

CS/EE 120B Custom Laboratory Project Report

Bloons Tower Defense

Arthur Hittinger (ahitt003)
12/11/2024

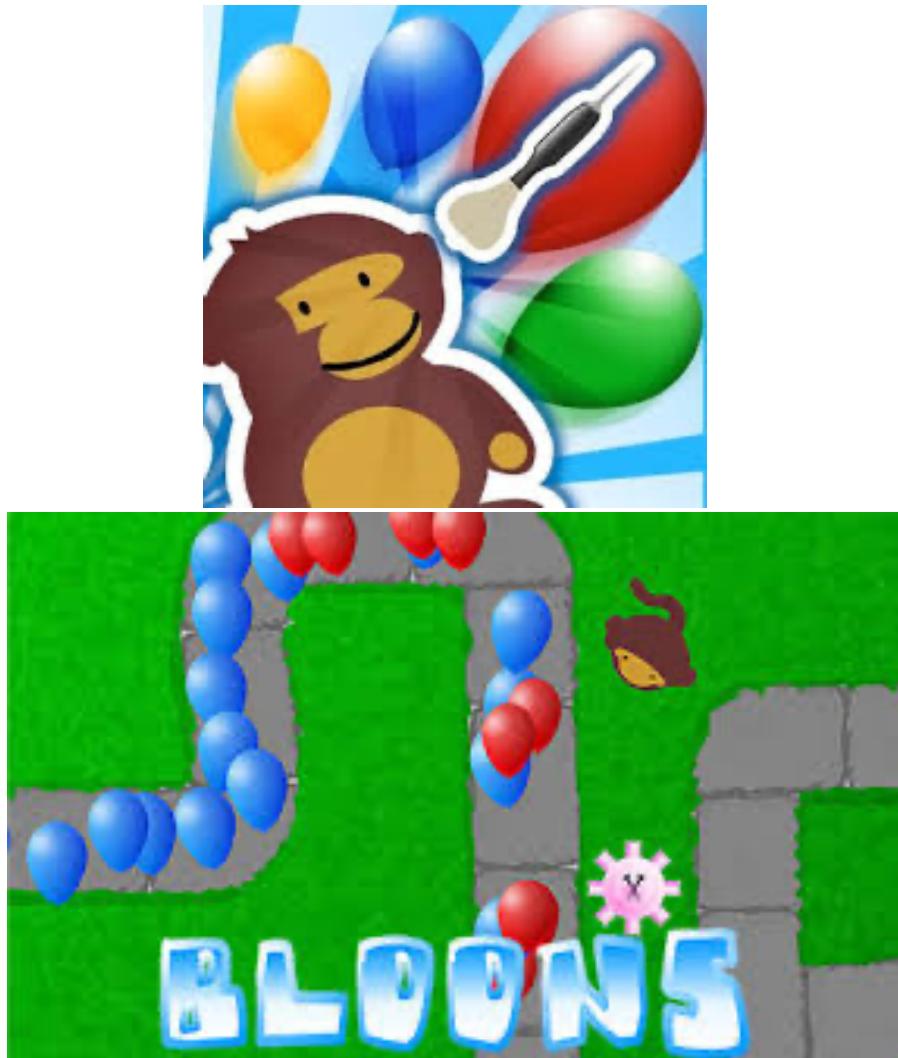


TABLE DES MATIERES

<u>INTRODUCTION</u>	2
<u>BUILD-UPONS</u>	2
<u>USER GUIDE</u>	3
<u>HARDWARE COMPONENTS USED</u>	3
<u>SOFTWARE LIBRAIRIES USED</u>	3
<u>WIRING DIAGRAM</u>	4
<u>TASK DIAGRAM</u>	5
<u>SYNCHSM DIAGRAM</u>	5

INTRODUCTION

This project implements a tower defense game inspired by *Bloons Tower Defense* on an Arduino Uno R3. The game uses a 16x2 LCD for menu navigation and status updates, and a 128x128 LCD to display an 8x8 grid map where gameplay occurs. Players aim to prevent waves of balloons from reaching the end of a set path by placing turrets of different types and abilities. A joystick and buttons allow for seamless control of both screens, enabling players to switch between managing the menu and the map.

