ARKWRIGHT
ENGINEERING SCHOLARSHIPS
part of The Smallpeice Trust

APPLICANT 4 – A COMPUTER SCIENCE PROJECT

What does/did your project aim to achieve? **Please write up to a maximum of 300 characters**

I aimed to design a two-player dice game that involves calculations and uses subprograms like functions and procedures. This project is my non-examined assessment for my Computer Science GCSE. I had to design a program that meets requirements and guidelines, these are described in details in the attached document.

Please describe and explain your project making clear and direct reference to your supporting documentation. **Please write up to a maximum of 1000 characters**

I was not given any instructions of how to start the program and so I had to decide how the program is going to run without any logic error or syntax error. Also, I needed to identify the purpose of the program and the most efficient way to achieve it. To show skills and techniques that I have learnt is the target that I set at the start to avoid any unnecessary steps to be done and to present a well planned project.

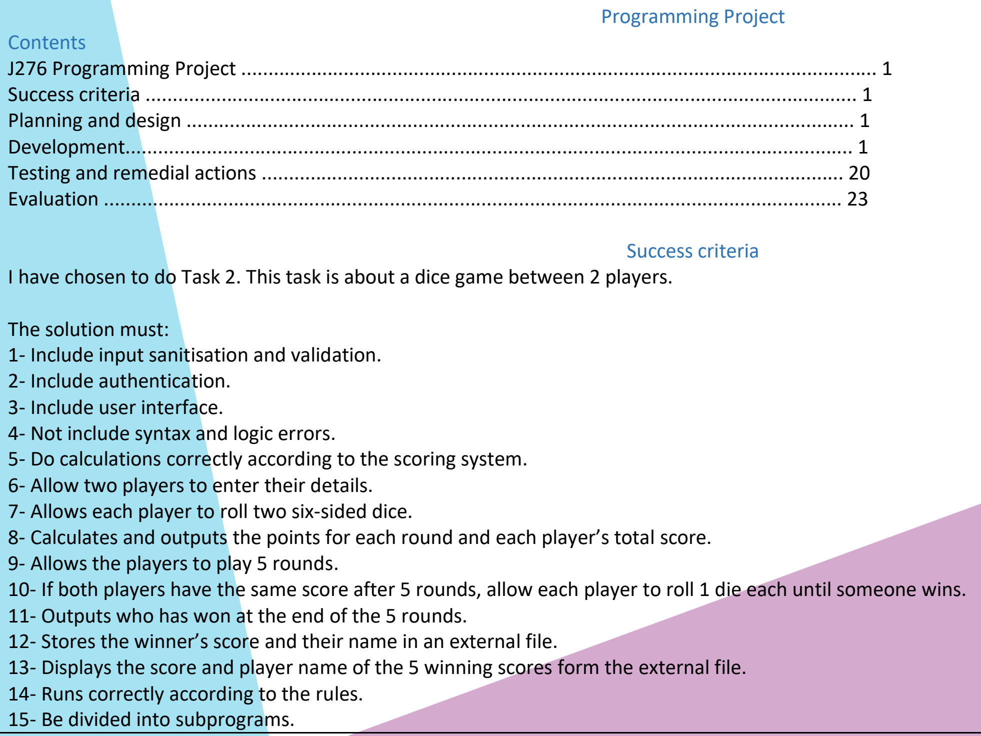
What have been the successes and failures of you project so far? **Please write up to a maximum of 500 characters**

As it progressed, I produced a check list to ensure that I was producing a robust program before I tested the program. I checked all validations and sanitisations are done so that the program would not stop running halfway and would not result in a wrong outcome. Then, I tested the first subprogram, authentication, which was processed without any errors.

What lessons of an engineering nature have you learnt from working on this project? **Please write up to a maximum of 500 characters**

Problem solving skills and logical thinking appeared to be essential when working on this project. Engineers need to think step by step, factor by factor like the way computers process instructions and data. Some people think computers are super brains but I think engineers are the real super brains who understand how adaptations take place and so human beings could adapt to the 'computational' society and computers could simulate human actions.

VISUAL EVIDENCE



Programming Project

Contents

Success criteria

I have chosen to do Task 2. This task is about a dice game between 2 players.

The solution must:
1- Include input sanitisation and validation.
2- Include authentication.
3- Include user interface.
4- Not include syntax and logic errors.
5- Do calculations correctly according to the scoring system.
6- Allow two players to enter their details.
7- Allows each player to roll two six-sided dice.
8- Calculates and outputs the points for each round and each player’s total score.
9- Allows the players to play 5 rounds.
10- If both players have the same score after 5 rounds, allow each player to roll 1 die each until someone wins.
11- Outputs who has won at the end of the 5 rounds.
12- Stores the winner’s score and their name in an external file.
13- Displays the score and player name of the 5 winning scores form the external file.
14- Runs correctly according to the rules.
15- Be divided into subprograms.

