



LeeDuku Project

Advanced Higher Computing Science

Lee Glen
2016/2017

Project Proposal

My project proposal will be Sudoku which is a puzzle game that makes the user find the valid numbers in each of the rows and columns of a certain sized grid (for the Sudoku game I will be making will be 4x4).

The end-user group who would be interested in this game would be people aged 8 upwards (mainly teenagers or adults because I won't think children from the age down from 8 could possibly do Sudoku) who are interested in puzzles involved with numbers or who like anything to do with mathematics.

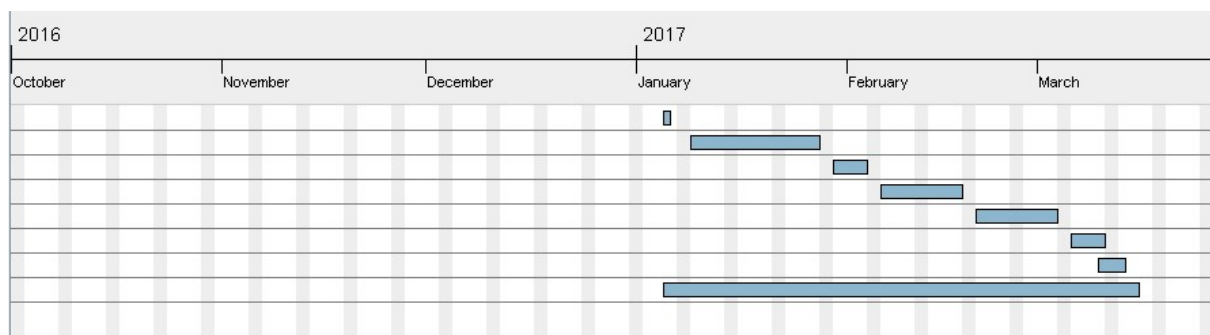
This game will require me to develop it using the relevant knowledge and skills using 2D arrays to make the grid and using a bubble sort for a scoreboard that will show the fastest people who can do the Sudoku game and using file handling the game can save the progress and can be loaded in at another date. It will also show how I apply generic skills in planning, research/analysis, problem solving and evaluation. All necessary resources can be obtained easily (Scholar notes, Module Library) and all potential barriers can be overcome.

Reflective commentary –

Looking back, I should've picked a much easier game to develop as a lot of the time doing the project was the implementation or wished that I learned SQL or PHP to a point I was comfortable programming it before ever entering adv. higher computing to use some kind of table using SQL.

Project Plan

Task	Time	Target Date
Requirements Specification	1 Day	6/1/17
Test Plan	3 Weeks and 6 days	27/1/16
Interface Design	1 Week	3/2/17
Program/Data Structure Design	2 Weeks	17/2/17
Implementation	2 Weeks	3/3/17
Final Testing	1 Weeks	10/3/17
Evaluation	3 Days	13/3/17
Final Submission of Project	2 Days	15/03/17



• Requirements Specification	05/01/17	05/01/17
• Test Plan	09/01/17	27/01/17
• Interface Design	30/01/17	03/02/17
• Program/Data Structure De...	06/02/17	17/02/17
• Implementation	20/02/17	03/03/17
• Final Testing	06/03/17	10/03/17
• Evaluation	10/03/17	13/03/17
• Final Submission of Project	05/01/17	15/03/17

The program will be split into manageable sub-tasks which will be then refined and be tested promptly.

Requirements Specification – I can easily do the requirements specification in one day as it won't really take that long and I have both double last on a Wednesday and Thursday to complete this part of the project. Need to show what will be included inside the program like sort for leader boards and possibly recursion for the timer.

Test Plan – As at the time I’m typing this I will be off for the Christmas holidays I’ll have to complete it as soon as I come back which during the holidays I can make a mock-up of what I’ll need to test at home and at school and use this once I come back at school to complete the test plan. I’ll need to talk about what to test for the program like the sorting and validation for example and give test data to use for the program.

Interface Design – Designing the interface should only take up to one week as I already have a vague idea what my program will look like. Program/data structure design will probably take longer than it should as I don’t have any clue how to program the game but I’ll think of something and I could always come back to this if I have changed or added code that is new. Should show what the program will in the end will look like with detail of what each buttons and fields will do in the program and what will ask the user as well.

Implementation will also take a long time because of the creation of the program and finding a way to get it into a working state and checks off all the requirements and this might not be finished in the end. Firstly I’ll have to create the user interface with a grid, buttons and a field. Then I’ll have to program each button for each difficulty and to reset the game. Then I’ll have to program the grid so that the numbers they input are only numbers that are from 1 to 4 and that the user can’t enter any number that is also in the same row or column. I’ll need to program the leader boards so that when the user selects a difficulty the leader boards will load a file containing all the times for that specific difficulty and after the user completes one of the difficulties fully and has entered their name and is on the leader board then it will save the current leader board results onto the correct file.

UPDATE –

I have changed the program so that it only has one difficulty as it was difficult to implement multiple difficulties.

Final testing might be sabotaged or might not be done because of the implementation but hopefully I have a working program that does what it says and testing this program would only take a small amount of time for me. Just need to show if the program will correctly do the thing related to the test data inserted like for example the normal and the extreme it should accept the data and exceptional should not accept the data. Need to show that the sort works and that the files do save and load in the right places.

Evaluation would also take a small amount of time as I would have already knew what I did through the diary and I have a vague clue on what to write about already for the evaluation. Need to talk about what I could of included in the program like for example linking a database to the leaderboards using SQL and talking about the reason why I didn't try it and also talk about what I could of done earlier and what could have been changed.

Resource Management –

I would need to use the Scholar notes which can be accessed online and through book form which I already have. I will also need Livecode which is on every computing classroom computers and can be downloaded for free online. This will be accessed for program/data structure design and implementation part of the project. Word will also be used to keep record of changes during the development of the program and to be able to records test results which is currently on every computer in school and I have installed at home as well.

Potential barriers –

There won't be any barriers related to the program as I have all that I need already at hand and I will not need to go anywhere at all.

Survey of end users

Using Surveymonkey I created a survey for my game which I released so that I know what people would like to have included inside my game and what it should do.

Here is what the survey looked like:

1. How often do you play puzzle games?

- ☐ Very Often
- ☐ Often
- ☐ Not Often
- ☐ Very Not Often

2. Do you like Sudoku puzzles?

- ☐ Yes
- ☐ No

3. What do you expect when you play a Sudoku game?

4. What would you like in a Sudoku like game?

5. How big do you like your grid?

- ☐ 4x4
- ☐ 5x5
- ☐ 6x6

6. Do you like the idea of a scoreboard for the game?

- ☐ Yes
- ☐ No
- ☐ Maybe

7. Do you like the sound of a difficulty setting for the Sudoku game?

- ☐ Yes
- ☐ No

8. What do you find more enjoyable?

- ☐ A easy simple sudoku puzzle
- ☐ A hard complex sudoku puzzle

9. Can you complete a Sudoku game if there is:

- ☐ No numbers done already?
- ☐ Quarter of the numbers done already?
- ☐ Half of the numbers done already?

10. If you have any additional ideas please type here.

Done

Q1

Customize

Export ▾

How often do you play puzzle games?

Answered: 9 Skipped: 0

Answer Choices ▾	Responses ▾	
Very Often	33.33%	3
Often	22.22%	2
Not Often	33.33%	3
Very Not Often	11.11%	1
Total		9

Q2

Customize

Export ▾

Do you like Sudoku puzzles?

Answered: 9 Skipped: 0

Answer Choices ▾	Responses ▾	
Yes	77.78%	7
No	22.22%	2
Total		9

Q3

Export ▾

What do you expect when you play a Sudoku game?

Answered: 9 Skipped: 0

It will be fun and challenging, but not too hard...

12/11/2016 7:12 PM [View respondent's answers](#)

Challenge

11/29/2016 9:40 AM [View respondent's answers](#)

anger

11/28/2016 3:11 PM [View respondent's answers](#)

A challenge, something to make you think

11/28/2016 10:13 AM [View respondent's answers](#)

you solve a number puzzle

11/23/2016 1:56 PM [View respondent's answers](#)

To be able to complete the game.

11/23/2016 12:25 PM [View respondent's answers](#)

a 9x9 grid of numbers which needs to be sorted so that no 2 of the same number are in the same row, column or box.

11/23/2016 12:17 PM [View respondent's answers](#)

A reasonable sized grid (maybe 5x5 or bigger)

11/23/2016 11:00 AM [View respondent's answers](#)

For it to look nice and for the game to be complete-able.

11/23/2016 10:56 AM [View respondent's answers](#)

Q4

What would you like in a Sudoku like game?

Answered: 9 Skipped: 0

It to be fun and exciting.

12/11/2016 7:12 PM [View respondent's answers](#)

Progression from easy to hard

11/29/2016 9:40 AM [View respondent's answers](#)

easy stuff

11/28/2016 3:11 PM [View respondent's answers](#)

A challenge

11/28/2016 10:13 AM [View respondent's answers](#)

it to be different every time i try it

11/23/2016 1:56 PM [View respondent's answers](#)

A game that I can complete.

11/23/2016 12:25 PM [View respondent's answers](#)

For it to be challenging yet possible

11/23/2016 12:17 PM [View respondent's answers](#)

A challenge. Not too many numbers should be filled in when I start

11/23/2016 11:00 AM [View respondent's answers](#)

A timer and a live clock so you don't "forget the time"

11/23/2016 10:56 AM [View respondent's answers](#)

Q5

Customize

Export ▼

How big do you like your grid?

Answered: 9 Skipped: 0

Answer Choices	Responses	
▼ 4x4	0.00%	0
▼ 5x5	44.44%	4
▼ 6x6	55.56%	5
Total		9

Q6

Customize

Export ▼

Do you like the idea of a scoreboard for the game?

Answered: 9 Skipped: 0

Answer Choices	Responses	
▼ Yes	66.67%	6
▼ No	11.11%	1
▼ Maybe	22.22%	2
Total		9

Q7

Customize

Export ▼

Do you like the sound of a difficulty setting for the Sudoku game?

Answered: 9 Skipped: 0

Answer Choices	Responses	
Yes	88.89%	8
No	11.11%	1
Total		9

Q8

Customize

Export ▼

What do you find more enjoyable?

Answered: 9 Skipped: 0

Answer Choices	Responses	
A easy simple sudoku puzzle	33.33%	3
A hard complex sudoku puzzle	66.67%	6
Total		9

Q9

Customize

Export ▼

Can you complete a Sudoku game if there is:

Answered: 9 Skipped: 0

Answer Choices	Responses	
No numbers done already?	22.22%	2
Quarter of the numbers done already?	88.89%	8
Half of the numbers done already?	66.67%	6
Total Respondents: 9		

Q10

If you have any additional ideas please type here.

