UNIVERSITY of GUYANA

Faculty of Natural Science

Department of Computer Science

Course Title: Introduction to Programming

Course Code: CSE 1201

Lecturer: Andressa Morris

Due Date: 8th May, 2024

Project Title: Quiz Game

Group Members: Nickelcy Francois (1048549) - Group Leader,
Tashiera Layne (1047153),
Kalief Sober (1047967),

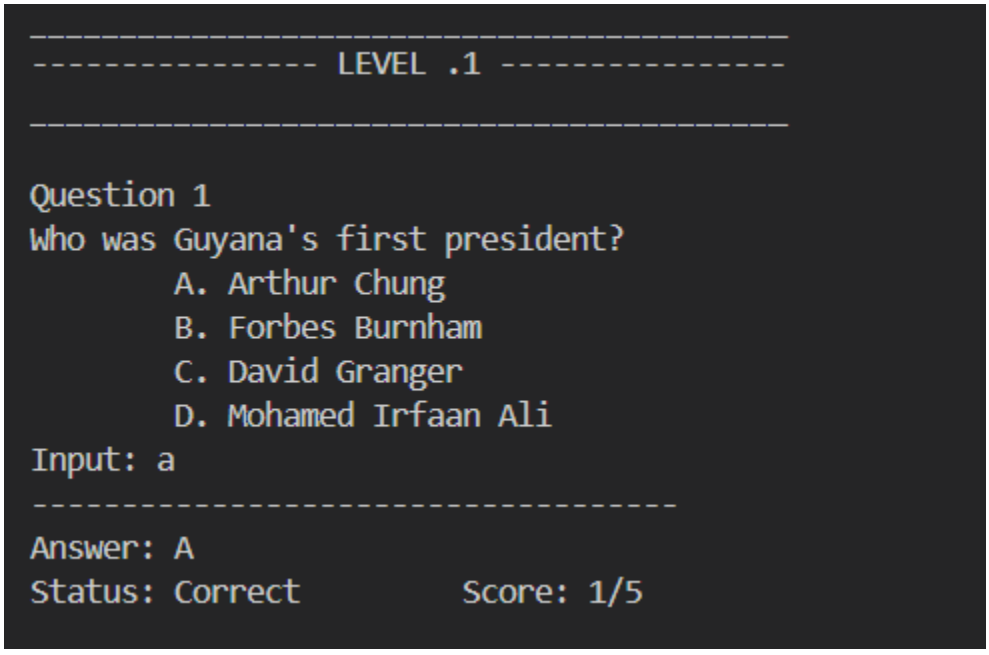
Table of Content

MANUAL FOR PLAYERS

Steps for game play:

- Identify the “main.exe” within the folder
- Click on the file
- The game should start at this point and you will be prompted by a multiple choice question.
- Choose the correct answer that is relative to the options whether it is A, B, C or D
- Achieve a score of four or five to advance to the second level.
- If you fail a level, you will be prompted to play again.

Example Question Prompt



DOCUMENTATION FOR DEVELOPERS

Introduction

The project is a terminal-based quiz application written in C, utilizing fundamental concepts of the language. It offers three levels, each featuring five random combinations of questions per round. The game aims to quiz players on topics related to Guyana or any other subject.

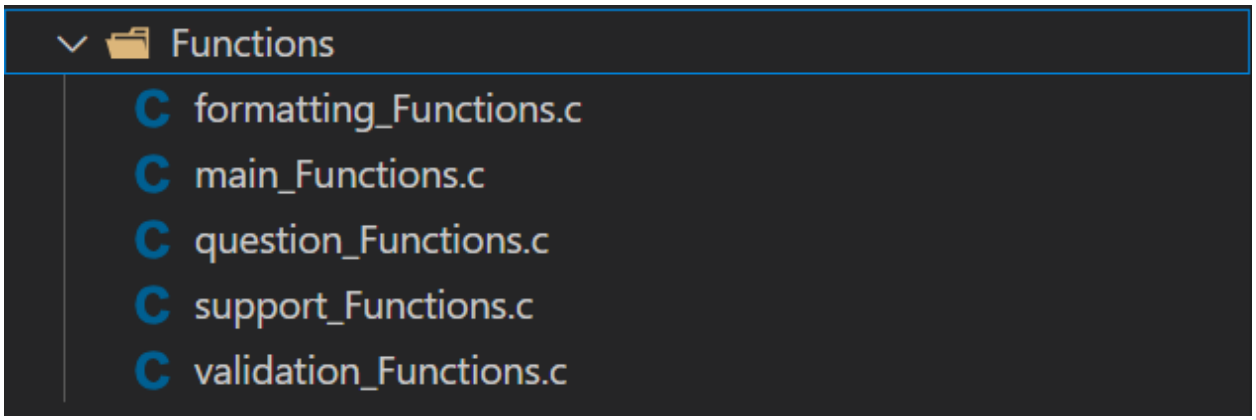
The entire application is encapsulated within a Do While loop, ensuring the game continues until completion or specific conditions aren't met. Initially, five questions are displayed to the player. To maintain engagement, each of these questions has five possible variations. Upon completion, the player can choose to try again if they fail to achieve a passing score of 4 out of 5.