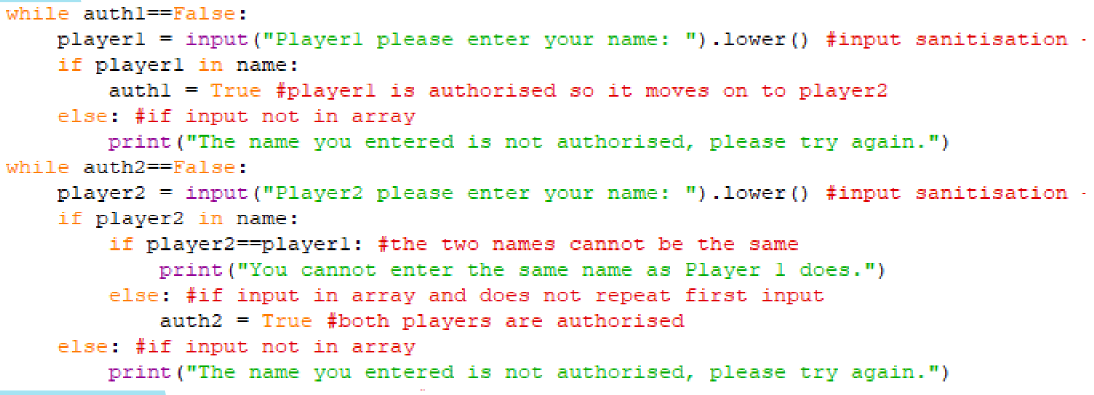
I have broken down the program in the way I have done because I think these are the main tasks that are essential for the program to run correctly and smoothly.

The main program is going to:
- Display welcome message
- Display result message
- Call subprograms

The Authorisation subprogram is going to:
- Ask the two players to enter their details one by one, the second player can only enter the details when the first player is confirmed authorised
- Has an array storing names in lower case of authorised players
- Do input sanitisation (as string and as lower case [presence check, type check]) then evalidation (compare to the names in the array)
- If the entered name is in the array, the player is authorised
- If the player has not entered authorised details, displays error message and askes the player to enter details again until two players are both authorised (If the first player is authorised and the second is not, ask the second player to enter details repeatedly until authorised details are entered.)
- Return the names of the players the main program – function
- iterative testing should be done during the program is being developed

The game subprogram is going to:
- Run a loop for 5 rounds and then compare results, the loop should run with following rules:
• The two players roll one by one, only when they both have finished their turn, one round is finished
• When the roll of the two dices of a player add up to be an even number, the number adds to the total which is zero before they rolled any die and 10 points are added to the total.
• When they add up to be an odd number, that number adds to the total and then 5 points are subtracted from the total. However, the total cannot go below zero at any point.
• When the two dices have the same number, the player get to roll one other die and the three numbers are all added to the total.
- After 5 turns, if the two players have the same total, they will roll one die one after the other and the number they get will add to their total. The two totals will then be compared again to identify the winner.
- Get the names of players from the main program and return the winner and the winner's total to the main program – function
- iterative testing should be carried out in order to ensure the correct calculations are done The external file program is going to:
- Get the name of the winner and the winner's total from the main program, does not return anything to the main program - procedure
- Prints the winner's name and total
- Prints the top 5 scores

After that, I had to add in codes that ask the players repeatedly to enter names if they have entered an unauthorised name. This could be done by not stopping the loop and output a try again message to instruct the players.



```
while auth1==False:
    player1 = input("Player1 please enter your name: ").lower() #input sanitisation
    if player1 in name:
        auth1 = True #player1 is authorised so it moves on to player2
    else: #if input not in array
        print("The name you entered is not authorised, please try again.")
while auth2==False:
    player2 = input("Player2 please enter your name: ").lower() #input sanitisation
    if player2 in name:
        if player2==player1: #the two names cannot be the same
            print("You cannot enter the same name as Player 1 does.")
        else: #if input in array and does not repeat first input
            auth2 = True #both players are authorised
    else: #if input not in array
        print("The name you entered is not authorised, please try again.")
```

MARKER'S COMMENTS

The visual evidence provided here are sections taken from the applicant's project folder. This applicant clearly works at a very high level and the project folder was easy to read and understand and was very logically laid out. If you use a Computer Science project in your application, then communication becomes even more important for the purposes of third-party assessment. We like the way that the lessons learnt are clearly linked to engineering and that the complexity of the programming is clear to see.