Answered: 4 Skipped: 5

Question 10 isn't worded very well. Size of grid question is also slightly confusing.

11/28/2016 10:13 AM [View respondent's answers](#)

both of the corners filled in- bottom right/top left

11/23/2016 1:56 PM [View respondent's answers](#)

A timed mode might be a good feature?

11/23/2016 11:00 AM [View respondent's answers](#)

A two player mode would be cool. GLHF

11/23/2016 10:56 AM [View respondent's answers](#)

UPDATE –

From the results giving above the results were varied. I found out different people play a different amount of puzzle games, the majority like Sudoku-like games, the common thing people said when expecting something from Sudoku is that you are able to complete the game and that it is a big grid. What I also found out is that people like it to be challenging, different each time and is overall fun. I found out also that they like their grid to be above 4x4 so like 5x5 and 6x6, the majority also like having a scoreboard inside the game, almost everyone thought that difficulty settings was good for the Sudoku game and most of the people that done the survey liked a harder version of Sudoku. 8 out of the 9 users who done the survey could do a Sudoku game with only quarter of the numbers in the grid, 6 out of the 9 users can do the Sudoku game if there is half the numbers and only 2 out of the 9 users can do the game if there is no numbers in the grid. Only 3 people had extra ideas which was both of the corners filled in – bottom right/top left which is giving me advice for maybe hard mode, a timer and a two player mode. As I got rid of difficulties the both corners difficulty got removed so that idea was scrapped entirely.

Feasibility Study

Copyright

I have researched into the copyright of Sudoku and found out it has been trademarked in the UK on the 16 of June 2014 and as the name has been trademarked my game will need to be renamed to something different like Leedoku or 4x4 Numbers Game.



Case details for trade mark EU012643821

[New Search](#)[Glossary of terms](#)[Print Friendly View](#)

Trade mark

Trade mark: SUDOKU
Status: Registered

Relevant dates

Filing date: 27 February 2014
Date of entry in register: 16 June 2014
Renewal date: 27 February 2024

List of goods

Class 5: Herbicides.

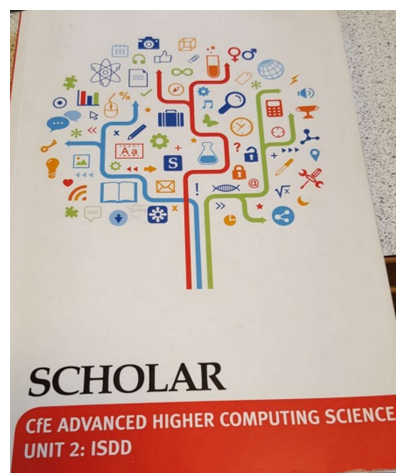
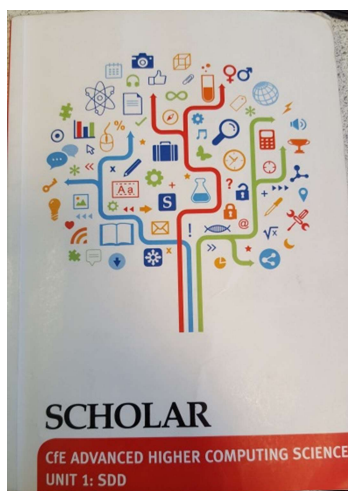
Name and Address details

Publication details

First advert: Journal : 2014/045 Date of publication : 07 March 2014

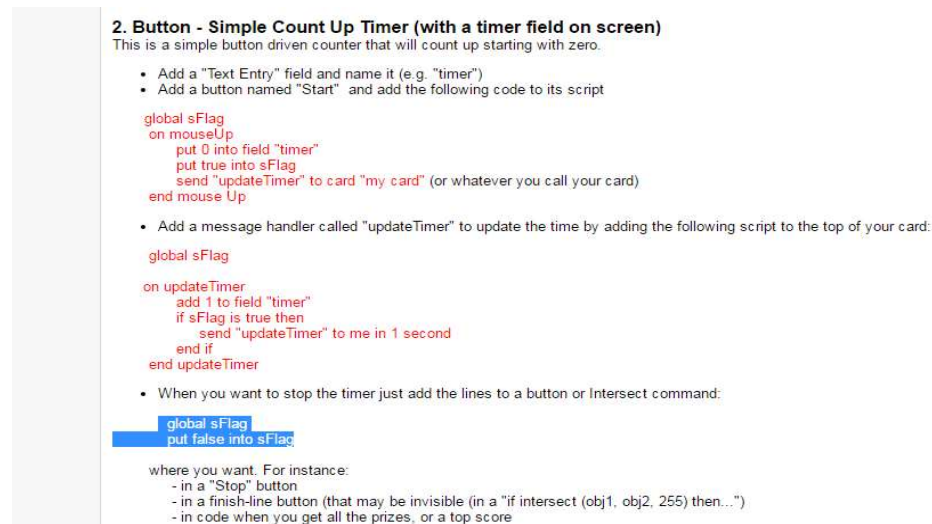
Software, Hardware or other resources required

For programming the software I would need the Live code software and if I need any reminders on how to do any part of the project I have the scholar notes and the standard algorithm and data types and structures booklets for reminders on how to program the advanced higher parts of programming. I will also need to use multiple computers as I will be in and out of class and will need to do at home and in school. I also need word to type up records of progress, test plans etc.



Additional Techniques

I will need to know how to make a timer in livecode that will count upwards to use for the leaderboards.



2. Button - Simple Count Up Timer (with a timer field on screen)
This is a simple button driven counter that will count up starting with zero.

- Add a "Text Entry" field and name it (e.g. "timer")
- Add a button named "Start" and add the following code to its script

```
global sFlag
on mouseUp
  put 0 into field "timer"
  put true into sFlag
  send "updateTimer" to card "my card" (or whatever you call your card)
end mouseUp
```

- Add a message handler called "updateTimer" to update the time by adding the following script to the top of your card:

```
global sFlag
on updateTimer
  add 1 to field "timer"
  if sFlag is true then
    send "updateTimer" to me in 1 second
  end if
end updateTimer
```

- When you want to stop the timer just add the lines to a button or Intersect command:

```
global sFlag
put false into sFlag
```

where you want. For instance:

- in a "Stop" button
- in a finish-line button (that may be invisible (in a "if intersect (obj1, obj2, 255) then...")
- in code when you get all the prizes, or a top score

UPDATE –

Economic feasibility –

This will be dealing with the cost in making the game. As it stands the game will not cost any money to make as live code is free to use and I was given the scholar books for free and is available online for free.

Technical feasibility –

This involves what types of technology I will be using to make this game. I will mainly be using a computer and a laptop that has live code installed. I will also be using a USB to transfer files and the game from my laptop and the computer and if the computer I'm using has not got live code on it, the USB has got the latest version of live code installed on it and if I lose the USB I have a back-up on Google Drive which is cloud-based storage. I will be using mainly two programs throughout the project which is live code and Microsoft Word, live code for the programming aspect of the project and word to document everything that is going on related to my project.

Legal feasibility –

This involves what laws I need to follow related to the project. One of those laws is the Copyright, Designs and Patents act of 1988 and because my game is fairly similar to Sudoku there is a possibility that someone has created a patent for this game. I will also

need to think of a different name because I can't call my game Sudoku as it is already copyrighted so I have named it Leeduku to try to avoid any legal action against me.

Economic feasibility –

This involves how long the project will take in total and how I will be on target to finish the project. What I have created is a Gantt chart to create and follow deadlines to finish parts of the project which should strict myself into working as hard as I can on the project.

UPDATE 2 –

I have researched if there was anyone who might have patented Sudoku and I did find one here:

Sudoku-type puzzle board game and method of play

US 7677564 B2

ABSTRACT

A game apparatus is provided for one or more players having a first game member that includes a puzzle or a game area having indicia forming a grid having sub-grids including a plurality of cells. Each cell is assigned indicia in a solution pattern of the puzzle such that a distinct indicia appears once in each row, column, and sub-grid. The first game member displays the solution indicia for some of the cells and the remaining cells are divided into a number of sub-cells bearing the possible solution indicia for the corresponding cell such that each sub-cell includes a distinct indicia. A player speculates which sub-cell bears the correct solution indicia for the respective cell. The apparatus further includes at least one game piece adapted to randomly display an indicia when manipulated, the indicia modifying a game parameter such as a player's score or number of possible speculations the player can make in one turn.

Publication number	US7677564 B2
Publication type	Grant
Application number	US 11/888,659
Publication date	16 Mar 2010
Filing date	31 Jul 2007
Priority date	26 Jan 2006
Fee status	Lapsed
Also published as	US20080054562, US20090224477, WO2007089671A2, WO2007089671A3
Inventors	Marc Kriger
Original Assignee	Marc Kriger
Export Citation	BiBTeX, EndNote, RefMan
Patent Citations (27), Non-Patent Citations (11), Referenced by (5), Classifications (15), Legal Events (3)	
External Links: USPTO, USPTO Assignment, Espacenet	

But as the fee status is lapsed (no longer valid) I will not need to pay this person any money at all hopefully.

Reflective Commentary

The project plan was easy to do as I knew what to do for the whole project besides the actual programming of the game and that I thought that the Gantt chart was off point but it was surprisingly enough. There were quite a few changes due to the feasibility not having technical, economical and legal and schedule in the project. I haven't done a two player mode as it would be too difficult to code and I have no idea how I would do a two player mode for Sudoku.

Requirement Specification

The purpose of this program is to recreate a Sudoku-like game which involves placing numbers in rows and columns without having the same number being the same row or column of each other which also has a time system to show how well the user is doing and how quick they are completing the game and to show how well the user is compared to other people that played the game and will also have difficulty modes so that people from all different age and skill ratings can play the game. It will also need to both save and read the leader boards so that if the user quits the game that the leader boards that were updated during that session will still be there. The scope of this project is that it is only 4x4 and each time the game is started it randomises the numbers in the grid depending on the difficulties and should have a leader board which shows your results and compares them to others and a timer that can show people how fast they completed the game and looking at the project the only boundaries that I might encounter is the difficulty settings taking more time than it should and because of my lack of knowledge on data grids it might take me longer to create the grid.

The game will be 4x4 and the numbers will only be 1 to 4 as doing it any bigger would take longer and would be harder to develop which the game might not be finished thus near the end of development. The user should then be able to click on a difficulty button being either easy, medium or hard and should show the random right amount of number for each button – easy having most of the number done already, medium have less number involved and hard being no number showing. The leader boards should read the file for the difficulty selected that contains all the shortest times users have achieved.