Game Structure

The project consists of two folders and the main game file “Main.c”. The two folders namely “Functions” and “Questions_Levels” hold important modules for the game.

Functions Folder

This folder holds the major logic for the game and each of the functions is grouped together based on their purpose and included where needed.

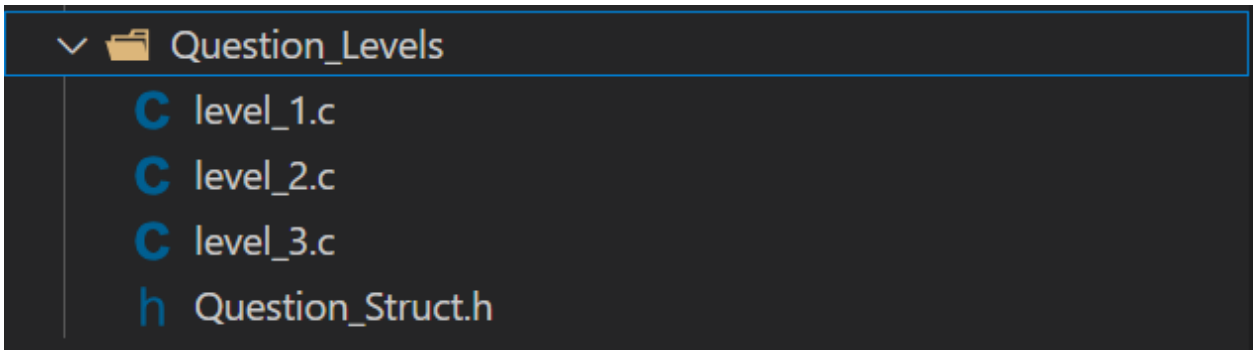


Major Modules:

- **Main_Functions.c :** This module is included in the "main.c" file and all other modules from the folder function are included in this file.
- **Question_Functions.c:** This module contains all question based functions such generating questions, getting responses and calculating the score if response is correct. (*Questions levels are included here since they are utilized in this module*)
- **Support_Functions.c:** This module contain logic relating to the randomized question generation and the logic for trying again after failing
- **Validations_Functions.c:** This module contains logic that verifies that the answer is correct and that the level is passed.

Question level Folder

This folder holds all levels of the game. Each level includes a header file called "Question_Struct.h" which defines the relationship between question, options and answer.



Question_Struct.h

This file initializes a structure named "levels". Each "level" structure contains an array of five "subject" structures. Each "subject" structure consists of a 20-character array named "name" and a "QA" structure called Questions, which define variables for questions, options, and answers.

```
You, 31 minutes ago | 1 author (You)
#ifndef QUIZ_STRUCTS_H
#define QUIZ_STRUCTS_H

You, 9 hours ago | 1 author (You)
typedef struct
{
    char question[5][200];
    char option[20][50];
    char answer[5];
} QA;

You, 3 weeks ago | 1 author (You)
typedef struct
{
    char name[20];
    QA questions;       You, 3 weeks ago • Init commit f
} subject;

You, 3 weeks ago | 1 author (You)
typedef struct
{
    subject subjects[5];
} level;

#endif
```

Level_1.c

Each level is defined using the structure from the header file "Question_Struct.h". For level one, there are five subject names initialized: "History & People", "Foods & Cuisines", "Holidays", "National Symbols", and "Landmarks", as shown in the "Level_1 Overview" picture.

