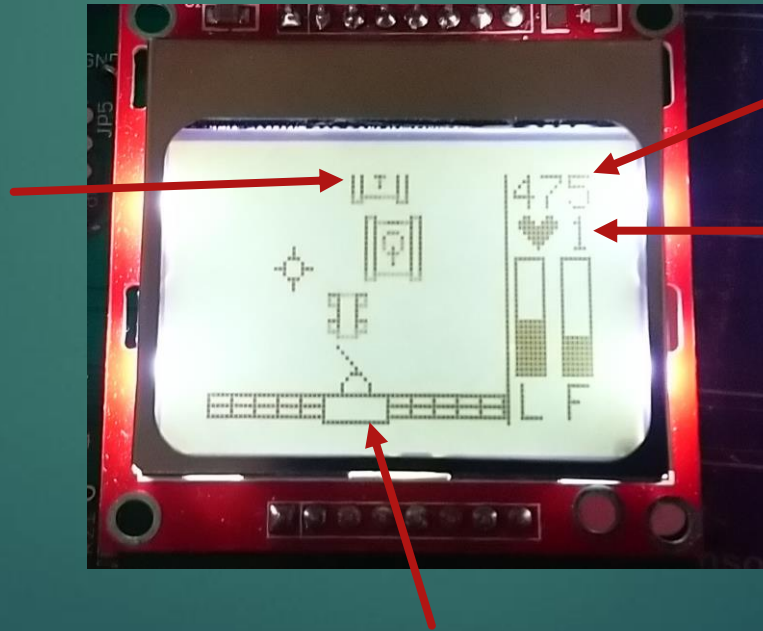
Thumb joystick



Game design: Goal

Survive for a specific time and destroy enemies before they destroy the turret or breach through the defence line

Enemy SUVs and Tanks
coming from the top



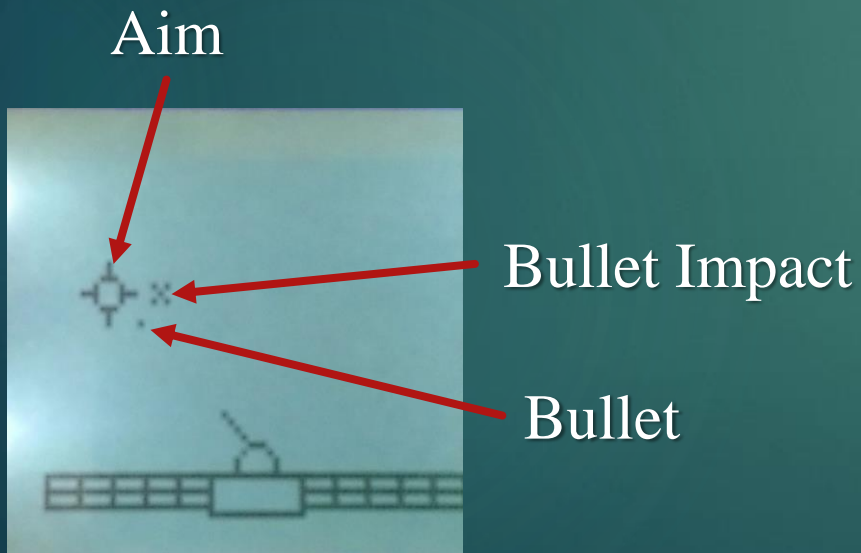
Remaining time

Turret life

Defence line with a turret

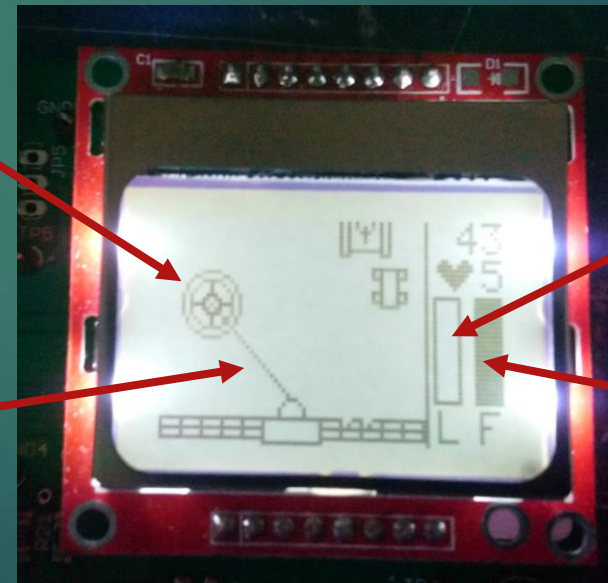
Game design: Weapon

- ▶ Gun: shoots bullets at aim; bullets have constant speed and do 1 damage each
- ▶ Laser: reaches aim instantly and kills any enemy. Requires short cooldown time after use
- ▶ Freeze: stops enemies for 4s. Requires long cooldown time after use