The timer should also start and the user should be able to enter numbers from 1 to 4 into the different squares and will be restricted if the put a same number that is in the same row or column and if the user enters all of the numbers correctly the game will finish with the users end time and will allow the user to enter a name for the leader boards which will show up after they fully finish the game and will reset the game. It will then save the updated leader boards. There will also be a reset button to restart the full program so that everything goes back to a default state.

UPDATE –

I have changed the program so that it only has one difficulty as it was difficult to implement multiple difficulties so the multiple leader boards have been removed to only have one leader board but that also means less time working on multiple difficulties and more time on the grid and leader board. The game has changed due to the implementation of a complete button instead of it doing it asynchronies with the grid.

Personas

<p>Dominick Matt Profile: Video Game Advocate Gender: Male Age: 21 Location: Edinburgh Occupation: Student</p>	
<p>Motivations</p> <p>Likes to play challenging video games but hardly finds any</p> <p>Currently enjoys playing puzzle games that challenge his mind</p> <p>They are very competitive at playing video games.</p>	<p>Goals</p> <p>Wants to play a challenging game he can play over and over again</p> <p>Wants to show off how great he is at the game</p> <p>Wants the game to be a puzzle game</p>
	<p>Frustrations</p> <p>Doesn't want it to be too easy.</p> <p>Not being able to challenge his mind and speed</p>

Jordon Ryder
Profile: Senior IT Manager
Gender: Male
Age: 58
Location: Glasgow
Occupation: IT Manager

Motivations

Likes to have a chill time off work playing some video games.

Wants something that isn't too challenging

Likes puzzles games that relate to numbers more than any video game.

Goals

Needs to have a difficulty setting for him so that he can casually have a game of Sudoku

Game needs to be some kind of numbers game.

Frustrations

The game might be too hard for Jordan

The game isn't Sudoku or anything related to numbers

User Stories

Dominick Matt

Video game advocate Dominick Matt has a busy time finding a puzzle game that can truly test his mental skills properly and is also very competitive. He would like a puzzle game that is challenging every time he plays it and can show off his fastest time he's beaten the game.

Jordan Ryder

Senior IT manager Jordon Ryder likes to come home and chill playing puzzle games that relate to numbers that isn't too challenging. He would like some kind of difficulty setting that targets his skill level.

User Scenarios

Dominick Matt

Dominick is a hard-core puzzle video game player that through previous experiences hard puzzle games he has played has not challenged his mind in anyway and is disappointed that they can't fit his need. He's tried many puzzle games but it won't fit his need and usually don't have scoreboard to show how good he is. He is usually bored and feels he hasn't gotten anything to chill to for the next month or two until another game comes out.

Jordan Ryder

Jordan comes back from work to usually play a simple puzzle game that involves numbers but usually the game is too hard for Jordan and gets frustrated over how he can't complete it and usually the puzzle game is set at one difficulty while he can't play it at a difficulty level that suits his skill level.

Use Case

Dominick Matt

1	Opens the Sudoku game
2	Chooses the difficulty setting of hard
3	Timer Starts
4	The grid shows no numbers
5	He inputs numbers for each box in the rows and columns
6	If he gets one answer wrong the number he inputted will go red
7	He will re-input the answer
8	If all numbers are in the correct position then the game will stop the timer and congratulate the player
9	The scoreboard for the hard difficulty will show up showing his final time in one of the positions
10	The game will return to the title screen and waits until the game is started again or is quit out

Jordan Ryder

1	Opens the Sudoku game
2	Chooses the difficulty setting of easy or medium
3	Timer starts
4	The grid shows quarter of the numbers or half of the number in the correct position
5	If he gets one answer wrong then the number he inputted will go red
6	He will re-input the answer
8	If all numbers are in the correct position then the game will stop the timer and congratulate the player
8	The scoreboard for either the easy or medium difficulty will show up showing his final time in one of the positions
9	The game will return to the title screen and waits until the game is started again or is quit out

Update –

As there is no more than one difficulty setting the users could possibly be turned off from the game as there might be confusion for example Dominick might expect difficulties in my game and will get it and realise that there is only one difficulty.

Reflective Commentary

Looking back, I discovered that programming different difficulties took too much time and I didn't know what I was doing so I just removed to get the project moving along. Besides that, the requirement specification was on point and did do most of what I said. The problems I had was the use case didn't follow my implementation as there is only one difficulty so people might be turned off at the game.

Test Plan

Parts of the program that would be tested are the grid, leader board, difficulty settings. The grid will be tested first as it would need to check if its only one number and the number is between or equal to 1 to 4 and will check if the number is a unique number that isn't the same in the included row or column. The leader board should have at least a three letter initial as the name and should go up in place if the time is better than a time that is on the leader board. The difficulty settings should randomly include numbers that are in different places in the grid for the different difficulties and two numbers should not be included in the same box in the grid.

Test Data	Normal	Extreme	Exceptional
Grid	3, 2,	1, 4	AAAYYYYY, 21
Leader board	AFC, GDR, FAW,	A, ZZZ	2324, ;''''l,, Banana

The test results will be recorded on a separate word document with expected results and the actual results side by side and the conclusion, if it works or if it doesn't work and how to improve for the future. The users will get a working build to test some of the features like the difficulty settings and see if they can get unexpected results like for example they are able to enter letters and numbers not just numbers alone and ask questions about the current game to receive further feedback on what works in the program and what needs to be changed for the better. The leaderboards should overwrite the save data that is given to it for all three difficulties. It should sort the player and it's time to be in the leaderboard like for example:

Leaderboard

LRG	06.23
CTE	04.32
CJE	12.23
SG	32.23
WOO	12.43
PEE	18.21
SOO	19.40
SAY	43.10
MAY	31.10
JOE	05.11

This will sort to:

CTE	04.32
JOE	05.11
LRG	06.23
CJE	12.23
WOO	12.43
PEE	18.21
SOO	19.40
MAY	31.10
SG	32.23
SAY	43.10

UPDATE –

I have changed the program so that it only has one difficulty as it was difficult to implement multiple difficulties meaning there will be only one leader board. There is now more buttons for A1-D4 of the grid because the program won't allow me to enter number by just clicking on them but the validation still is used for it. I also got people to test my game to see if there are any problems with my game at all. I have also created an end-user survey to get people to playtest and give feedback on the game. Here is the survey:

User survey for game:

Name:

Did you like playing my game?

What is your favourite part of this game?

What do you think that could be improved in the program?

What features could be added further into the game?

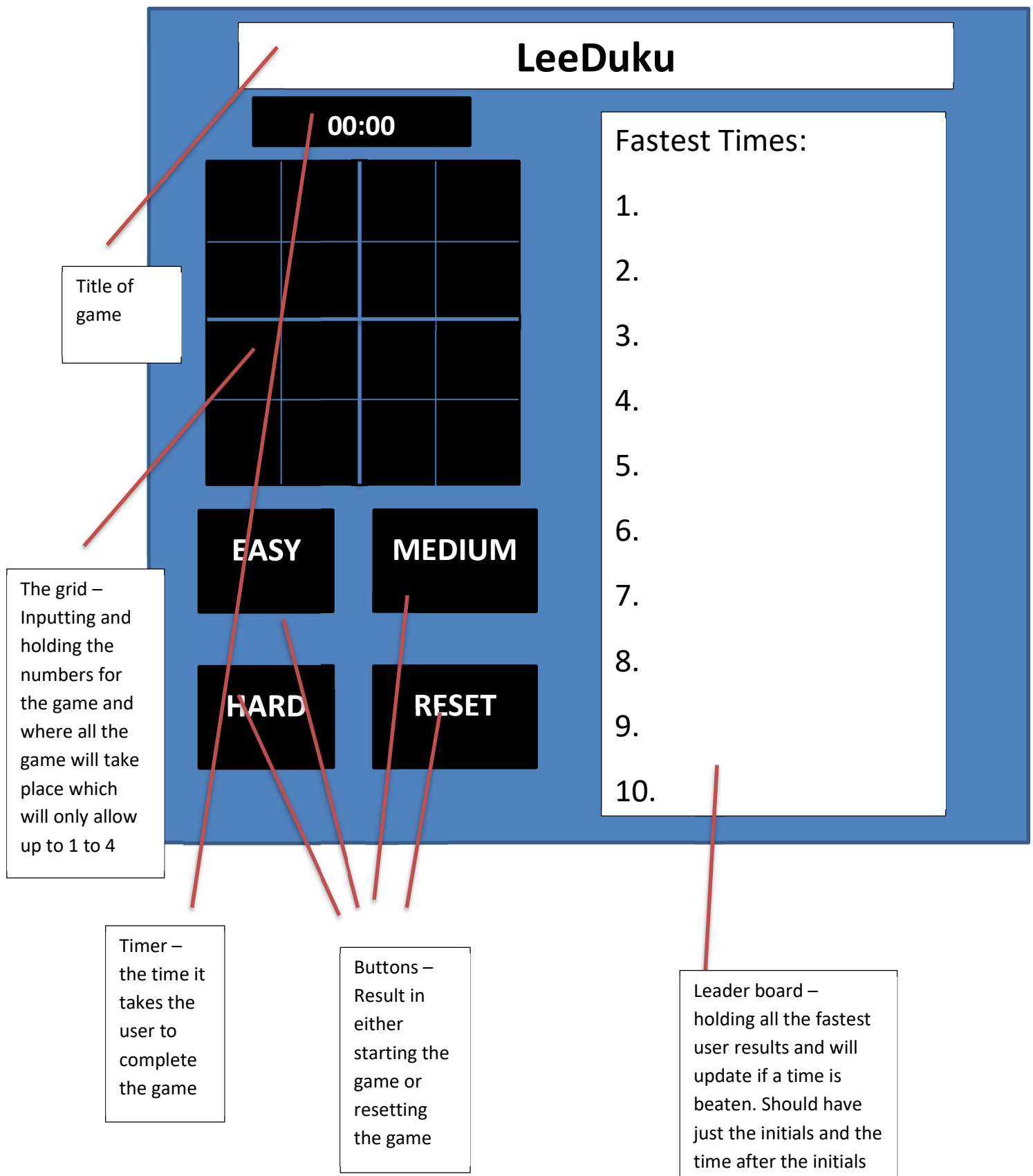
--

Did you have any problems playing this game at all?

Reflective Commentary

I did have an idea of what I was doing to test it as I knew there would be input validation and a bubble sort just for the leader board. The only thing that needed to be changed was there aren't multiple difficulties anymore and only just one meaning there wouldn't be testing if the right amount of numbers comes out. I also moved the end-user survey into the test plan as I originally had it in the final testing by accident.

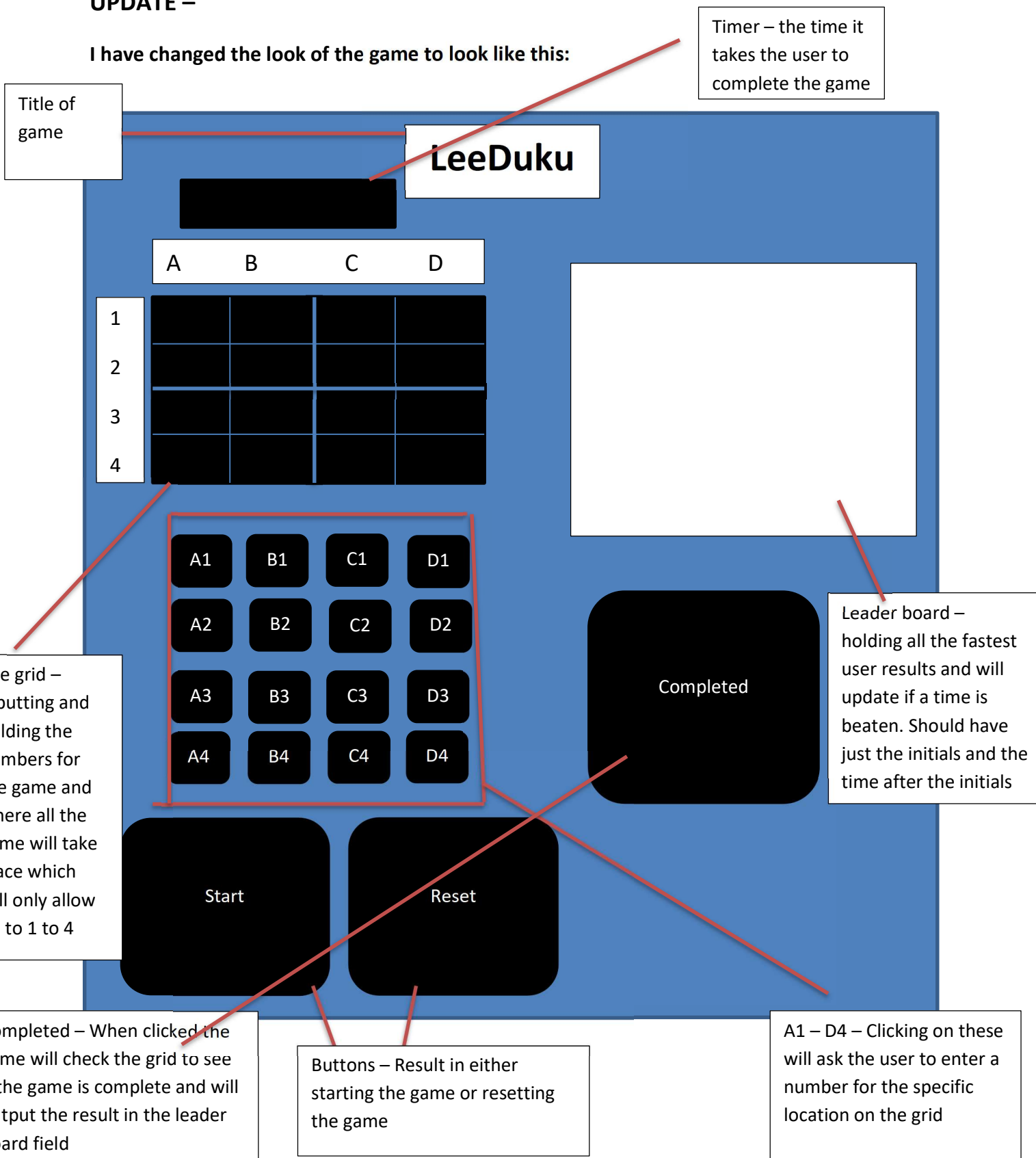
Interface Design



I have created a wireframe to show what the program could look like and where everything will be and to explain what will do what in the end of developing the program.

UPDATE –

I have changed the look of the game to look like this:



After the user complete the game something like this should pop up:

Please enter your initials please:

This pops up when the user has finished the game which asks them for their initials which can't be longer than 3 letters or numbers and if the user enters more than 3 then it should pop up an error message asking the user to enter only three characters.

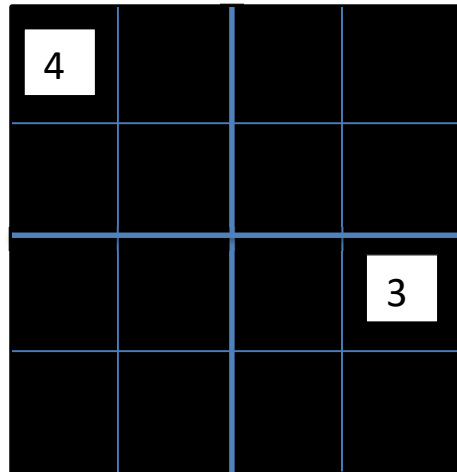
Please only enter three characters:

If the player presses one of the difficulties the grid should result in adding numbers randomly dependent on the difficulty for example:

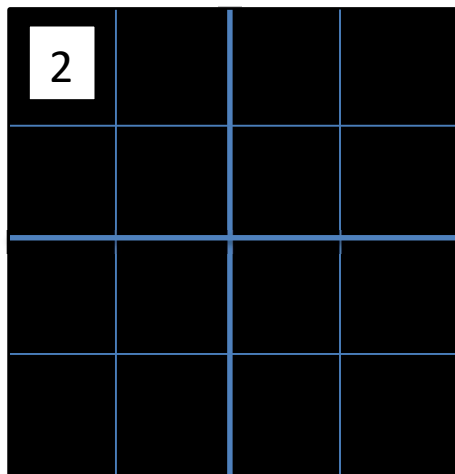
Easy –

1			2
		3	
	4		

Medium –



Hard (One Numbers) –



UPDATE –

As the program has changed to only one difficulty it should be something like the easy grid that should be outputted to the grid. I forgot to talk about the problems people could potentially have like the font size might be small so it should be changed for people who have problems with their eye sight. People with epilepsy can play this game because it has no flashing images of any kind, people with colour blindness can play the game fine as the game is mainly grey, black and white, people with deafness or problems with hearing can play this game because there is no sound in the game, people with impaired motor control can play this as it is mainly using a keyboard.

Reset should just remove all the numbers that are currently in the grid and reset the timer back to 0.00

As all the game will be on one screen there won't be any change in screen layout.

Reflective Commentary –

The interface design has changed drastically as I didn't expect it to not allow me to enter number at all so what I done was used buttons to enter the numbers into the grid and now there is only one difficulty there is only the start button and because of that the grid looks more like the easy mode version of the difficulty design but everything else stayed the same. I had to change the entire design of my interface due to the data grid not allowing me to enter any numbers by just clicking on the grid so I had to make new buttons to relate to the grid and cause I have removed the difficulties and only have one now I have now only one button, It also needs a complete button as I found out live code isn't asynchronies with checking the grid so I just designed around it by allowing the user to say that they've completed it at anytime

Program/Data Structure Design

Button - Easy:

1. Initialise variables that have been made for the button only.
 - 1.1 Reset the timer
 - 1.2 Put Easy into variable Difficulty
2. 4 numbers will be inserted using a fixed loop in random locations inside the 4x4 grid that is smaller than or equal to 4 and bigger than or equal to 1
 - 2.1 repeat using a loop 4 times
 - 2.2 put a random number into the field TheGrid[RandomLocation][RandomLocation]
 - 2.3 end loop
3. The timer will start to count up
 - 3.1 Put 0 into the field called timer
 - 3.2 Put true into the variable called UpTime
 - 3.3 Send "updateTimer" to card "Leedoku"
4. Check to see if the completed variable has been set to true
 - 4.1 If it is then answer with "Congratulations you finished the puzzle!
Please enter your initials please:"
 - 4.2 repeat until the length of the initials is smaller or equal to 3
 - 4.3 If it isn't then answer with "Please only enter three characters:"
 - 4.4 re-ask for their initials again

- 4.5 end loop
- 5. Send the information over to the easy difficulty leaderboard

Button - Normal:

- 1. Initialise variables that have been made for the button only.
 - 1.1 Reset the timer
 - 1.2 Put Normal into variable Difficulty
- 2. 2 numbers will be inserted using a fixed loop in random locations inside the 4x4 grid that is smaller than or equal to 4 and bigger than or equal to 1
 - 2.1 repeat using a loop 2 times
 - 2.2 put a random number into the field TheGrid[RandomLocation][RandomLocation]
 - 2.3 change the random number being inserted into the grid
 - 2.4 end loop
- 3. The timer will start to count up
 - 3.1 Put 0 into the field called timer
 - 3.2 Put true into the variable called UpTime
 - 3.3 Send "updateTimer" to card "Leedoku"
- 4. Check to see if the completed variable has been set to true
 - 4.1 If it is then answer with "Congratulations you finished the puzzle!
Please enter your initials please:"
 - 4.2 repeat until the length of the initials is smaller or equal to 3
 - 4.3 If it isn't then answer with "Please only enter three characters:"
 - 4.4 re-ask for their initials again
 - 4.5 end loop
- 5. Send the information over to the normal difficulty leaderboard

Button - Hard:

- 1. Initialise variables that have been made for the button only.
 - 1.1 Reset the timer
 - 1.2 Put Hard into variable Difficulty
- 2. The timer will start to count up
 - 2.1 Put 0 into the field called timer
 - 2.2 Put true into the variable called UpTime
 - 2.3 Send "updateTimer" to card "Leedoku"
- 3. Check to see if the completed variable has been set to true
 - 4.1 If it is then answer with "Congratulations you finished the puzzle!
Please enter your initials please:"
 - 4.2 Put the users initials into a variable called Initials
 - 4.3 repeat until the length of the initials is smaller or equal to 3

- 4.4 If it isn't then answer with "Please only enter three characters:"
- 4.5 re-ask for their initials again
- 4.6 end loop
- 5. Send the information over to the hard difficulty leaderboard

Squared Field - TheGrid:

- 1. The numbers from the selected difficulty setting is placed
- 2. If the game is completely done and correct
 - 2.1 Set the completed variable to true

Field - Leaderboards:

- 1. Load up a file related to the difficulty in the variable Difficulty containing the times for players who completed the game (should be black if starting up game for the first time)
 - 1.1 Create local variables called LeaderInitials and LeaderTime
 - 1.2 Check what difficulty it was from the variable called Difficulty
 - 1.3 Change the name of the leaderboard to be related to the difficulty
 - 1.4 Ask the user where the file leaderboard that is related to the difficulty is currently located
 - 1.5 Put the url into a variable called LeadLocation
 - 1.6 Put the information into the variable LeadLocation
- 2. Receive the time and initials from either of the difficulties and sort them using bubble sort
 - 2.1 Make the local variables swapped and SizeOfLeader
 - 2.2 Split the LeadLocation data by comma and put the information in the two variables LeaderInitials and LeaderTime
 - 2.3 Put the amount of players from the variable LeaderInitials into SizeOfLeader
 - 2.4 Initialise it to be true
 - 2.5 Repeat it until swapped is equal to true
 - 2.6 Set the swapped variable to be false
 - 2.7 repeat with loop from 1 to the number from SizeOfLeader
 - 2.8 if the variable LeaderTime [loop-1] is bigger than LeaderTime [loop]
 - 2.9 It will swap the data inside the variable LeaderTime
 - 2.10 It will also swap the data inside LeaderInitials
 - 2.11 Set the variable swapped to true
 - 2.12 End if
 - 2.13 End repeat
 - 2.14 End repeat

3. Display the results
 - 3.1 Create a variable called Counter
 - 3.2 Initialise the variable Counter to have the value of 1
 - 3.2 Put the text "Initials" then tab and then "Time" into the first line of field Leaderboard
 - 3.3 Repeat with loop from 1 to 10
 - 3.4 add 1 to the variable Counter
 - 3.2 put the variable LeaderInitials then tab and then LeaderTime into line Counter + 1 of the field Leaderboard
4. Save and overwrite the file that was loaded into the program.
 - 4.1 local LeaderboardContents
 - 4.2 put the field Leaderboard into variable LeaderboardContents
 - 4.3 put the data in variable LeaderboardContents into the file location that is currently in LeadLocation
 - 4.4 Answer with "Leaderboard saved"
5. Reset the Grid

UPDATE –

During the implementation process due to not having any more difficulty settings and the added fact of a completed button I have changed the code significantly which is shown here:

Button – Start – Starts the game which puts random numbers into the grid

In/out - completed, griddata, gridoutput, timer

In - row, column

1. Declare variables
 - 1.1 Create variables timer, griddata, completed, row, column, gridoutput
2. Initialise the variables
 - 2.1 Reset the timer
 - 2.2 Put false into variable completed
 - 2.3 Empty variable timer
 - 2.4 Empty variable griddata
 - 2.5 Empty variable gridoutput
 - 2.6 Empty the grid
 - 2.7 Empty the field leaderboard
3. Initialise the grid
 - 3.1 Create local variables firstnum, secondnum, thirdnum, forthnum, truth1, truth2, truth3, truth4
 - 3.2 Put random numbers in four local variables

- 3.3 Repeat for every row and column
- 3.4 Put a random number into the variable griddata
- 3.5 End repeat
- 3.6 Repeat until the random number does not equal a number in the same row or column
- 3.7 If the number that is being inserted is the same number in the same row or column
- 3.8 Randomise the number again
- 3.9 Else put the number into the variable gridoutput and tell the program that the number is unique
- 3.10 End if
- 3.11 End repeat
- 3.12 Set the grid to the variable gridoutput

Button – A1 to D4 – Puts a number the user has entered into the defined area of the grid

In/out - gridoutput

In - griddata

I will use A1 as an example. All buttons that go up to D4 do the exact same just for their specified locations.

1. Declare what global variables are being used
 - 1.1 declare the variables griddata and gridoutput
2. Insert the number in A1
 - 2.1 create local variable A1grid
 - 2.2 Ask the user what number they are inserting and input it into the variable A1grid
 - 2.3 If the user has entered C then the data in A1 will be cleared
 - 2.4 Exit back to the menu
 - 2.5 end if
 - 2.6 repeat until the number inside A1grid is bigger or equal to 1 and less than or = 4
 - 2.7 tell the user that the data is invalid and get them to enter a number that is valid
 - 2.8 put it into the variable A1grid
 - 2.9 end repeat
 - Put A1grid into griddata
 - 2.10 Repeat until the number being inserted from A1grid is not in the same row or column
 - 2.11 Set C1 to true as it is the column is going in
 - 2.12 Set R1 to true as it is the same column is going in

- 2.13 If the number in griddata is equal to a number in the row or column
- 2.14 tell the user to change the number to something that isn't in the same row or column
- 2.15 put it into A1grid
- 2.16 put the number of A1grid into griddata
- 2.17 end if
- 2.18 end repeat
- 2.19 put the number of griddata into variable gridoutput
- 2.20 set the grid to the gridoutput

Stack – Leeduku – Is where timer counts up and the leaderboard is saved and loaded

In/out - locationoflead (for opening the leaderboard), actualleaderboard

In - locationoflead (for saving the leaderboard)

1. Declare what global variables are being used
 - 1.1 declare the global variables timer, completed, locationoflead and actualleaderboard
2. Timer Start
 - 2.1 add 1 to the field Timer
 - 2.2 if completed is equal to true or is equal to reset then
 - 2.3 put the field timer into the variable timer
 - 2.4 else show the number through the field timer
 - 2.5 end if
3. Saving the leaderboard
 - 3.1 Create local variable leaderboardcontent
 - 3.2 put the field leaderboard into the variable leaderboardcontent
 - 3.3 save the contents of leaderboardcontent into Documents
4. Open the leaderboard
 - 4.1 open the leaderboard that is located from Documents and put it into the variable actualleaderboard

Button – Completed – Checks if the game is completed and will load and save the leaderboard results

In - gridoutput, timer, actualleaderboard initials, initials, counter, actualleaderboard, initials, timer, Name_, TheTime

1. Declare what global variables are being used
 - 1.1 declare the global variables completed, gridoutput, timer
 - 1.2 create new global variables that are called locationoflead, alreadyafile, actualleaderboard, initials, counter, Name_, TheTime

2. Get the leaderboard
 - 2.1 put the url of Documents into the variable locationoflead
 - 2.2 if there is a leaderboard in Documents then
 - 2.3 Put Y into AlreadyAFile
 - 2.4 Else put N into AlreadyAFile
3. Check to see if game is finished
 - 3.1 repeat until the game has checked the grid is filled up
 - 3.2 if a part of a row or column is empty then
 - 3.3 answer that they haven't completed it
 - 3.4 exit to top
 - 3.5 else tell the program that it isn't empty
 - 3.6 end repeat
 - 3.7 put true into Completed
4. Get the time into minutes
 - 4.1 divide the number in Timer by 60
 - 4.2 Round the number inside Timer by 2
5. Get the Initials of the user
 - 5.1 ask what is their first three initials
 - 5.2 put it into the variable initials
 - 5.3 get the length of the initial and put it into the variable lengthofinitials
 - 5.4 repeat until the initials is bigger than or equal to 1 and less than or equal to 3
 - 5.5 tell the user that they need to enter up to three letters and put it into initials
 - 5.6 put the length of the initials into lengthofinitials
 - 5.7 end repeat
6. Output the variables
 - 6.1 put 1 into the variable counter
 - 6.2 if AlreadyAFile is equal to Y then
 - 6.3 repeat with loop equals 1 to the number of lines in the variable
actualleaderboard
 - 6.4 put everything from actualleaderboard into the variable lead
 - 6.5 split lead by spaces
 - 6.6 put the first part of lead into the variable Name_
 - 6.7 put the second part of lead into the variable TheTime
 - 6.8 add 1 to counter
 - 6.9 if the loop is equal to the line of actualleaderboard then
 - 6.10 put the data of initials into Name_ after everything from actualleaderboard
 - 6.11 put the data of timer into TheTime after everything from actualleaderboard
 - 6.12 Call the Bubble_Sort
 - 6.13 End if
 - 6.14 End repeat
 - 6.15 Repeat with a loop that is equal to 1 to the number in counter

- 6.16 Put the contents of Name_ space TheTime into the field leaderboard
- 6.17 End repeat
- 6.18 End if
- 6.19 If variable AlreadyAFile = N then
- 6.20 Put the variable initials space timer into line one of field leaderboard
- 6.21 End if
- 7. Sorting the leaderboard
 - 7.1 put true into the variable swapped
 - 7.2 repeat while swapped
 - 7.3 put false into the variable swapped
 - 7.4 repeat with a loop = 1 to Counter
 - 7.5 if the first time is bigger than the second time then
 - 7.6 swap the data in the variable TheTime
 - 7.7 swap the data in the variable Name_
 - 7.8 put true into swapped
 - 7.9 end if
 - 7.10 end repeat
 - 7.11 end repeat
- 8. The actual swap process
 - 8.1 put the data in variable item 1 into temp
 - 8.2 put the data in variable item2 into variable item1
 - 8.3 put temp into variable item2

Button – reset – This will reset the program

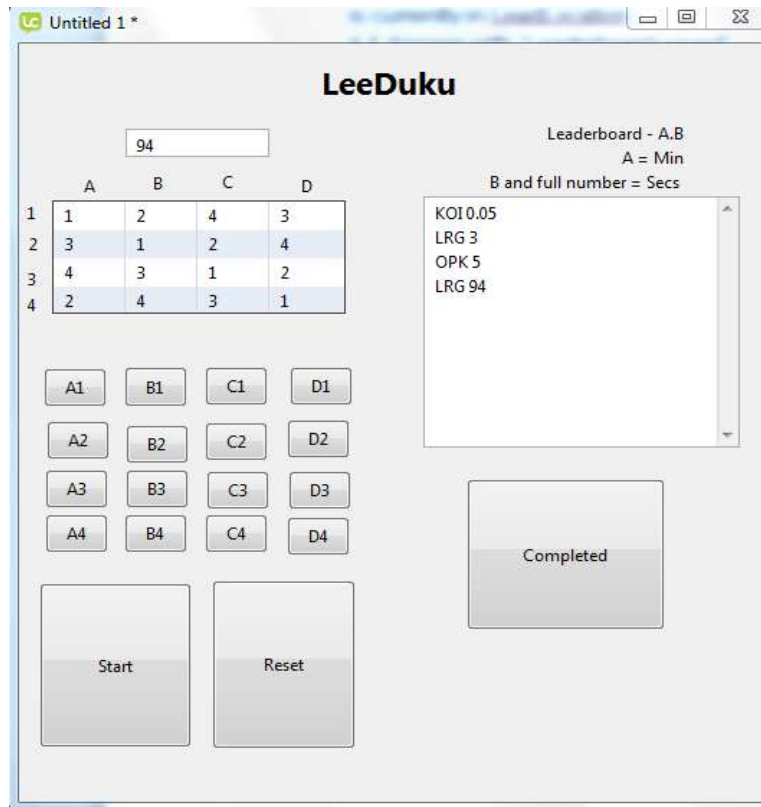
- 1. Initialise the game
 - 1.1 set the data inside the grid to empty
 - 1.2 empty the field leaderboard
 - 1.3 empty the field timer
 - 1.4 put reset into completed

Reflective Commentary –

The pseudocode wasn't far off but It needed to be changed to suit the changes I done for the implementation but it wasn't a lot so like the difficulties only change to suit the one standard difficulty and getting the location for documents and stuff like that so I don't see it being that different from what it is now. I found out live code isn't asynchronies with checking the grid so I just worked around it by allowing the user to say that they've completed it at any time by clicking on a button.