After initializing the subject values, we can proceed to initialize the values for questions, options, and answers. For instance, after initializing the first subject, we can use the dot operator to access the subject's name and the variables within the structure variable "questions", including questions, options, and answers, as illustrated in the image below.

```
[0] = {
    .name = "History and people",
    .questions = {
        .question = {
            "Which of the following has never ruled Guyana?",
            "Which of the following is not one of Guyana's natural regions?",
            "Guyana gained its independence from Britain in 1966.",
            "Who were the first people to settle in Guyana?",
            "Who was Guyana's first president?"
        },
        .option = {
            "A. Dutch", "B. Italians", "C. Spanish", "D. French",
            "A. Hilly, Sand & Clay Region", "B. Mountainous & Meadows Region", "C. Low Coastal Plain", "D. Highland Region",
            "", "A. True", "B. False", "",
            "A. Europeans", "B. Amerindians", "C. East Indians", "D. Africans",
            "A. Arthur Chung", "B. Forbes Burnham", "C. David Granger", "D. Mohamed Irfaan Ali"
        },
        .answer = {'B', 'B', 'A', 'B', 'A'}
    }
},
```

Level_1 Overview

main.c | level_1.c

You, 8 hours ago | 1 author (You)

```c
#include "Question_Struct.h"

level one = {      You, yesterday • Completed Template 1
    .subjects = {
        [0] = {
            .name = "History and people",
            .questions = {
                .question = {…
                .option = {…
                .answer = {'B', 'B', 'A', 'B', 'A'}
            }
        },
        [1] = {
            .name = "Foods & Cuisines",
            .questions = {
                .question = {…
                .option = {…
                .answer = {'A', 'A', 'A', 'B', 'C'}
            }
        },
        [2] = {
            .name = "Holidays",
            .questions = {
                .question = {…
                .option = {…
                .answer = {'A', 'B', 'A', 'B', 'B'}
            }
        },
        [3] = {
            .name = "National Symbols",
            .questions = {
                .question = {…
                .option = {…
                .answer = {'D', 'A', 'D', 'A', 'B'}
            }
        },
        [4] = {
            .name = "Landmarks",
            .questions = {
                .question = {…
                .option = {…
                .answer = {'C', 'D', 'A', 'B', 'C'}
            }
        }
    }
};
```