Players can start new games, view their highest score, toggle sound effects, and monitor their health, money, and score. Balloons move along the map path, and the player must place turrets strategically to pop them. Each balloon type has different health, and each turret has distinct attack capabilities. Players lose health when balloons reach the end of the path.

While the core gameplay and mechanics were successfully implemented, some planned features, such as turret upgrades, save/load functionality, and multiple maps, were not completed due to time constraints and memory limitations. Other challenges included color inversion and brightness issues on the 128x128 LCD and difficulties debugging large header files, which slowed development. Despite these challenges, the project demonstrates a creative and functional approach to tower defense gameplay on limited hardware.

[LINK FOR THE DEMO VIDEO](#)

BUILD-UPONS

Successfully Implemented :

1. **Dynamic Menu on 16x2 LCD**
2. **Passive Buzzer for Music**
A buzzer plays music from my childhood games that I converted myself from hear (Dofus 1.29 music for main menu and in game music)
3. **128x128 TFT LCD for Game Map**

Partially Implemented :

4. **Turret and Balloon Mechanics**
Basic turret placement and balloon waves work (I had to configure a second timer), but turret damages and upgrades aren't implemented

Not Implemented :

5. **Game State Memory**
The "Load Game" and "GOAT" menu option exists, but saving/loading via EEPROM was not implemented (yet).
6. **Additional Gameplay Features**
Turret upgrades and more content were planned but not completed due to time and memory limits.

USER GUIDE

When the system is on, players are greeted with a menu displayed on the 16x2 LCD. Using the joystick, players can navigate the menu options, which include starting a new game, loading a saved game, viewing the best score (GOAT), and toggling sound on or off. To select an option, players press down on the joystick.

Once "New Game" is selected, players choose a difficulty level (Baby, Child, Adult, Senior), which determines starting health and turret's prices. The game then transitions to the main gameplay screen displayed on the 128x128 LCD. Here, players see an 8x8 grid map with a road that balloons will follow. Balloons spawn from one side (left) and move toward the other.

The player's goal is to prevent balloons from reaching the end of the road by placing turrets. Using the joystick, players move a cursor across the 8x8 grid to position turrets. Three turret types are available, each with unique abilities: pink (short-range, multi-target), brown (medium-range, double-target), and blue (long-range, single-target). The player must manage their resources, as turrets cost money to place. Money is earned by popping balloons, and the player loses health when balloons reach the end of the road.

Players switch control between the 16x2 LCD (for menu options like starting waves, viewing stats, selecting turret's type) and the 128x128 LCD (for selecting turret positions) using the right button. The left button resets the system if a bug occurs. Player health, money, and score are displayed on the 16x2 LCD during gameplay. The game runs continuously until the player's health reaches zero, at which point the player can return to the main menu to try again.