Laser Impact

Laser Beam

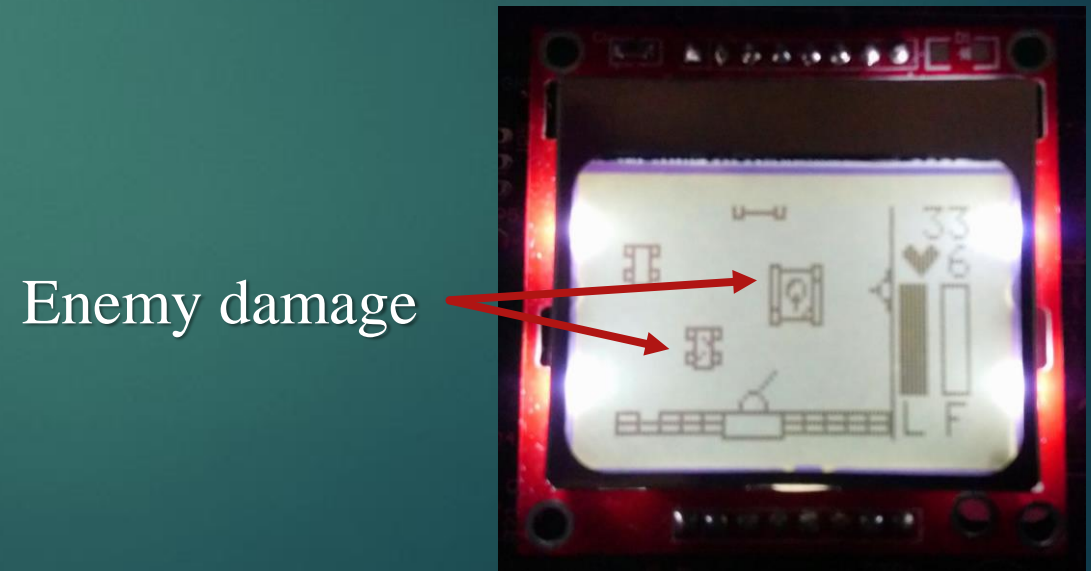
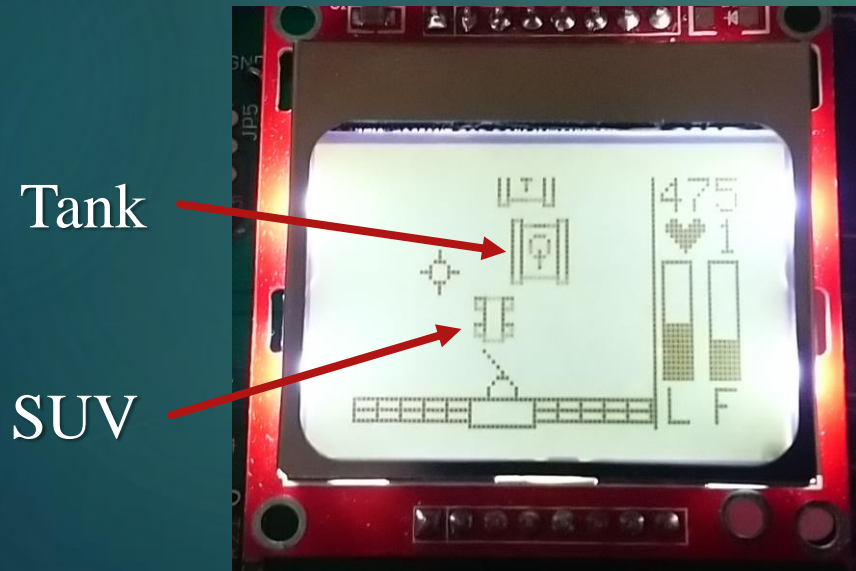