Game Mechanics

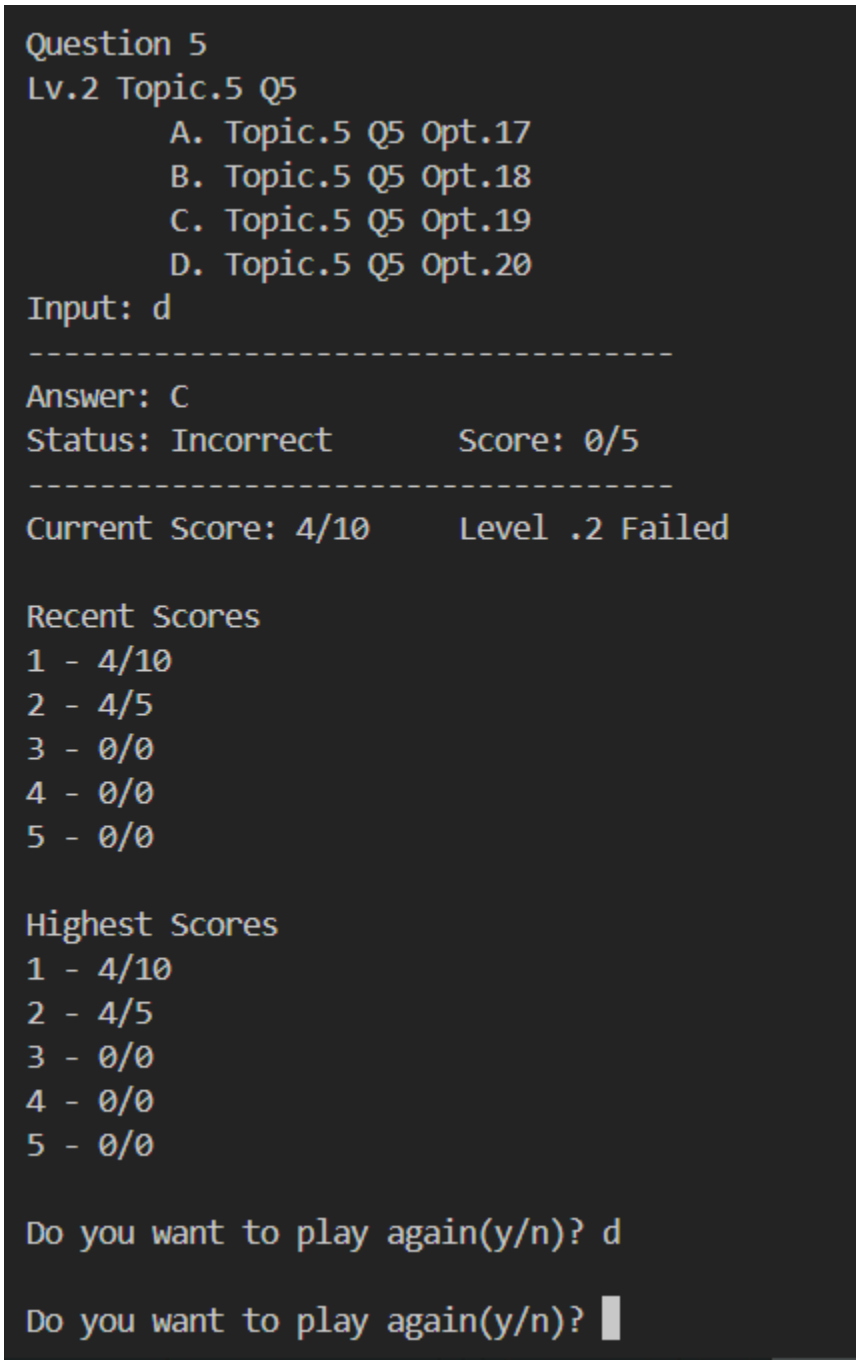
The game starts off by stating the current level. The first question is then asked which can be any of its four other variants. The user then places the input based on the option as A, B, C or D. The input also accepts common letters. If the option input is correct the next line will output the “Correct” after that the current score is outputted as either n/5, n/10, n/15 up until n/x where x/5 represent a level. See example output below.

```
----------------------------------------
--------------- LEVEL .1 ---------------
----------------------------------------

Question 1
Who was Guyana's first president?
        A. Arthur Chung
        B. Forbes Burnham
        C. David Granger
        D. Mohamed Irfaan Ali
Input: A
-----------------------------------
Answer: A
Status: Correct         Score: 1/5

Question 2
What is the national dish of Guyana?
        A. Pepperpot
        B. Cook-up Rice
        C. Metemgee
        D. Roti
Input: a
-----------------------------------
Answer: A
Status: Correct         Score: 2/5

Question 3
What is the national holiday celebrated on February 23rd in Guyana?
        A. Independence Day
        B. Mashramani
        C. Emancipation Day
        D. Christmas
Input: b
-----------------------------------
Answer: B
Status: Correct         Score: 3/5
```

After all five questions are outputted and answered a record of all recent scores are keeped.

```
Question 5
Lv.1 Topic.5 Q5
        A. Topic.5 Q5 Opt.17
        B. Topic.5 Q5 Opt.18
        C. Topic.5 Q5 Opt.19
        D. Topic.5 Q5 Opt.20
Input: a
-----------------------------------
Answer: C
Status: Incorrect       Score: 4/5
-----------------------------------
Current Score: 4/5      Level .1: Passed

Recent Scores
1 - 4/5
2 - 0/0
3 - 0/0
4 - 0/0
5 - 0/0
```

A level can only be passed if the score is greater than or equal to 4. If all levels are passed or if one level was failed, the end output would include the five most recent scores, as well as the five highest scores. Additionally, there would be an option to play again. Note that if an invalid input is provided for the play again prompt, you will be prompted again.



Error handling

As you may have notice there are two error handlers:

a. **Question input error handler:** When the user is prompted to answer a question they are allowed to place either common or capital letters of A, B, C or D.

   Note: *At the time of writing the documentation for this code a possible bug was identified. The user can choose common or capital A, B, C or D but what if inputted values such as "S" of "AB"? In the case of "S" the code would interpret this as incorrect but in the case of "AB" the code would interpret this as two inputs and use the second input for the following question.*

b. **Play again error handler:** If the player inputs a invalid response to the prompt (y or n) you will be prompted again.

   Note: *Capital letters are not taken into consideration*

Game Features

The game implements all required features:

a. Minimum of 5 questions per game or round
b. Informs user of correct or incorrect answer
c. Keeps track of users individual scores (up to 5 scores, most recent or highest)
d. Maintains a Leader board of 5-10 highest scores

The game implements two optional features

a. Different difficulty levels
b. Multiple combinations of questions for each round (Making the individual level unique for every iteration)