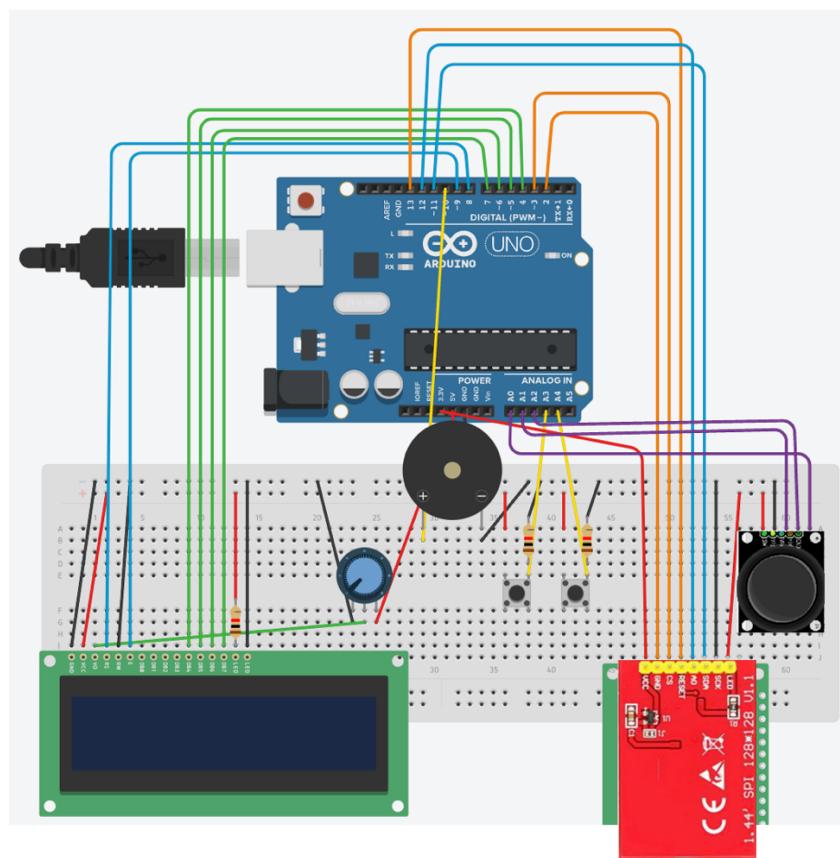
HARDWARE COMPONENTS USED

- Arduino Uno R3: Main microcontroller.
- 16x2 LCD: Displays menu options, player stats, and gameplay information.
- 128x128 SPI TFT LCD: Displays the 8x8 grid, turrets, balloons, and path.
- Joystick: Used to navigate menus and control the game.
- Right Button: Switches joystick control between the 16x2 and 128x128 LCDs.
- Left Button: Resets the entire system.
- Buzzer: Plays music and sound effects.
- Potentiometer: Adjusts contrast on the 16x2 LCD.