Implementation

Game home:



Start button:

```
//Create global variables to be used during the program
global Timer, GridData, Completed, Row, Column, GridOutput

on mouseUp
  InitialiseVariable Completed, Timer, GridData, GridOutput
  InitialiseGrid Row, Column, GridData, GridOutput
  countdown Timer
end mouseUp

//Initialise the variables and clear out the data grid and the leaderboard
on InitialiseVariable
  put 0 into field "Timer"
  put false into Completed
  put 0 into Timer
  put 0 into GridData
  put empty into GridOutput
  set the dgdata of group "TheGrid" to empty
  put "" into field "Leaderboard"
end InitialiseVariable

//This will create the 2Darray that will be used to create the data grid
on InitialiseGrid Row, Column, @GridData, @GridOutput
  local FirstNum, SecondNum, ThirdNum, ForthNum, Truth1, Truth2, Truth3, Truth4
  //This is putting random numbers into 4 different variables
  put Random(4) into FirstNum
  put Random(4) into SecondNum
  put Random(4) into ThirdNum
  put Random(4) into ForthNum
  //This will randomly place 4 numbers from 1 to 4 into the rows and columns
  repeat with row = 1 to 4
    repeat with Column = 1 to 4
      put random(4) into GridData[Row, Column]
    end repeat
  end repeat

  //This is validation for the random numbers to check if there isn't the same number in the row or column its currently being placed in
  repeat until Truth1 = True and Truth2 = True and Truth3 = True and Truth4 = True
    if GridData[Row, 1] = GridOutput[SecondNum][Col 2] or GridData[Row, 1] = GridOutput[ThirdNum][Col 3] or GridData[Row, 1] = GridOutput[ForthNum][Col 4] then
      put random(4) into GridData[Row, 2]
    else
      put GridData[Row, 1] into GridOutput[FirstNum][Col 1]
      put True into Truth1
    end if
    if GridData[Row, 2] = GridOutput[FirstNum][Col 1] or GridData[Row, 2] = GridOutput[ThirdNum][Col 3] or GridData[Row, 2] = GridOutput[ForthNum][Col 4] then
      put random(4) into GridData[Row, 2]
    else
      put GridData[Row, 2] into GridOutput[SecondNum][Col 2]
      put True into Truth2
    end if
    if GridData[Row, 3] = GridOutput[FirstNum][Col 1] or GridData[Row, 3] = GridOutput[SecondNum][Col 2] or GridData[Row, 3] = GridOutput[ForthNum][Col 4] then
      put Random(4) into GridData[Row, 3]
    else
      put GridData[Row, 3] into GridOutput[ThirdNum][Col 3]
      put True into Truth3
    end if
    if GridData[Row, 4] = GridOutput[FirstNum][Col 1] or GridData[Row, 4] = GridOutput[SecondNum][Col 2] or GridData[Row, 4] = GridOutput[ThirdNum][Col 3] then
      put Random(4) into GridData[Row, 4]
    else
      put GridData[Row, 4] into GridOutput[ForthNum][Col 4]
      put True into Truth4
    end if
    if the result = "cancel" then exit to top
  end repeat
  //Will put the numbers into the data grid
  set the dgdata of group "TheGrid" to GridOutput
end InitialiseGrid
```

A1 to D4 Buttons:

//Getting the variables from the start button

global Timer, GridData, Complete, GridOutput, Column, Row

on mouseUp

 InsertNumberA1 GridData, GridOutput

end mouseUp

//This asks the user to input a number from 1 to 4 to be placed in its right row and column

on InsertNumberA1 GridData, @GridOutput

local A1Grid,

put empty into A1Grid

ask "What is the number being put inside the grid? Insert C for Clear"

put it into A1Grid

if A1Grid = "C" **then**

put empty into GridOutput[1]["Col 1"]

exit to top

end if

// This input validation checks if its less or equal to 4 and bigger or equal to 1

repeat **until** A1Grid >=1 **and** A1Grid <=4

ask "Invalid data. Please insert a number between 1 and 4"

put it into A1Grid

if the result = "cancel" **then** **exit to top**

end repeat

put A1Grid into GridData[1, 1]


```

//Checks to see if there is a same number inside the row or column the number is being placed. C=Column, R=Row
Repeat until C1 = True and C2 = True and C3 = True and C4 = True and R1 = True and R2 = True and R3 = True and R4 = True
  put True into C1
  put True into R1
  if GridData[1,1] = GridOutput[1]["Col 2"] then
    ask "Invalid data. Please insert a number that isn't in the same row or column!" put it into A1Grid
    put A1Grid into GridData[1,1]
  else put True into C2
  if GridData[1,1] = GridOutput[1]["Col 3"] then
    ask "Invalid data. Please insert a number that isn't in the same row or column!" put it into A1Grid
    put A1Grid into GridData[1,1]
  else put True into C3
  if GridData[1,1] = GridOutput[1]["Col 4"] then
    ask "Invalid data. Please insert a number that isn't in the same row or column!" put it into A1Grid
    put A1Grid into GridData[1,1]
  else put True into C4
  if GridData[1,1] = GridOutput[2]["Col 1"] then
    ask "Invalid data. Please insert a number that isn't in the same row or column!" put it into A1Grid
    put A1Grid into GridData[1,1]
  else put True into R2
  if GridData[1,1] = GridOutput[3]["Col 1"] then
    ask "Invalid data. Please insert a number that isn't in the same row or column!" put it into A1Grid
    put A1Grid into GridData[1,1]
  else put True into R3
  if GridData[1,1] = GridOutput[4]["Col 1"] then
    ask "Invalid data. Please insert a number that isn't in the same row or column!" put it into A1Grid
    put A1Grid into GridData[1,1]
  else put True into R4
  if the result = "Cancel" then exit to top
end Repeat
//Puts the number in the right column and row in the data grid
put GridData[1, 1] into GridOutput[1]["Col 1"]
set the dgdata of group "TheGrid" to GridOutput
end InsertNumberA1

```

This is the same for A2, A3 etc. Only things that changed around are the rows and the columns and the A, B, C, D.

Completed button:

```

//Gets the global variables from the buttons previous and creates new global variables
Global Completed, GridOutput, LocationOfLead, AlreadyAFile, ActualLeaderboard, Initials, Timer, Counter, Name_, TheTime

on mouseUp
  Finished GridOutput
  TimerCalculation Timer
  //Gets the url for the location of Documents so that it gets saved there
  put specialFolderPath("Documents") & "/Leaderboard.txt" into LocationOfLead
  //Checks to see if there is already a leaderboard document
  if there is a file LocationOfLead then
    OpenING LocationOfLead, ActualLeaderboard
    put "Y" into AlreadyAFile
  else
    put "N" into AlreadyAFile
  end if
  InitialsOfUser Initials
  Output Counter, ActualLeaderboard, Initials, Timer, Name_, TheTime
  SaveING LocationOfLead
end mouseUp

```


//Checks to see if the grid is full and complete otherwise will tell the user that they didn't complete it. Variables are I(Number) as it is more quicker to type and create.

on Finished GridOutput

local I1, I2, I3, I4, I5, I6, I7, I8, I9, I10, I11, I12, I13, I14, I15, I16

repeat until I1 = True and I2 = True and I3 = True and I4 = True and I5 = True and I6 = True and I7 = True and I8 = True and I9 = True and I10 = True and I11 = True and I12 = True and I13 = True and I14 = True and I15 = True and I16 = True

if GridOutput[1]["Col 1"] is empty then

answer "You didn't complete it"

exit to top

else put True into I1

if GridOutput[2]["Col 1"] is empty then

answer "You didn't complete it"

exit to top

else put True into I2

if GridOutput[3]["Col 1"] is empty then

answer "You didn't complete it"

exit to top

else put True into I3

if GridOutput[4]["Col 1"] is empty then

answer "You didn't complete it"

exit to top

else put True into I4

if GridOutput[1]["Col 2"] is empty then

answer "You didn't complete it"

exit to top

else put True into I5

if GridOutput[2]["Col 2"] is empty then

answer "You didn't complete it"

exit to top

else put True into I6

if GridOutput[3]["Col 2"] is empty then

answer "You didn't complete it"

exit to top

else put True into I7

```

if GridOutput[4]["Col 2"] is empty then
    answer "You didn't complete it"
    exit to top
else put True into I8
if GridOutput[1]["Col 3"] is empty then
    answer "You didn't complete it"
    exit to top
else put True into I9
if GridOutput[2]["Col 3"] is empty then
    answer "You didn't complete it"
    exit to top
else put True into I10
if GridOutput[3]["Col 3"] is empty then
    answer "You didn't complete it"
    exit to top
else put True into I11
if GridOutput[4]["Col 3"] is empty then
    answer "You didn't complete it"
    exit to top
else put True into I12
if GridOutput[1]["Col 4"] is empty then
    answer "You didn't complete it"
    exit to top
else put True into I13
if GridOutput[2]["Col 4"] is empty then
    answer "You didn't complete it"
    exit to top
else put True into I14
if GridOutput[3]["Col 4"] is empty then
    answer "You didn't complete it"
    exit to top
else put True into I15
if GridOutput[4]["Col 4"] is empty then
    answer "You didn't complete it"
    exit to top
else put True into I16
end repeat
put True into Completed
end Finished

```

```

//Gets the users initials for the leaderboard
on InitialsOfUser @Initials
    local LengthOfInitials
    ask "What is your first three initials?" put it into Initials
    if the result = "cancel" then exit to top
    put the length of Initials into LengthOfInitials
    //Input validation for the initials so that it can only be at least 3 letters long
    repeat until LengthOfInitials >=1 and LengthOfInitials <=3
        ask "Please enter three characters please!" put it into Initials
        put the length of Initials into LengthOfInitials
        if the result = "Cancel" then exit to top
    end repeat
end InitialsOfUser

