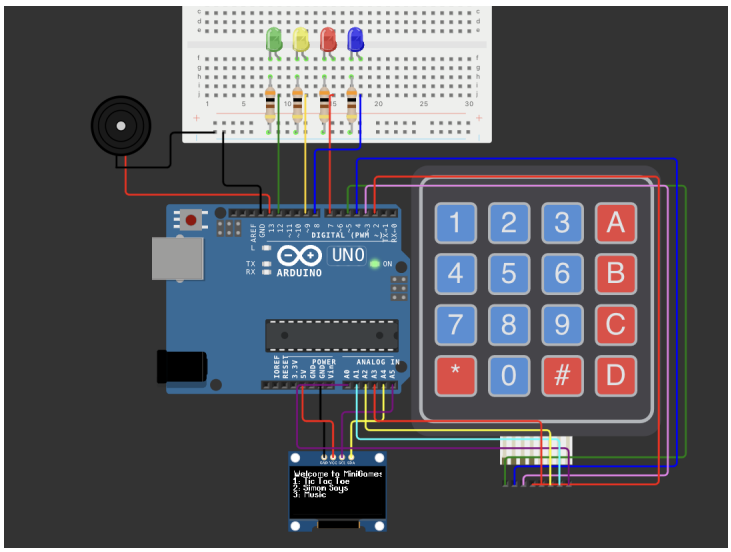
Methods of Computer Science Education: Design

Wintersemester 2024/25

Fabian Stiewe



Project name	**Mini Games** using Arduino
Coding files	All necessary files for coding should be located inside the `src` folder. Inside that folder create a subfolder for each game and use the `sketch.ino` file as the main input source.
Storyline	It is simple and quick to develop diverse games for the Arduino. This project includes the classic TicTacToe game (with a simple computer player), the known SimonSaysGame and a music player. • TicTacToe a simple two-player game where players take turns marking a 3x3 grid with their respective symbols (usually X and O). The objective is to be the first to get three of your symbols in a row, either horizontally, vertically, or diagonally. If all nine squares are filled without either player achieving this, the game is considered a draw. • SimonSaysGame is a memory game that uses 4 lights. The game generates a random sequence of lights, and the player must repeat the sequence in the correct order. Each round, the sequence gets longer, increasing the difficulty. The player wins by successfully repeating the sequence for a set number of rounds or loses if they make a mistake. • Music, just include your favorite songs (a song library is included in the project (clone from GitHub))
Target	It is for the **fourth year** and LSSA - Liceo Scientifico Scienze Applicate

Level	The base outline will be given to the students, so they only have to program the games. • TicTacToe (intermediate) • TicTacToe AI (hard) • SimonSaysGame (intermediate) • Music (easy) • any other game the students want to program (difficulty depends on the game)
Learning goals	The students should get used to program on the Arduino and Cpp. Over that they will have to learn how to use different hardware components (e.g. display, keypad, lights and so on). Depending on each student's interest they can choose to do a game with more or less hardware. Students have to acquire information themselves, depending on their project choice. Therefore, independent learning is encouraged very much.
Hardware	Each student has to understand how to use: • keypad • oled display Over that it depends on the preference of each student, what their project is about. Some prefer more hardware and some less, so they are not forced to a specific hardware. **Just experiment around :-)**
Software	The students will have to learn how to write clean and safe code in *Cpp*. They will have to use GitHub for collaboration (no terminal just via IDE).
Operating description	As we discussed earlier, this project is all about encouraging students to explore their programming skills independently. They should find an interesting game online and just start coding it. The teacher will be there to help them with any hardware or software issues they might encounter. Likely this will be a group project, so students can help each other (size 2-4 persons per group).
Handiwork	Nothing has to be created by hand by the students. But if they come up with a game idea which includes handiwork, they can do so.

Materials list	Depending on the student's choice they need different materials. Students should research on their own what they need, the following list is a suggestion: • wokwi-arduino-uno **(Only mandatory component)** • wokwi-buzzer • board-ssd1306 • wokwi-membrane-keypad • wokwi-breadboard-half • wokwi-resistor • wokwi-led (220Ω) – blue – green – red – blue • diverse cabels
Lesson planning	This project will stretch over approximately $6-7$ weeks: 1. Brainstorming and choosing a game for each group 2. Researching the game and the hardware needed. First (hardware) build phase and starting with a simple code 3. Using the template provided by the teacher, students will start coding their game 4. Continue coding (a) (Continue coding) buffer week 5. Finish coding 6. Presentation of the game
Project details	The final project deliverables should include the following components: • **Documentation:** A brief ReadME.md file outlining the project's purpose, design, and implementation details, and how to use the game(s). • **Code:** Fully functional and well-documented Arduino code for the game(s) developed. • **Supporting Materials:** Anything the students think is necessary to understand their project better (e.g., schematics, images, videos, etc.). • **Repository:** A link to a GitHub repository containing all relevant files.