

Tic-Tac-Toe & Hangman SDLC

By Cameron McGowan & Julian Peen



Contents

[Design Brief 2](#_Toc514946601)

[Data dictionary: 3](#_Toc514946602)

[Deployment 3](#_Toc514946603)

[Pseudocode 3](#_Toc514946604)

[Tic Tac Toe 3](#_Toc514946605)

[Hangman 6](#_Toc514946606)

[Flowchart 0](#_Toc514946607)

[Evaluation 1](#_Toc514946608)

[Gantt Chart 4](#_Toc514946609)

[Self evaluation 0](#_Toc514946610)

[Bibliography 0](#_Toc514946611)

[Journal 1](#_Toc514946612)

# 

# Design Brief

Using visual basic 6, you have to design and create a game board which allows the selection of tic tac toe or hangman. The project shall be done in pairs.

The project will involve many parts, these include:

* Tic-tac-toe game
* Hangman game
* Pseudocode for each
* Flowchart for each
* GUI for each game, as well as a starting page to select game
* Testing the game for bugs or errors
* Evaluation of the game to find out if was enjoyable
* SDLC for the entire project

The project will be made by Julian Peen and Cameron McGowan. It will be designed using Visual Basic 6, with Adobe photoshop for GUI and Microsoft word for documents. The project will be completed by the 24th of May 2018. Our target audience will be primarily primary school aged children, and as such should be colourful and engaging.

The aim of the project is to create a fully functional game which is error free. We must also create a well-documented portfolio, thorough pseudocode and flowchart as well as an informative evaluation questionnaire.

The goal of our project is to create a bug free, working collection of Tic-Tac-Toe and Hangman. We are required to use Visual Basic 6. Our first goal is to complete the Graphical User Interface of tic-tac-toe, and then complete the coding. After that we must repeat the process but for a hangman game. Once this is done, we must test our program for any errors. After recording all bugs, debugging can take place.

Our program is designed to run on Visual Basic 6. This can be run on any Windows system post 98. On mac or linux, virtualization software such as VirtualBox or VMware can be used to simulate a Windows computer. The system requirements are low, and most modern day PCs or laptops will be capable of running it. The program will work on both 32-bit and 64-bit architecture.

# Data dictionary:

Words: Kayak, Azure, Equip, Abyss, Crypt, Cobweb, Absurd, Injury, Exotic, Topaz, Kiosk, Quiz, Larynx, Moon, Galaxy

Hints: A one person canoe, The colour of the sky on a clear day, To supply with necessary items for a purpose, A deep or seemingly bottomless chasm, An underground room or vault beneath a church, An old and dusty spiderweb, “Wildly unreasonable, or illogical”, “The instance of being damaged, or wounded”, Originating in a distant foreign area, A typically yellow-orange precious stone, “A small store where newspapers, drinks, etc are sold”, “A test of knowledge, or the act of questioning”, The technical name of the voice box, Celestial bodies that orbit planets, An area containing many solar systems

# Deployment

We deployed our game on 24/05/18, before 6:00pm. This was the given deadline, and the project turned out well enough to be released without any noticeable bugs or glitches.

In the project’s current state, there is no required maintenance. Unless a game breaking bug appears, there does not need to be any form of updated release.

The advantage of constructing an SDLC is that it allows for departmentalisation and control. A schedule can be set with deadlines for each stage. The disadvantage is that it doesn’t allow for much reflection or revision. It is very difficult to change something that was not well documented or thought upon.

# Pseudocode

## Tic Tac Toe

BEGIN

DEFINE PLAYERNAME AS STRING

DEFINE PLAYERLETTER AS CHARACTER

DEFINE COMPUTERLETTER AS CHARACTER

DEFINE POSSIBLEMOVES AS ARRAY

DEFINE ISPLAYING AS BOOLEAN

DEFINE TURN AS BOOLEAN

CLEAR BUTTON CAPTIONS

ENABLE ALL BUTTONS

IF BUTTONPRESS:

IF TURN = TRUE:

CAPTION = “X”

SET TURN AS FALSE

DISABLE BUTTON

IF TURN = FALSE:

CAPTION = “O”

