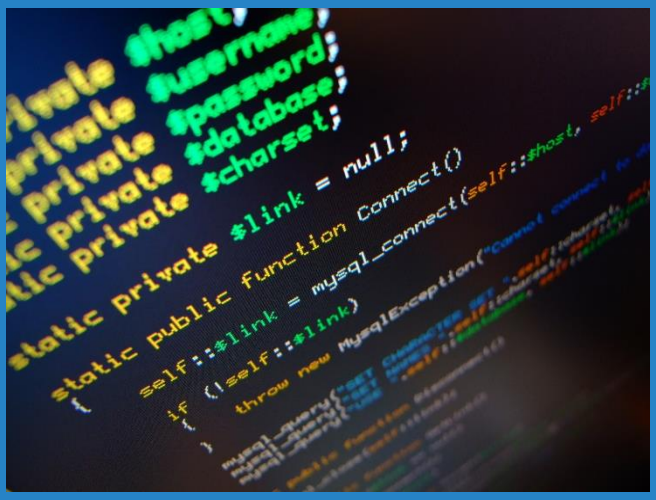


INTRODUCTION TO PROGRAMMING

Author: Tomas Atanasov
Student Number: 30221022

Software Engineering and Game Development/Computer Science Level 4

CONTENTS

INTRODUCTION

According to the assignment brief, game development company is looking for a junior developer which must work on a small Noughts and Crosses game in C++. The purpose of this task is to test the developers skills according to the three-stage process of the company, and these stages are designing, developing and testing.

The Noughts and Crosses game must show:

- Modular methodology.
- User Interface.
- Interaction with users.
- Loop that loops until a winner is found.
- A loop that will allow a rematch.

Additionally, the company wants the developer to use his skills and creativity and design additional features, extras, options, music, colour, and many other different advancements within the game.

In order to complete the assignment, a report on black box vs. white box testing methods must be included. Test plan and design documentation also has to be written, to make it accessible for others to recreate the game.

BASIC AND ADDITIONAL DEVELOPMENT BUILDS.

The basic Noughts and Crosses development includes libraries required to execute the program, functions, declarations and global values. Therefore, a main function is included which presents a welcome screen and main menu, it is then followed by several functions like the grid design of the game, function which displays the user input and function that validates the user input and check if there is a winner or a draw, according to different winning combinations and conditions.

Finally, a function is given for the player to continue or quit the game. The whole game runs within a loop which starts as soon as the game is started and is ended upon quitting or repeated when the player chooses to continue. Additional features like music, colour, menu settings and extras are also scattered across the code in the appropriate order to make the experience more entertaining and pleasant for the users.

FUNCTION CATALOGS

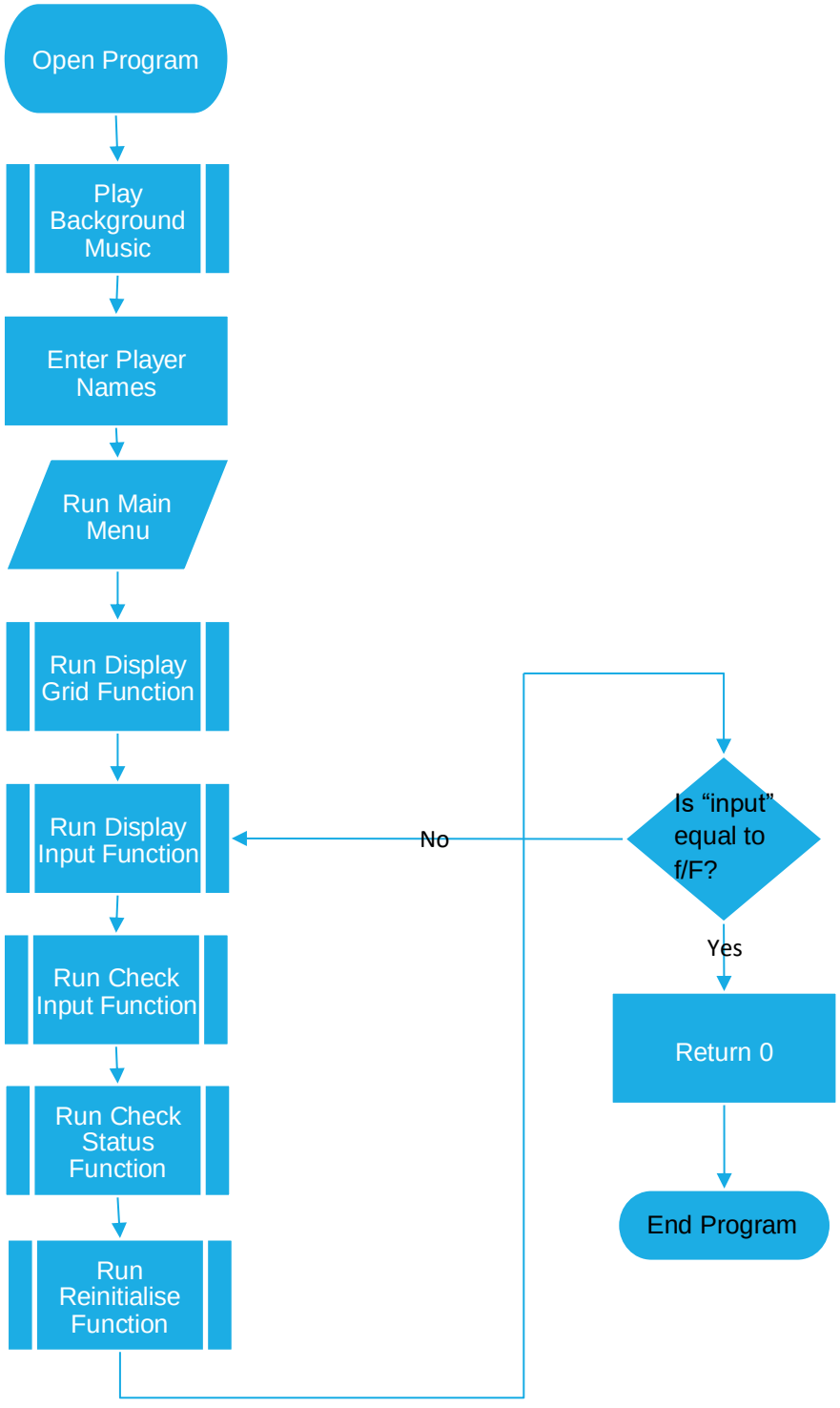
HIGH LEVEL FUNCTION CATALOG (BASIC DEVELOPMENT)