```

```

//Creates the leaderboard
on Output Counter, ActualLeaderboard, Initials, Timer, Name_, TheTime
  local Lead,
  put 1 into Counter
  //Checks to see if there is a leaderboard document already in place
  if AlreadyAFile = "Y" then
    //Gets the data from the loaded leaderboard document and then puts them into variables
    Repeat with loop = 1 to the number of lines of ActualLeaderboard
      put line loop of ActualLeaderboard into Lead
      split Lead by space
      put Lead[1] into Name_[loop]
      put Lead[2] into TheTime[loop]
      add 1 to Counter
      //Puts the current users values
      if loop = number of lines of ActualLeaderboard then
        put Initials into Name_[loop+1]
        put Timer into TheTime[loop+1]
        Bubble_Sort Counter, TheTime, Name_
      end if
    end Repeat
    //puts the variables into the leaderboard
    repeat with loop = 1 to Counter
      put Name_[loop] && TheTime[loop] into line loop of field Leaderboard
    end repeat
  end if
  //If no document is found then it puts the current users values into the leaderboard
  if AlreadyAFile = "N" then
    put Initials && Timer into line 1 of field "Leaderboard"
  end if
end Output

```

```

On Bubble_Sort @TheTime, @Name_
  local swapped
  put true into swapped
  repeat while swapped
    put false into swapped
    //Will check to see if the current time selected is bigger than the next time and if it is will swap the two times and the initials.
    repeat with count = 1 to Counter
      if TheTime[count-1] > TheTime[count] then
        swap TheTime[count-1], TheTime[count]
        swap Name_[count-1], Name_[count]
        put true into swapped
      end if
    end repeat
  end repeat
end Bubble_Sort

//This is the actual swap process.
on Swap @item1, @item2
  put item1 into temp
  put item2 into item1
  put temp into item2
end Swap

```

Leeduku stack:

```
global Timer, Completed, LocationOfLead, ActualLeaderboard

//The acutal timer that will count up until the game is completed or its been reset
on countdown
  add 1 to field "Timer"
  if Completed = True or Completed = "Reset" then
    put field "Timer" into Timer
  else
    send countdown to me in 1 sec
  end if
end countdown

//Saves the current leaderboard into the document loaded or creates a new document if there is none
on SaveING LocationOfLead
  local LeaderboardContent
  put field "Leaderboard" into LeaderboardContent
  put LeaderboardContent into url ("file:" & LocationOfLead)
end SaveING

//Opens up the leaderboard document.
on OpenING @LocationOfLead, @ActualLeaderboard
  put url ("file:" & LocationOfLead) into ActualLeaderboard
end OpenING
```

Reset Button:

```
global Completed

on mouseUp
  //Initalises the grid and puts reset into the variable completed
  set the dgdata of group "TheGrid" to empty
  put "" into field "Leaderboard"
  put "" into field "Timer"
  put "Reset" into Completed
end mouseUp
```

Reflective Testing –

The implementation was the worst thing I have done. The main problem was that the data grid didn't work half the time meaning I get frustrated over some weird thing the data grid wants to do primarily the whole grid moving down one. The code could've been much, much tidier and simpler rather than writing a lot of code because I had problems looking for any errors in that code. This whole part of the project changed a lot towards everything I have done for the project like the project plan, requirement specification, test plan etc.

Final Testing

User input numbers:

Test Data	Expected	Results
1, 2, 3, 4 – No same number in the row or column	Allow	Allow
1, 2, 3,4 – Same number in the row or column	Disallow	Disallow
@\$2-21	Disallow	Disallow

No same number in the row or column

	A	B	C	D
1				2
2	4			
3		3	1	
4				

A1

What is the number being put inside the grid? Insert C for Clear

OK Cancel

	A	B	C	D
1	1			
2				2
3	4			
4		3	1	

Same number in the row or column

	A	B	C	D
1	1			
2				2
3	4			
4		3	1	

B1

What is the number being put inside the grid? Insert C for clear

OK Cancel

Invalid data. Please insert a number that isn't in the same row or column!

OK Cancel

ESC

	A	B	C	D
1	1			
2				2
3	4			
4		3	1	

Random letters or numbers too big:

	A	B	C	D
1	1			
2				2
3	4			
4		3	1	

B1

What is the number being put inside the grid? Insert C for clear

OK Cancel

Invalid data. Please insert a number between 1 and 4!

OK Cancel

ESC

	A	B	C	D
1	1			
2				2
3	4			
4		3	1	

It clears:

	A	B	C	D
1		3		1
2	2			
3			4	
4				

A2

What is the number being put inside the grid? Insert C for Clear

C

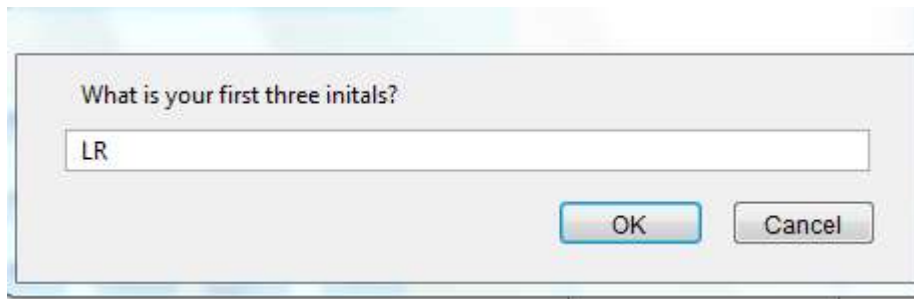
OK Cancel

	A	B	C	D
1		3		1
2				
3			4	
4				

The users Initials:

Test Data	Expected	Results
LR – Normal	Allow	Allow
L or LRG – Extreme	Allow	Allow
LLLL or 4444	Disallow	

Normal:

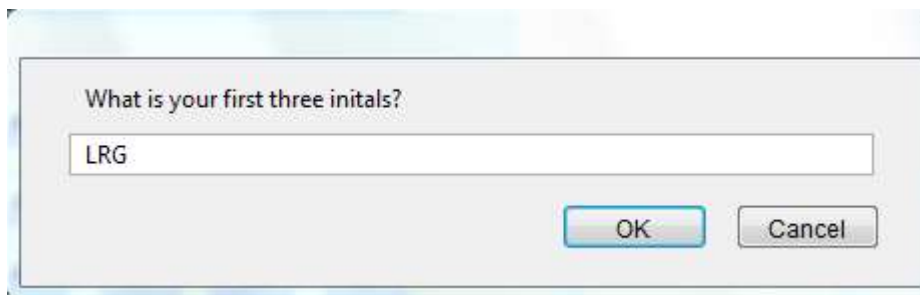


A screenshot of a Windows-style dialog box with a light gray background and a thin blue border. The title bar is not visible. The dialog contains the text "What is your first three initials?" in a black sans-serif font. Below the text is a white text input field containing the letters "LR". At the bottom right of the dialog are two buttons: "OK" with a blue border and "Cancel" with a gray border.

Coming from leaderboard

LR 586

Extreme:

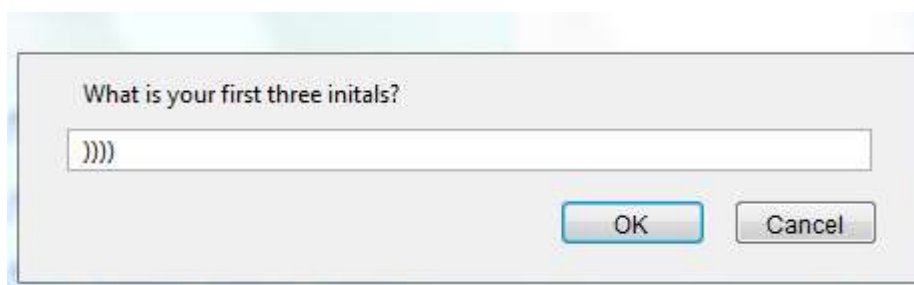


A screenshot of a Windows-style dialog box with a light gray background and a thin blue border. The title bar is not visible. The dialog contains the text "What is your first three initials?" in a black sans-serif font. Below the text is a white text input field containing the letters "LRG". At the bottom right of the dialog are two buttons: "OK" with a blue border and "Cancel" with a gray border.

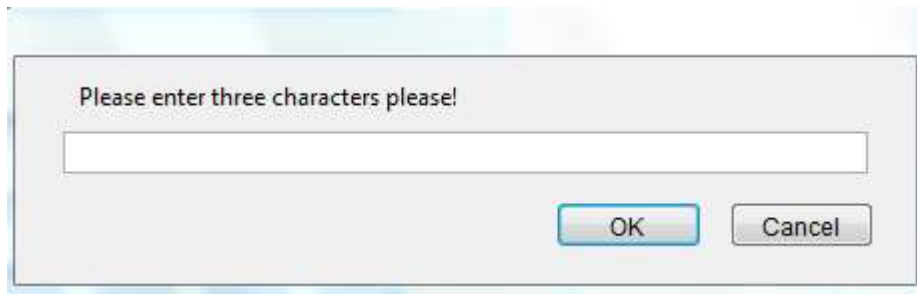
From leaderboard

LRG 63

Exceptional:



A screenshot of a Windows-style dialog box with a light gray background and a thin blue border. The title bar is not visible. The dialog contains the text "What is your first three initials?" in a black sans-serif font. Below the text is a white text input field containing four closing square brackets "]]]]". At the bottom right of the dialog are two buttons: "OK" with a blue border and "Cancel" with a gray border.



ESC



Bubble Sort:

User information will be put into the document Leaderboard and will sort it out with the current user's information:

LRG 54

OPQ 32

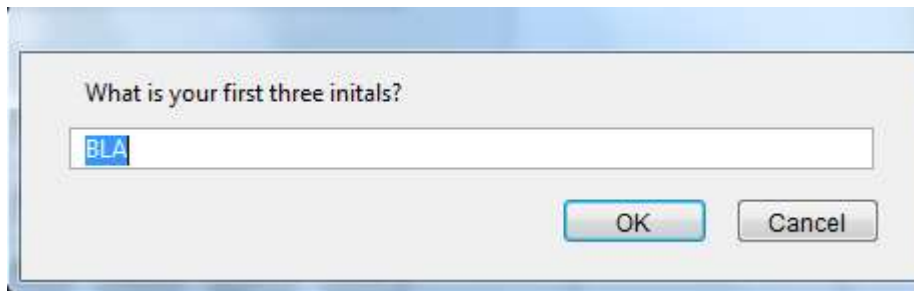
ARF 20

AEW 21

LKI 40

Users Time -

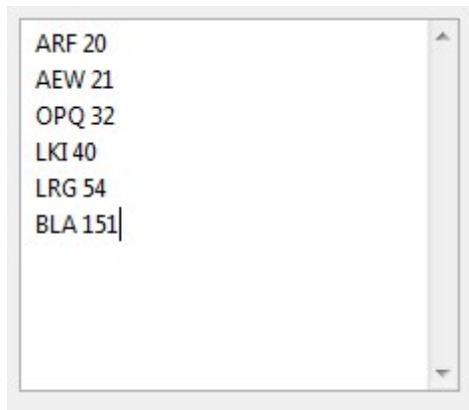
151



What is your first three initials?