Laser Cooldown

Freeze Cooldown

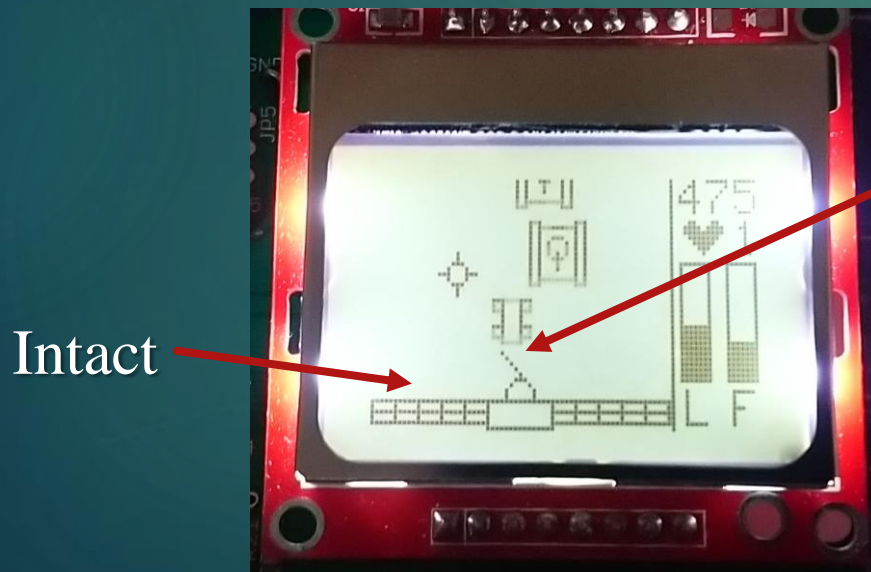
Game design: *Enemies*

- ▶ Using light-dependent sensor for enemy generation random seed (position, type, time)
- ▶ SUV requires 2 bullet hits to be destroyed and Tank requires 4 bullet hits
- ▶ Collision with Turret reduces its health by 1; collision with defence wall damages bricks
- ▶ Visual representation of damage for enemies

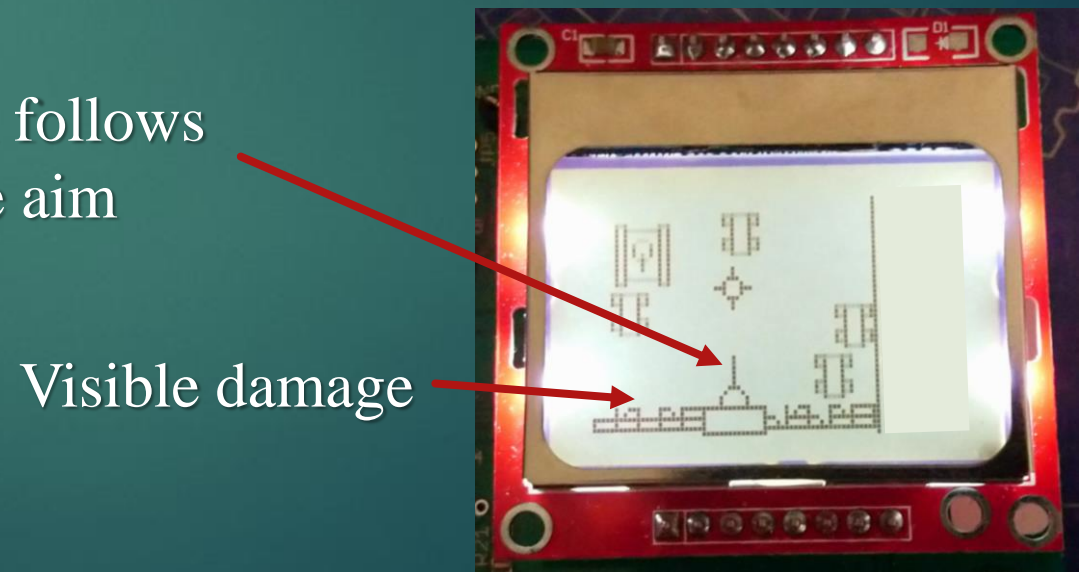


Game design: *Defence Line*

- ▶ Individual brick can withstand 2 enemy collisions and then breaks
- ▶ If two consecutive brick are broken, then enemies can break through the line (game lost)
- ▶ Turret's barrel follows the aim
- ▶ Visual representation of damage for the defence line