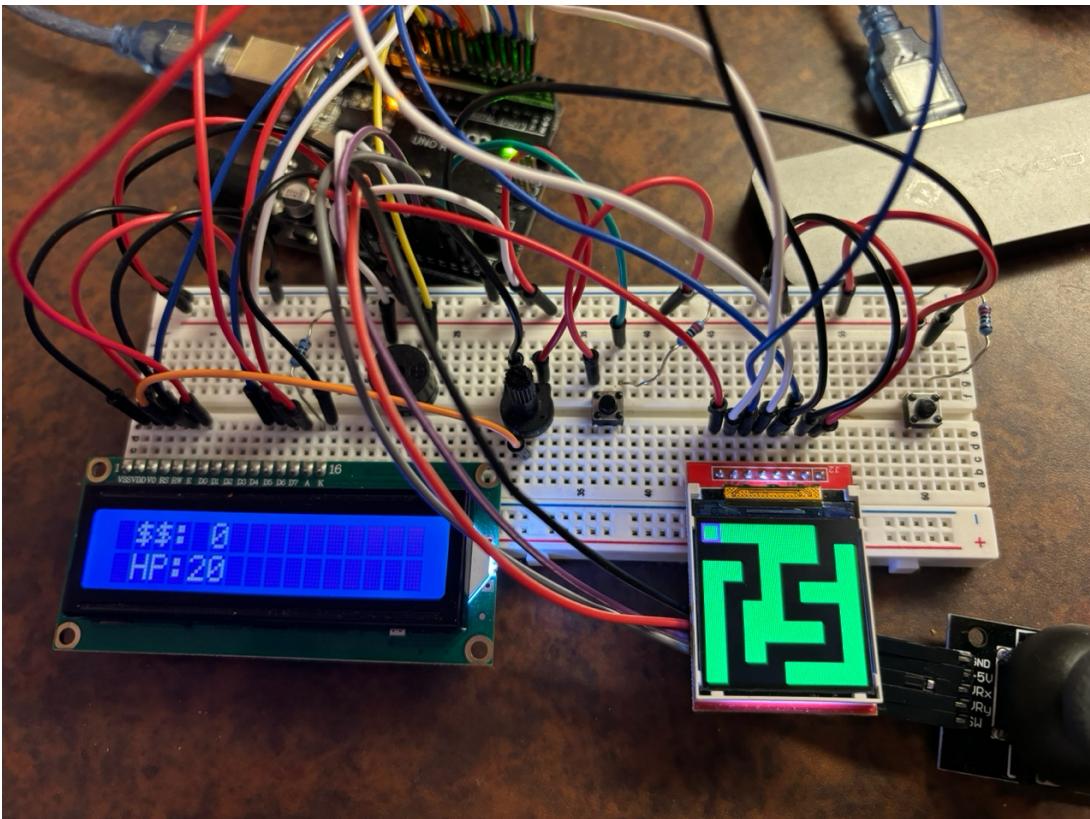
SOFTWARE LIBRARIES USED

- Adafruit GFX : enables drawing shapes, text, and custom graphics on the TFT display.
- Adafruit ST7735 and ST7789 : provides hardware-specific functions to control the ST7735-based LCD, handling initialization, rotation, and pixel-level drawing.
- Arduino Core Libraries (e.g., Arduino.h, avr/io.h) : offers fundamental functions and definitions for interacting with microcontroller hardware (I/O, timing, interrupts).

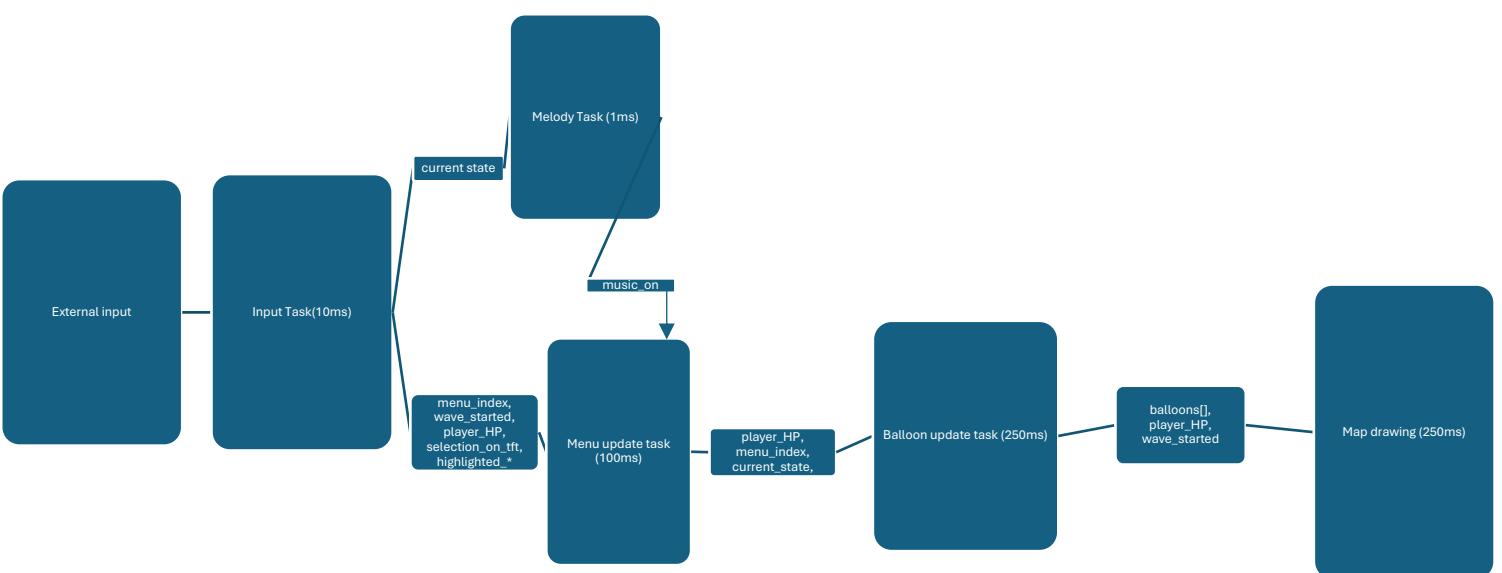
WIRING DIAGRAM



(128x128 is inverted)



TASK DIAGRAM

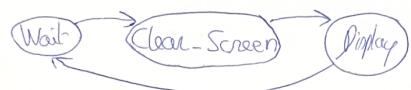


SYNCH SM DIAGRAM

Input Task:



Menu Update Task:



Balloon update task:



Melody Task:



Map drawing Task:

