**IQ Test Program**

Will:

1. Test user’s IQ
2. Involve 30 questions, 6 different sections. Each section with five randomly picked questions. Multiple choice questions.
3. Implement pygame to display quiz
4. Have a timer, that can fail user if run down. Will be used to find time taken for user.
5. Take users score and gives IQ value. Will add name, IQ and time taken to a file.
6. Have a menu, which allows user to take test, view high scores, or quit. High score option will sort high scores and output for user, first in descending order of score, and then in ascending order of time taken.

**Analysis of Requirements**

List of functions necessary:

1. Menu function – Displays title and buttons to start game, view high scores or quit game.
2. Quiz function – Decides what questions to display, based on how many questions already displayed. Makes sure only 30 questions displayed.
3. High Scores function – reads high scores from file and sorts them into top 10. Displays for user.
4. Update High Scores function – Takes name, IQ score and time taken and appends to a file.
5. User name function – Allows user to enter their name. Only allows entering of letters.
6. Time Out function – When the timer runs out will stop quiz and tell user they ran out of time.
7. Text Question function – Reads question, answers and correct answer from file. Picks randomly. Ensures question not repeated. Timer, question displayed. Answers displayed as buttons.
8. Maths Question function – Picks random operator and numbers to generate random question. Displays timer and question.
9. Button function – Takes x, y position, width, height, text, text x, y position and size of text and function of button. Displays button based on inputs. Light up when hovered over to indicate functionality. Allows to be clicked that commences other functions.
10. Display text function – Takes text, width and height positions and also size of text. Displays text according to those values. Blits text.
11. Text objects function – Takes text and font and renders text in that font on a rectangular surface.
12. Next line function – Reads next line of a file. Returns line.
13. Open file function – Takes name and mode and opens file. Produces an error if unable to open.

Initialisation:

1. Initialise pygame, and also screen width, caption plus colours.
2. Initialise variables, number of questions, used questions and score
3. Initialise clock speed with FPS

**Pseudocode**

Initialise pygame

Display menu title + start buttons

If start is clicked commence quiz

If number of questions < 5

Choose random operator and numbers to generate random question

Display question and answers as buttons

If correct is clicked

Score + 10

Else continue

Elif 5 < n < 30

Read question and answers from file into variables

Display question and answers as buttons

If correct clicked

Score + 10

Else continue

Else

Get name, score and time taken

Add to file

Ask user what they want to do and display buttons

If retake test commence quiz

Elif main menu display menu + start buttons

Elif quit exit game

elif high scores is clicked

Reads high scores from file into a list

Sorts list score descending then time ascending

Displays top ten scores and buttons

If menu button clicked display menu and buttons

Elif quit clicked exit game

Elif quit clicked

Exit game

Open file function:

Try opening file

Except IOError

Output error

Next line function:

Open file function

Return read line

Text Objects function:

Create surface with rendered with rendered text

Return surface and surface rectangle

Display Text function:

Get font and size of text

Render text surface and text rectangle with text objects function

Center text in rectangle

Blit text surface and text rectangle

Button function:

Gets mouse position and if mouse has been clicked

Draw rectangle and display text within rectangle

If mouse inside rectangle draw rectangle aqua blue

If left clicked = True

If button function is start, start quiz

Elif button function quit, quit game

Elif button function correct, add 10 to score

Elif button function wrong, continue quiz

Elif button function menu, start menu function

Elif button function high scores, start high scores function

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Number | Data | Reason | Expected | Actual | Pass/Fail |
| 1 | User left click on start button | Make sure quiz starts | Quiz starts | Quiz starts | Pass |
| 2 | User left click on high scores | Make sure high score screen is reached | High score screen shown | High score screen shown | Pass |
| 3 | User left click on quit | Make sure program quits when clicked | Program exits | Program exits | Pass |
| 4 | Right answers clicked | Make sure score is added with each correct answer | User should get score of 200 | User gets score of 200 | Pass |
| 5 | Opening game | Check if mixer is initialised and plays music | Music plays | Music plays | Pass |
| 6 | Clicks on pause/play buttons | Check if music can be paused and played | Music can be paused/played | Music can be paused/played | Pass |
| 7 | Left click on start/answers | Check if timer works on start and throughout quiz | Timer counts down continually | Timer counts down continually | Pass |
| 8 | Left click on start/answers and then nothing | Check if program displays message when timer runs down | Screen comes up saying time out | Screen comes up saying time out | Pass |
| 9 | Any letters numbers symbols | Check if program only accepts letters for name | Program only accepts letters | Program only accepts letters | Pass |
| 10 | Complete quiz to give name, score and time | Check if program properly sorts high scores in order of score and then time | Sorts scores first then times | Sorts scores first then times | Pass |
| 11 | User left click on retake test | Check is score, timer, and questions used are reset so new test can be carried out | Everything reset and test works like new | Everything reset and test works like new | Pass |
| 12 | Mouse position | Check if buttons highlight when mouse hovers over them | Box turns aqua blue | Box turns aqua blue | Pass |
| 13 | Clicking answers | Check if maths questions are generated randomly | Maths questions different every time | Maths questions different every time | Pass |
| 14 | Text files | Check if text files opened and read | Questions + answers read and displayed | Questions + answers read and displayed | Pass |
| 15 | Clicking answers | Check if quiz outputs 30 questions, 5 from each section | Outputs 30 questions, 5 from each section | Outputs 30 questions, 5 from each section | Pass |
| 16 | Opening game | Check if it has caption “IQ Test” | Has “IQ Test” as caption | Has “IQ Test” as caption | Pass |

**Testing**

**Critique**

What worked:

1. Implementing pygame.
2. Displaing text and creating buttons in pygame.
3. Questions consistently randomised.
4. Created a timer that counts down across functions.
5. High scores are sorted effectively.
6. High usability, user can quit anytime; buttons for various functions.
7. Added music as an extra with buttons to pause and play.
8. Converted the game to executable so can be played on devices without python or pygame.

What didn’t work:

1. Couldn’t create a spotting patterns section as I didn’t have time to make all the patterns needed.

What I could’ve improved upon:

1. Added the pattern sections to show I could load images and display them.
2. Gave it a nicer design.