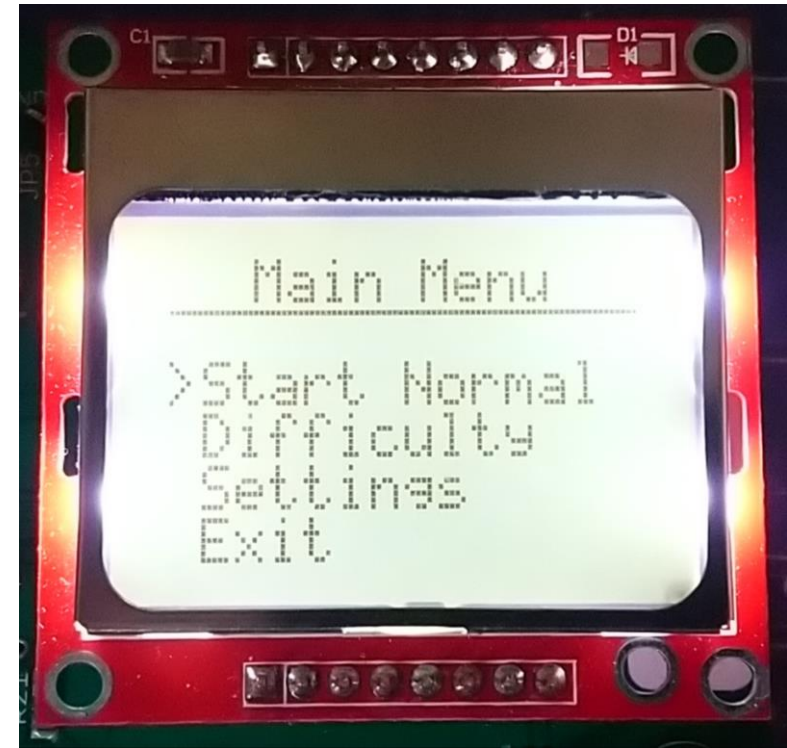


Barrel follows
the aim



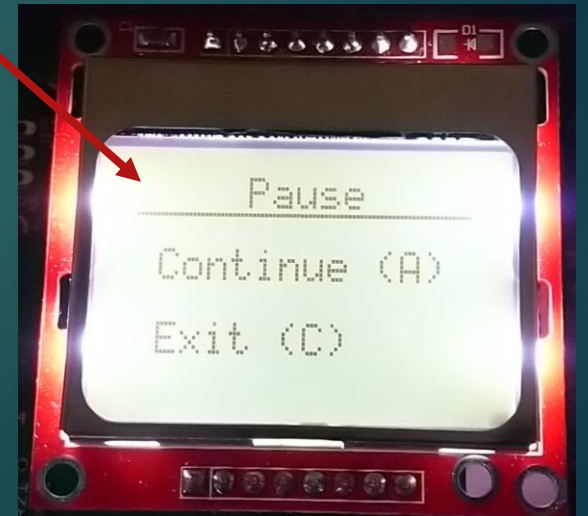
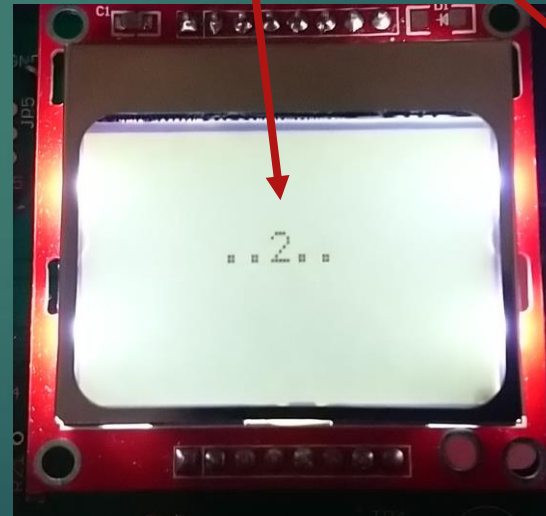
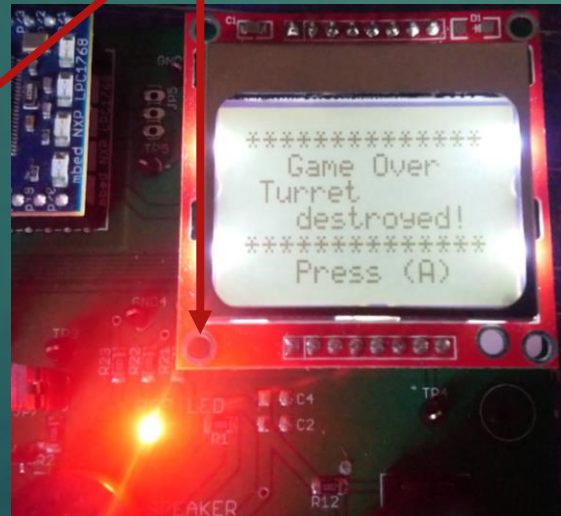
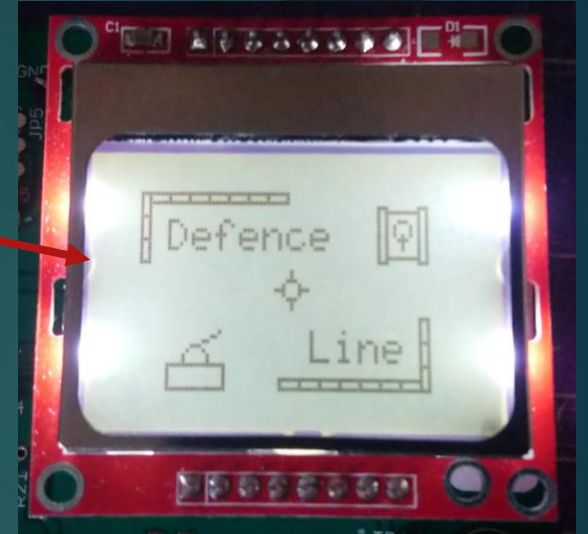
Game design: Menu

- ▶ Difficulty selection: affects the number of enemies and the SUV:Tank spawn proportion
- ▶ Settings: adjusting wave time in minutes, enemy speed in pixels per second, and turret health.



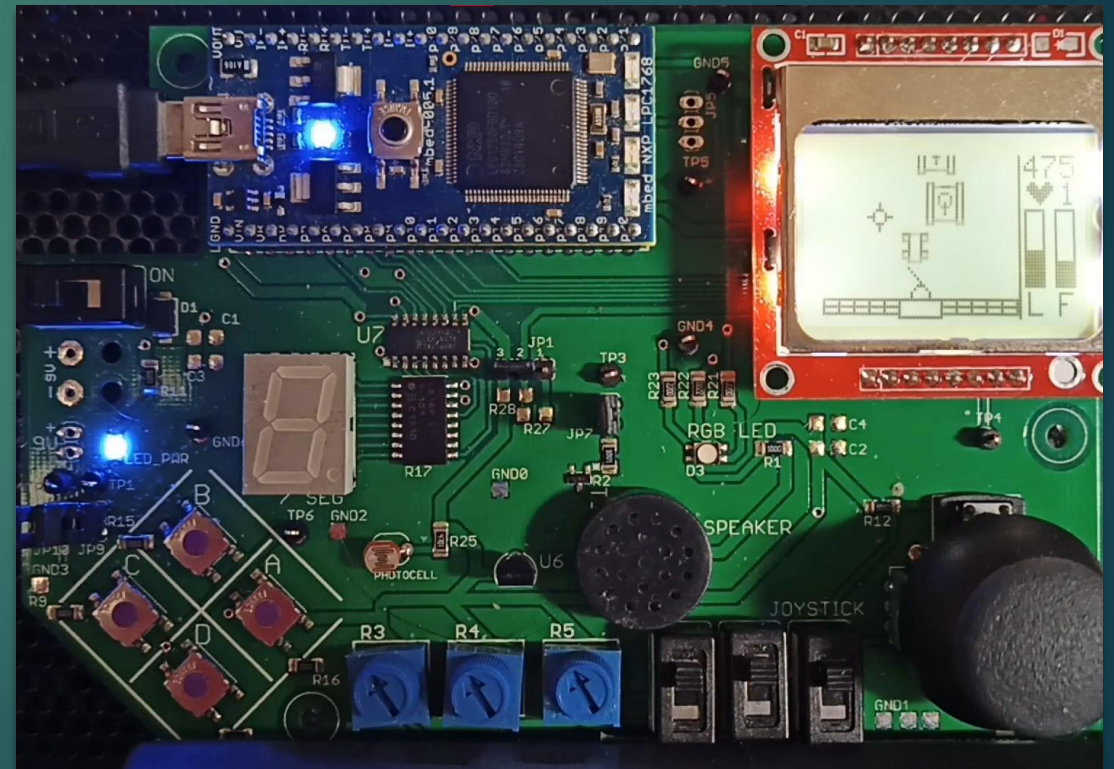
Game design: additional features

- ▶ Start screen and end screens for every outcome
- ▶ Ability to pause the game, then continue or exit
- ▶ Countdown before starting or continuing the game
- ▶ LED indication of events



Game Controls

- ▶ Move thumb joystick to change the position of the aim
- ▶ Press B button to shoot a bullet from Gun
- ▶ Press C button to use Laser if available
- ▶ Press A button to use Freeze if available
- ▶ Press D button to pause the game

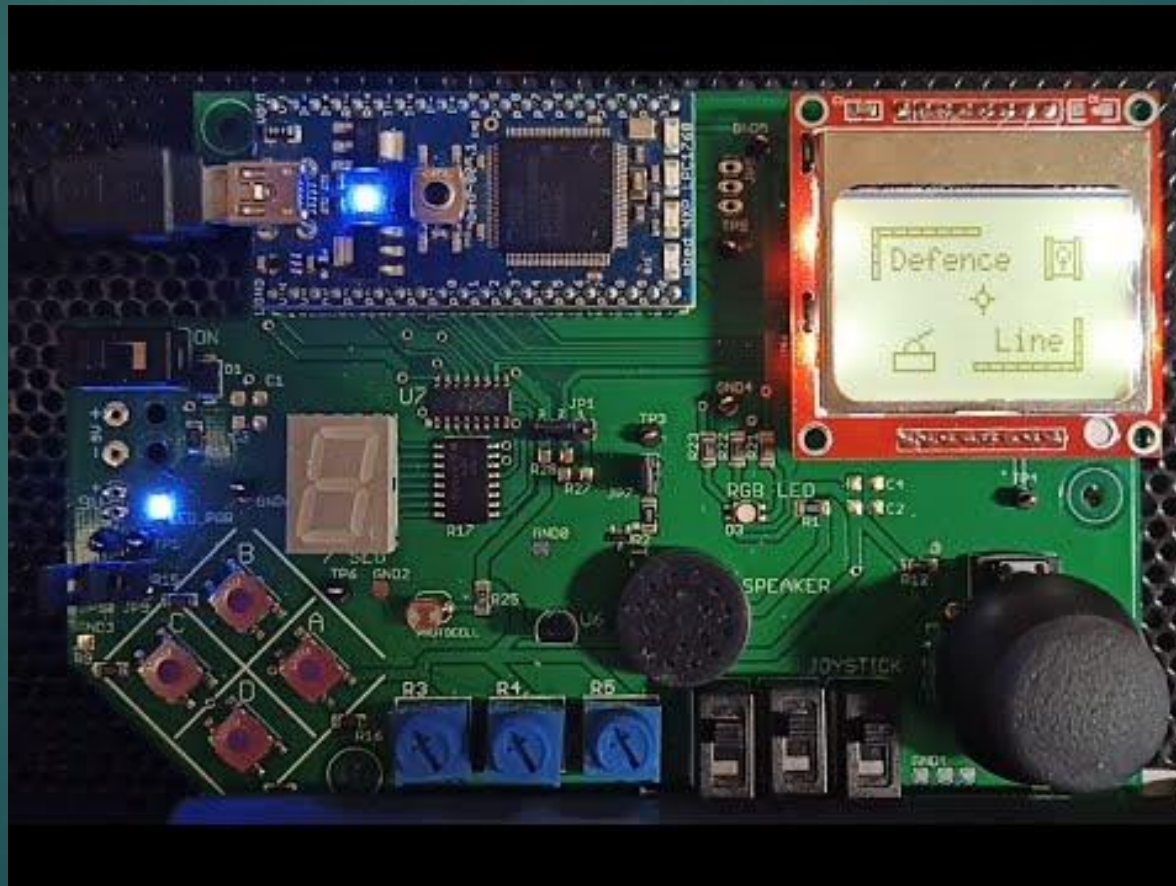


Programming Highlights

1. Interrupts for buttons to provide more responsive control, time interrupts to increase time precision, and sleep function to save power
2. Multiple Finite State Machines to simplify behaviour programming
3. Bullet and enemy generation via dynamic memory allocation
4. Modularity of the project: there are 7 classes, 4 structs, 3 enum classes in 17 files, and the only global variables are the ones related to the peripherals. Thanks to this, the program is easily modifiable and static RAM efficient
5. There are 45 #define directives related to the screen and time parameters, which makes the game easily portable to a device with a bigger screen

Game Demonstration

Access the video via the link: <https://youtu.be/2o6DjbYQ1rU>





Thank you!