SET TURN AS TRUE

DISABLE BUTTON

END IF

CHECKWIN

CHECKWIN

IF

BUTTON1 CAPTION = “X” AND BUTTON2 CAPTION = “X” AND BUTTON3 CAPTION = “X” OR

BUTTON4 CAPTION = “X” AND BUTTON5 CAPTION = “X” AND BUTTON6 CAPTION = “X” OR

BUTTON7 CAPTION = “X” AND BUTTON8 CAPTION = “X” AND BUTTON9 CAPTION = “X” OR

BUTTON1 CAPTION = “X” AND BUTTON4 CAPTION = “X” AND BUTTON7 CAPTION = “X” OR

BUTTON2 CAPTION = “X” AND BUTTON5 CAPTION = “X” AND BUTTON8 CAPTION = “X” OR

BUTTON3 CAPTION = “X” AND BUTTON6 CAPTION = “X” AND BUTTON9 CAPTION = “X” OR

BUTTON1 CAPTION = “X” AND BUTTON5 CAPTION = “X” AND BUTTON9 CAPTION = “X” OR

BUTTON3 CAPTION = “X” AND BUTTON5 CAPTION = “X” AND BUTTON7 CAPTION = “X”:

DISPLAY (PLAYER1 “ Has won the game.”)

ELSE

BUTTON1 CAPTION = “O” AND BUTTON2 CAPTION = “O” AND BUTTON3 CAPTION = “O” OR

BUTTON4 CAPTION = “O” AND BUTTON5 CAPTION = “O” AND BUTTON6 CAPTION = “O” OR

BUTTON7 CAPTION = “O” AND BUTTON8 CAPTION = “O” AND BUTTON9 CAPTION = “O” OR

BUTTON1 CAPTION = “O” AND BUTTON4 CAPTION = “O” AND BUTTON7 CAPTION = “O” OR

BUTTON2 CAPTION = “O” AND BUTTON5 CAPTION = “O” AND BUTTON8 CAPTION = “O” OR

BUTTON3 CAPTION = “O” AND BUTTON6 CAPTION = “O” AND BUTTON9 CAPTION = “O” OR

BUTTON1 CAPTION = “O” AND BUTTON5 CAPTION = “O” AND BUTTON9 CAPTION = “O” OR

BUTTON3 CAPTION = “O” AND BUTTON5 CAPTION = “O” AND BUTTON7 CAPTION = “O”:

DISPLAY (PLAYER2 “ Has won the game.”)

ELSE

BUTTON1 CAPTION != “” AND BUTTON2 CAPTION != “” AND BUTTON3 CAPTION != “” AND BUTTON4 CAPTION != “” AND BUTTON5 CAPTION != “” AND BUTTON6 CAPTION != “” AND BUTTON7 CAPTION != “” AND BUTTON8 CAPTION != “” AND BUTTON9 CAPTION != “”:

DISPLAY (“The game is a draw.”)

ASKUSER (PLAY AGAIN, QUIT)

IF PLAY AGAIN:

GOTO START

ELSE:

EXIT

## Hangman

ACCEPT DIFFICULTY AS STRING

ACCEPT WORDLIST AS ARRAY

ACCEPT HINT AS ARRAY

DEFINE WORD AS ARRAY

DEFINE TURN AS INTEGER

WHILE TRUE:

IF NEW\_WORD PRESSED:

IF DIFFICULTY EASY:

DEFINE i AS RANDOM INTEGER (0,4)

SELECT ITEM i FROM WORDLIST

ADD ITEM TO WORD

SPLIT INTO SINGLE CHARACTERS

ELIF DIFFICULTY NORMAL:

DEFINE i AS RANDOM INTEGER (5,9)

SELECT ITEM i FROM WORDLIST

ADD ITEM TO WORD

SPLIT INTO SINGLE CHARACTERS

ELIF DIFFICULTY HARD:

DEFINE i AS RANDOM INTEGER (10,14)

SELECT ITEM i FROM WORDLIST

ADD ITEM TO WORD

SPLIT INTO SINGLE CHARACTERS

ELIF DIFFICULTY INSANE:

DEFINE i AS RANDOM INTEGER (15,20)

SELECT ITEM i FROM WORDLIST

ADD ITEM TO WORD

SPLIT INTO SINGLE CHARACTERS

IF OSK\_CHARACTER PRESSED:

:keypress

DEFINE LETTER AS STRING

LETTER = OSK\_CHARACTER

IF LETTER IN WORD:

DISPLAY LETTER IN WORD

IF WORD COMPLETE:

DISPLAY WIN MESSAGE

RESET HANGMAN

ELSE:

BREAK

ELIF LETTER NOT IN WORD:

ADD 1 TO TURN

IF DIFFICULTY EASY:

IF TURN >= 8:

PLAYER LOSE

BREAK

ELIF DIFFICULTY NORMAL:

IF TURN >= 6:

PLAYER LOSE

BREAK

ELIF DIFFICULTY HARD:

IF TURN >= 5:

PLAYER LOSE

BREAK

ELIF DIFFICULTY INSANE:

IF TURN >= 3:

PLAYER LOSE

BREAK

IF KEY\_PRESSED:

GOTO keypress

IF EXIT BUTTON PRESSED:

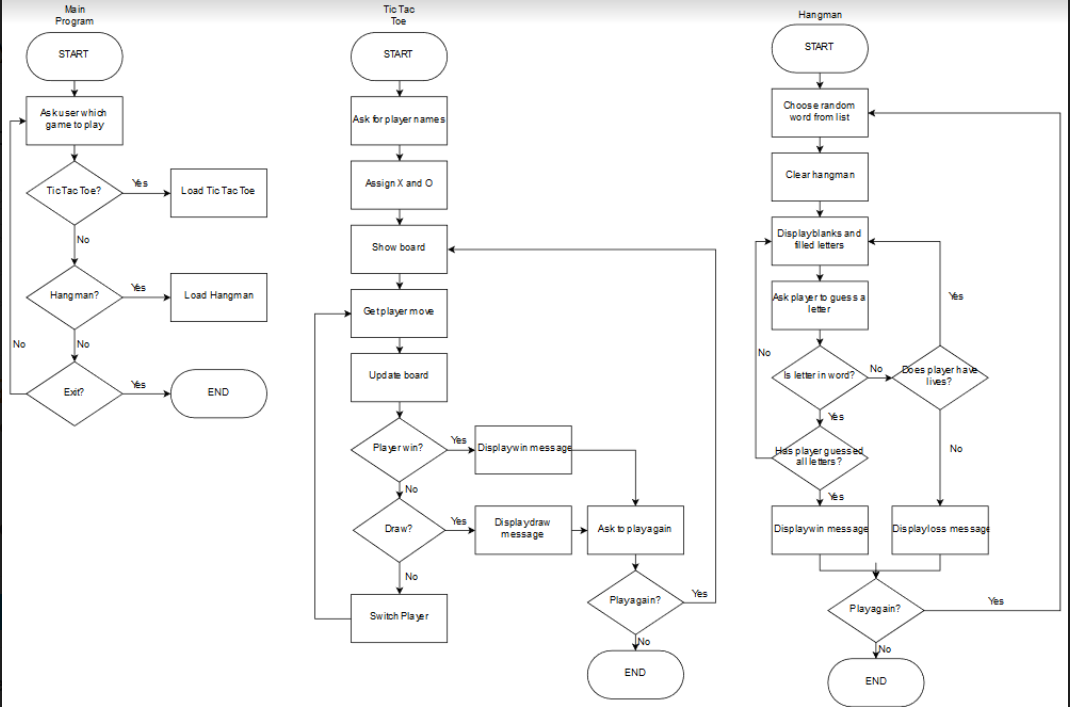
END

IF HINT PRESSED:

CHECK CURRENT WORD

CHANGE HINT LABEL TO WORD INDE

# Flowchart



# Evaluation

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name: | Did you think the user interface was intuitive? Why/ why not? | Did you think the game was aesthetically pleasing? Why/why not? | Did you find the game engaging? Why/why not? | Was there anything that you disliked about the game? | Which features would you like added or changed? |
| Lachlan | Yes, it was intuitive. This is because the text on the buttons were related to what it opened and thus allowed me to know what I was doing when clicking the buttons | The game’s design was very minimalistic and was quite pleasing to the eye. The colours used were not too vibrant and thus did not hurt my eyes when using playing the game | Though the design was very minimalistic which didn’t hurt my eyes, it also wasn’t too engaging. This is because there weren’t many engaging graphics such as images. | There were a few things that could have improved user experience. This included adding better graphics and possibly making the words bigger so it is easier to read. | As mentioned in the previous question, bigger font size would make it easier to read and in the Hangman game, the hint should be in a more readable colour. |
| Pranav | Yes, as the text was on the buttons, and the games were easy to understand, yet still challenging | For Visual Basic, yes, these games were aesthetically pleasing | Yes, as it had a minimalistic design and simple user interface | The hints in Hangman were slightly too difficult, and sometimes were not directly related to the topic | EASIER HANGMAN HINTS!!!  Also, the color scheme was a little bland |
| Isaac | Yes the interface was intuitive and it wasn’t too complicated. This made it pretty easy to play straight away. | The game design was once again not too complicated, but it looked reasonably good. | I found the game engaging and it was quite fun while the content lasted. The hangman words were varied and not too easy. | The graphics while alright could have been improved on to look better. Maybe a background image would have helped. | I would add a score for hangman so you can tell how many words you got correct or incorrect. |
| Sarah | Yes, it was easy to use and flowed smoothly. | It was a good size and the neutral colour scheme made playing for longer more comfortable and less distracting from the actual games. | The game choices were classic. There’s a reason they’ve stood the test of time! | Consistency in the level of difficulty for the Hangman clues would have been good. | A choice of levels (easy, intermediate and difficult) for the hangman clues would have been good. Would have also been good to have the letters I’d chosen for being shown as I played each hangman game. |

# Gantt Chart



## Self evaluation

I think that overall that we worked quite well on the project. The project involved many aspects, some of which we were better at than others. We especially enjoyed the coding and we found that we were pretty good at it. I think for future tasks we could have improved on our portfolio work. However in the end our game is error free and reasonably easy to use. We believe our pseudocode and flowcharts accurately reflect our program, and is simple enough that anyone with knowledge of basic coding practice can understand how it works, and recreate it.

We finished much of our project working with each other outside of school (at our houses), which gave us additional time which was much appreciated when it came to tweaking our program and adding to our portfolio.

We both worked quite well together, working on most of the project together and giving each other ideas on how we could best implement our ideas into the game. This approach reduced errors as we were both checking each other’s work as the project progressed. This took longer but due to working together outside of school hours we still had enough time to finish.

# Journal

Project wk. 1

Begun the initiation phase. We looked at the task and discussed how we would meet the criteria, e.g. what kind of game we would make.

Project wk. 2

We decided how we would implement our ideas and make out code work. We proceeded to make a flowchart.

Project wk. 3

We started typing our pseudocode for our tic-tac-toe game.

Project wk. 4

We made the pseudocode for our hangman game.

Project wk. 5

Development started. We began with the tic tac toe game and started creating the GUI interface as well as the backend coding.

Project wk. 6

Development continued. We moved onto the hangman task. Once again we first created the GUI (buttons, keys etc). We then moved onto creating an array to store our words and letters in.

Project Wk.7

We finished coding the hangman task, we also added extra features and finished coding the hints. We now had a working hangman game.

Project Wk. 8

Now that development was finished, we started testing our game. When errors were found we went back and changed the code to fix them up. We also tweaked the interface to make it more user friendly.

Project Wk. 9

We made some final tweaks to our code. We also added some music and sound effects in to make the game more engaging. We used evaluation sheets to get feedback from friends on our game.

# Bibliography

Vbforums.com. (2018). *[RESOLVED] vb6 background image-VBForums*. [online] Available at: http://www.vbforums.com/showthread.php?574006-RESOLVED-vb6-background-image

numbers, H. (2018). *How to generate random numbers*. [online] Stack Overflow. Available at: https://stackoverflow.com/questions/14963493/how-to-generate-random-numbers

Vbforums.com. (2018). *[RESOLVED] playing two audio clips at the same time-VBForums*. [online] Available at: http://www.vbforums.com/showthread.php?574868-RESOLVED-playing-two-audio-clips-at-the-same-time

Walsingham, V. (2018). *The Dark Truth Behind The Hangman Game*. [online] Ranker. Available at: https://www.ranker.com/list/history-of-hangman-game/veronica-walsingham?utm\_source=facebook&utm\_medium=historypost&pgid=642850749204637&utm\_campaign=history-of-hangman-game

YouTube. (2018). *[Electro] - F.O.O.L - Throne [Monstercat FREE Release]*. [online] Available at: https://www.youtube.com/watch?v=xGLizMZmA5c

Anon, (2018). [online] Available at: https://www.quora.com/What-is-the-origin-of-the-game-hangman