Function Type	Function name	Description
int	main	This function is the starting point of every C++ console application. It contains the main instructions which allow the program to function the way we want.
void	CheckStatus	The purpose of this function is to validate the player input, making sure the move is correct. If a space on the grid is empty, the function allows the placement of an 'X' or 'O', otherwise, it prompts the player to choose a different square until an empty one is selected.
void	CheckInput	This function verifies the user input to confirm if the move is correct or not.
void	DisplayGrid	This function is responsible for displaying information entered by players, like the noughts and crosses grid, the players' moves, and the names initially inputted at the start of the program.
void	Init	This function replaces all the existing noughts and crosses on the current game board with blank spaces, allowing for the game to be reset and played again.

HIGH LEVEL FUNCTION CATALOG (ADDITIONAL DEVELOPMENT)

Function Type	Function name	Description
int	Main	Added a welcome screen and a function to take and store player's names.
int	mainMenu	This function is called in the main loop of the program to present the main menu to the user and receive their menu choice. The user can choose to play a game, learn how to play, or quit the game based on the options provided in this menu.
int	howToPlay	This function displays instructions on how to play the game, including the rules and an example grid.
void	playSoundThread()	This function is intended to operate in its own thread and is tasked with playing background sound effects while the main program is running.
void	displayInput	Added F statement at the start of the function which helps the players know whose turn it is during the game.

FLOW CHART


Open Program
Play Background Music
Enter Player Names
Run Main Menu
Run Display Grid Function
Run Display Input Function
Run Check Input Function
Run Check Status Function
Run Reinitialise Function
No
Is "input" equal to f/F?
Yes
Return 0
End Program

TEST PLANS

I did a thorough investigation of Black and White box testing methodologies and from my new understanding I stepped right into the testing of my C++ game. While i was writing the code, i encountered some trivial issues, from misspelling, to forgotten syntax like '')'', '';'', or ''}'', but i went through my code line by line and carefully fixed all of them.

The tests that I have completed are the following:

Event or Input	Expected Result	Actual Result	Test Passed?
Test Player 1 name input	Player 1 name is entered and stored	Name is accepted and used correctly in the game.	Pass
Test Player 2 name input	Player 2 name is entered and stored	Name is accepted and used correctly in the game	Pass
Test the background music	Music starts when the program is opened and run uninterrupted until the program is closed	The music starts when the program is opened and stops when the program is closed.	Pass
Test Display Grid	The grid is displayed correctly in the game	The grid is visually clear and represents the gaming bord correctly.	Pass
Test Fill Square	The program accurately fills squares based on player input.	After a series of moves, the corresponding squares are filled as expected with 'X' or 'O' depending on the player's turn.	Pass
Test Check Input	The program prompts the user on invalid input and continues after a valid input.	Entered a valid and invalid characters during the game and it runs smoothly if the input is valid or	Pass

		correctly prompts and guides the user if the input is invalid	
Test Main Menu	The program displays the main menu correctly and takes the user input correctly.	Main Menu is displayed correctly, all the main menu options have been selected and they are executed correctly.	Pass
Test How to Play menu option	The program displays the How to Play guide.	How to Play option was accessed and the guide is visually clear and provides accurate information.	Pass
Test Quitting the Program	The program exits correctly when the user chooses to quit.	After selecting '3. Quit the game' the program displays a closing message and exit without errors.	Pass
Overall test of the game.	The program is handles transitions between menu options smoothly, restarts correctly, and provides a good user experience.	Played multiple games, navigated the main menu, and restarted the game after a win and draw. Program executes smoothly without errors.	Pass

TEST LOGS, WITH AMENDMENTS RECORDED.

Event or Input	Expected Result	Actual Result	Test Passed?	Amendment
Test the launch sound and background music	Music starts when the program is opened and run uninterrupted until the program is closed	Program starts with the music and black screen, only the music is working.	Fail	void playSoundThread() added and running the sound through different thread.
Test the starting sound effect and background music	Music starts when the program is opened and run uninterrupted until the program is closed	Program runs fine until the user user wants to quit, prompting an Error message.	Fail	sound thread was not properly terminated when the program exits. soundThread.detach() added after each menu option that causes the program to detach the sound thread and terminate properly.
Test File Format Compatibility	Different types of sound files running smoothly without error	Attempted to use MP3 files and got error message indicating unsupported file format	Fail	Replaced the mp3 files with WAV.
Test entering an invalid character during the main menu	Program should display "Please enter a valid choice (1, 2, or 3)" and wait for a key press.	The program displayed the message, but instead of waiting for a key press, it became unresponsive, started flashing and freaked out.	Fail	Modified the code to include cin.clear() and cin.ignore(INT_MAX, '\n') after displaying the error message to clear the input buffer and ensure the program waits for a key press before proceeding,

Link for the program

Tomas-Assignment-2-of-2-Lee-Holroyd-11-Dec-2023-master.zip