Gaming Arcade

GP106: Computing Project

Batch: E/20
Project Mode: Group Project consisting of 3 sub-groups of 4 members each.
Time Duration: 7 Weeks
Milestones: 2 Milestones + Bonus Milestone

Introduction

This project aims to provide students with practical knowledge in constructing a system that combines hardware and software elements. The implementation of the project will be based on the concepts learned in class and the lab activities.

Description

In this project, each main group has to implement a Gaming Arcade consisting of three games. Each subgroup implements one of the Tic-Tac-Toe, Hangman, and Piano Tiles. All the games are simplified versions of their original game which will be easier to implement using the equipment provided. The project is divided into two phases.

Milestone 1

The sub-groups should work independently in this milestone, focusing only on implementing the game assigned to the relevant sub-groups. At the end of milestone 1, each group must upload the **flowcharts** used to design the implementation, the **schematic of the hardware design** (You can use tinkerCAD, or a hand-drawn circuit) used to implement the game using Arduino Uno and the provided components, the **python script(s)** they've developed, and a **video demonstrating** how their game works.

You should note that there isn't a single 'correct' way to implement the game. Try as much as you can, to emulate a gaming arcade experience.

Milestone 2 + Bonus Milestones

All three subgroups must combine the three games developed into one command-line user interface to implement a simple dashboard. Users can select which game to play, quit, check user stats (wins and losses), etc. Furthermore, each sub-group will receive bonus marks for implementing bonus milestones for their respective game. After finishing this milestone each main group has to do a **simple demonstration** about your implementation. Furthermore, you are supposed to submit all the **python scripts(s)** which were used to implement the project.

Timeline

Task	**Deliverable**	**Deadline**
Milestone 1	Flowchart, hardware design, python scripts, and video demonstration	**5th May 2023**
Milestone 2 + Bonus Milestones	Python scripts	**21st May 2023**
Demonstration	5 minutes face to face demonstration	**21 May - 26 May Week**. The date will be specified before the 21st of May.

Group A: Tic-Tac-Toe

About the game

Tic-tac-toe is a simple, two-player game played on a 3x3 grid. The players take turns placing their mark (either an X or an O) on one of the empty squares in the grid. The first player to get three of their marks in a row (either horizontally, vertically, or diagonally) wins the game. If all the squares are filled, and neither player has won, the game is a tie.

More about tic-tac-toe: Always Win tic tac toe

Milestone 1

The game should be implemented using the provided components. The input should be taken from the Arduino. You can use provided push buttons to implement this. Also, the state of the game should be visible to the players using the given LED bulbs. Note the number of digital pins available in Arduino Uno is 14. Therefore, each team can devise a solution to implement the 3*3 grid and how to select each cell. For example, there could be two buttons for row and column selectors to select a specific cell in the grid. For milestone 1, it is sufficient to implement only a two-player game mode. Player vs. Computer Mode is not necessary.

You should implement a command line interface-based dashboard to display the current player, scores, or even the tic-tac-toe grid with the current state. Furthermore, you can implement commands to quit the game, reset the scores, start a new game, etc. What to display and not to display is up to the team to decide.

Milestone 2 is common for all three groups A, B, and C. Check the section on Milestone 2 for more information.

Bonus Milestones for Tic-Tac-Toe

For tic-tac-toe, implement a player vs. computer mode. A user should be able to select either to play with another person or to play with the computer. Marks will be allocated depending on the accuracy of the algorithm and the difficulty level of the computer mode.

Group B: Hangman Lite

About the game

Hangman is a word-guessing game that two or more players typically play. For this project, the computer chooses a random word and displays a series of dashes representing the letters in the word. The other player then attempts to guess the word by suggesting letters that might be in the word before he runs out of attempts. The maximum number of attempts is 3.

More about Hangman: How to Play Hangman

Milestone 1

The input can be given in Morse code using a flashlight and an LDR (Morse code for the alphabet is provided). The current representation of the word should be displayed on the terminal. If the letter is in the word, update the displayed word by replacing the corresponding dash(s) with a given letter. If the letter is not in the word, decrement the number of attempts by one. Three LED bulbs can be used to indicate the remaining number of attempts. To generate random words, you can use a dictionary library (PyDictionary, NLTK, etc.).

Additionally, a command line interface should be implemented to display the current states of the word, the number of attempts, etc. What to display and what not to display is up to the team to decide on.

Milestone 2 is common for all three groups A, B, and C. Check the section on Milestone 2 for more information.

Bonus Milestone for Hangman Lite

You are supposed to implement a GUI to display the word and animation of a hangman that will be drawn when the player makes wrong choices. You can use pygame, turtle, kivy, tkinter, etc.).

Additionally, optimize the game by introducing different levels of difficulty. You can decide on the criteria to classify the difficulty level. (Ex: Word length, the complexity of the word, etc.). Furthermore, add categories of words from which the word will be randomly selected. You can use different topics of your choice.

Note: More marks will be allocated for the GUI implementation.

Group C - Piano Tiles

About the game

Piano Tiles is a game that most of you are familiar with. Here we are considering a simplified version of it. It is a simple game that challenges players to remember and repeat a pattern of a given pattern.

Milestone 1

The game consists of a 3x3 grid of LED lights, which light up in a random pattern. Only one bulb in each row should be lit up at any moment. Each column contains a push button at the end, and the players must press the buttons correctly to repeat the pattern when the pattern hits the last row. If the players press the buttons in the correct order, a buzzer will ring, indicating success. If they press the buttons in the wrong order or run out of time, the buzzer should go off, indicating that the player has ended the game. You can use different ring patterns to show successful button presses and failures.

You should implement the game for a single-player using the components provided for your group.

Milestone 2 is common for all three groups A, B, and C. Check the section on Milestone 2 for more information.

Bonus Milestones for Piano Tiles

Sample music: **Coffin Dance from TOO EASY to INSANE**

You can implement simple music patterns and create music using the laptop itself without the buzzer. And also, the game itself can store the previous player's details. You have the freedom to decide the rules of how the game should end, how points are given, etc.

A, B, and C Groups

Milestone 2

All the sub-groups A, B, and C have to work together to connect all three games onto one dashboard. You can implement this dashboard in a command line interface. A user should be able to select which game to play, then the dashboard will direct the user to the selected game. The python script written for milestone 1 should be imported as a module, and the hardware should be connected to one laptop/desktop.

The dashboard can display high scores for each game, no. of games played, the total time spent on each game, options to exit, etc. Similar to milestone 1, there is no single ‘correct’ way of designing the dashboard; try as much as you can, to emulate a real gaming arcade interface.

Marks Allocated

Milestone	**Criteria**	**Description**	**Marks**
Milestone 1	Flow charts	Flowcharts for the algorithm and code implementation	2
	Game design	Implementation of the game at its bare minimal level as per specifications provided and additional functionality introduced to enhance the game experience.	4
	Hardware implementation	A design of the circuit was implemented using the provided equipment.	4
	Milestone completion	Completion of milestone 1 as expected	2
	Total (Milestone 1)		**12**
Milestone 2	Overall Functionality & Design	Overall design and implementation of the gaming arcade.	4
	Interface and modularization	Design of the user interface and its user-friendliness. Modularizing the scripts from milestone 1	4
	Bonus	Completing the bonus milestones	4
	Total (Milestone 2 + Bonus Milestones)		**12**
Total (Including Bonus Marks)			**24**