BLA

OK Cancel



- ARF 20
- AEW 21
- OPQ 32
- LKI 40
- LRG 54
- BLA 151

Systematic User Testing

I have sent my game to five different people and gave them a survey to fill out so that I can get feedback on my game. These are the people's results from their surveys:

Name: Cameron Ewing

Did you like playing my game?

Yes

What is your favourite part of this game?
--

I liked the puzzle aspect of it
--

What do you think that could be improved in the program?

The generation of numbers

What features could be added further into the game?
A bigger grid

Did you have any problems playing this game at all?
When I entered a digit the grid moved down some of the time, and other times the game was unbeatable

Name: Cameron Gemmell

Did you like playing my game?
yes

What is your favourite part of this game?
The game is randomly generated every time.

What do you think that could be improved in the program?
The game could generate so it is possible to win every time.

What features could be added further into the game?
Different difficulty settings where more or fewer numbers could be shown at the beginning.

--

Did you have any problems playing this game at all?
Some games were unwinnable and the numbers on the grid changed unexpectedly.

Name: Euan Hepburn

Did you like playing my game?
It was alright

What is your favourite part of this game?
Completing it

What do you think that could be improved in the program?
Don't make the lines jump when the first number is entered

What features could be added further into the game?

Expand it to 9x9 grid

Did you have any problems playing this game at all?
--

There were several jumping value errors but once they stopped happening it worked well

Name: Marc Butler

Did you like playing my game?

No

What is your favourite part of this game?
--

How it changes it randomly throughout the game

What do you think that could be improved in the program?

That it can give out unbeatable games.

What features could be added further into the game?
--

More difficulty settings and maybe a bigger grid

Did you have any problems playing this game at all?
--

It would move the grid down one and would give me unwinnable games.
--

Name: Ross Kelly

Did you like playing my game?

Yes it was v. good and I had fun

What is your favourite part of this game?
--

The leader board part as I can show off my skill

What do you think that could be improved in the program?

It could have more difficulty levels

What features could be added further into the game?
--

More difficulty levels and a bigger grid

--

<p>Did you have any problems playing this game at all?</p> <p>It seriously moved the data grid down one row halfway through my record breaking Sudoku run</p>

From what I have gathered from my systematic user testing I have found out what people like about my game and what people didn't like and what problems I missed through the implementation of the game.

Reflective Commentary –

The whole testing part showed me problems with the code which helped a lot for many reasons as I might have missed something and livecode would tell me what was wrong so like parameter passing was wrong one of the buttons so it told me pacifically where the problem was. Having people testing my game was fun and the people who were testing gave me good feedback and advice.

Evaluation

This project has tested all my knowledge of Computing Science and programming in Livecode.

Program:

The program was mostly close to the requirement specification as it does have a 4x4 grid that goes from 1 to 4 that validates when entering a number into the grid checking if the number is not in the same row or column of the destination it's being placed in. It has a leader board that saves and loads the leader board from a certain file location and has a timer that is used to give a result for the leader board and same with the initials which uses input validation to check if the length of the text is bigger than or equal to 1 and less than or equal to 3. The two things that aren't a part of my program is the different difficulties as in

the requirement specification I said I would include difficulties but I found out that creating different difficulties and having different leader boards that will switch for those difficulties was hard to implement. The use case is completely different from what the user will have to do now because of the added A1-D4 buttons and that the complete isn't asynchronous with the program. The users specified in the personas, user stories, user scenarios and use case will only fit the one which is Jordan Ryder as it would only be one difficulty so Jordan should be able to play which was originally the easy mode but Dominick Matt will be put off as the game only has one difficulty now and would probably not get the game which in a real life scenario if a company was developing this game they could lose sales due to the lack of difficulty.

This program has been a massive pain taking most of the time out of the whole project. At the start I had no idea how I will program the game at all but looking at it now it seemed like most of this stuff made sense. I had to get rid of the difficulties as it would give me trouble as I will need to switch the leader boards for easy, normal and hard which I had no knowledge to actually do it.

With only one difficulty setting everything went swimmingly and only thing that troubled me through was with the data grid as my knowledge of developing for the data grid was pretty limited so using livecodes tutorials online on how to use data grids and using another program that was made using a data grid helped me through development. Another problem I faced was the file handling as I had no knowledge on how to set a location to a certain place like the Documents for example so I looked up online on livecodes' website and found the code which I modified and made it my own. There were problems with getting the numbers inputted into the boxes as at the start I already made a program that allowed me to type into the data grid boxes but when I tried it here it wouldn't work so I used 16 different buttons for the 4x4 grid.

Another problem was time as I tried to make it recurrence but that would break the entire program and crash it so using the source I had in the feasibility study I modified it for my purpose and created the timer. When creating the file handling I created the bubble sort to sort out the players initials and their time (which was another problem I'll get to) which didn't work at all so I went and looked up some other sort which couldn't work for switching the name as it could only do the time. I then went back to the original code I had and later realised that the variable that was coming into the bubble sort procedure was a local variable and not a global variable and after I changed it, it worked. I also had a problem with the initials and time that the users first three initials and time would not go into with the other initials and times which I found out that I had to put into the loop and then put it into loop + 1 which then worked out fine.

In the survey the testers found problems with the program like for example that sometimes when A1 or a similar button is pressed everything in the data grid moves down and now looking at the game I have no clue why it does it as I have checked the code and don't see

any problems with it. I only have one idea why it does this which is that due to the data grid having its column headings hidden the column headings still register as a row so everything moves down and that users can be given unwinnable games so that would be for example the first row the number 1 is put into column 1 and 2 is put into row 1, column 2 and 4 is put into row 1, column 4 and then 3 is put into row 2, column 3 which would game unbeatable as the user can't enter the number inside row 1, column 3. This whole idea of unwinnable games I didn't anticipate at all but I know this could be easily programmed as A1 – D4 buttons code can be added on to incorporate checking the second row if numbers have been entered in the first row etc. Problem I found after going back to the program is that users can override the randomly generated numbers making it so that they can put in anything they want and trying to fix this would in my opinion be too hard to program using in livecode.

The feedback I got from the ones who play tested my game was it was mostly good and besides the problems that occurred above they still enjoyed playing my game. There also could be problems with people who have dyslexia that might not be able to read what it says and that could be easily changed by just changing the font size of the program so all questions will be in a size they can read.

What I could have been included into the program:

If I could start this again I could've used SQL to create a table that would be linked to the program so that all the leader board results will be held in a SQL database which could show my skills with computing science especially at advanced higher. I could've added difficulties to the game as the user would get to choose a difficulty that will give less and less numbers so that it can appeal to a bigger and wider audience. From the survey people wanted a bigger grid which I could've done something like a 6x6 or an 8x8 grid and maybe using the different sized grids as difficulties.

What I have learned:

That I have realised throughout the project is that the implementation was not as bad I thought it was as at the start I had no clue how to program my game and I was worried that I would've skipped passed most of the implementation but I broke through using notes and my own skills and created a working Sudoku like game which I shouldn't worry that much on the future and focus on what I have currently going on now.

Record of Progress

Date	Work Complete	Help Required	Reflective and Evaluative Commentary	Related
25/10/16	Started to research the project I am going to do.			
26/10/16	I have created my project proposal for the Sudoku game I will be making.			
8/11/16	I have continued working on the Project proposal and finally finished it and have went onto the researching part of the project		Should of thought of an easier project to do as I don't think I could possibly program it through	..\Proposal\Project Proposal.docx
15/11/16	I have started to research and create a user survey for my project			..\Project Plan\Feasibility Study.docx https://www.surveymonkey.co.uk/r/PSPNWC2
23/11/16	I have completed the little part I needed to do for the feasibility study which was the additional techniques part of the study and have finished my user survey which has 10 questions and has been sent out on Yammer to collect responses.		Should have used another way to get more people to do my survey.	..\Project Plan\Feasibility Study.docx https://www.surveymonkey.co.uk/r/PSPNWC2
29/11/16	I have completed the assessment of end-user requirements showing what the end-users would want for my Sudoku game.			..\Project Plan\Personas.docx ..\Project Plan\Use Case.docx ..\Project Plan\User Scenarios.docx ..\Project Plan\User Stories.docx
14/12/16	I have created a basic test plan and know started to add more detail on how long it'll take in reality.			

05/01/17	I have finished of the whole of the project plan and know are on the requirement specification part of the project		Should of done this a lot sooner	
06/01/17	I have completed the requirements specification on time explaining what the program will do, what will the end users be and what is the functionality of the program and started the test plan finding out what requirements will need to be tested and creating test data and criteria.		Should have completed this months ago.	
11/01/17	I have looked at previous students projects so I have a more of an understanding on what to do and what not to do and started an Interface design doing a wireframe and also tweaked a few things for project, test plan and the requirement specification.		Quite helpful looking at those projects which can probably help me more to get through the project.	
19/01/17	I feel that I have finished my Interface Design as I feel this will be the layout that will fit the requirements for the game and that all of what will happen user interface wise follows my requirement specification.		This took me less time than I originally thought.	
24/01/17	I have started the program/data structure design using a draft on paper and then followed it up on		Should of thought on how I could create the program a	

	word and had to change a few things with my feasibility study as I had to learn how to create a timer in Livecode.		lot earlier as I feel like I can't do the 2-D array part of my game I'm creating.	
25/01/17	Kept adding onto the program/data structure design stuck on trying to figure out the pseudocode for the insertion sort that will be inserted for the leaderboards and I might just come back to the program/data structure design and just go onto the implementation		Wish I knew more about the programming parts like 2-D arrays and insertion sort	
1/02/17	I think I have finished the program/data structure design but I might have to come back to it to change due to the program might be done in another way and that I might have to change how I do 2-D arrays. I also changed the Insertion sort to be a bubble sort as I feel I can confidently make a bubble sort. I also changed a few things inside the requirements specification as it changed from a insertion sort to a bubble sort and added a bit more onto the test data showing what it should do when sorting and added everything into one word document		Right about now I think I can scramble a working program that does what it should. I have realised I thought it would take me a long time to do some parts of the project but quite frankly it was quite quick	
07/02/17	I started on the implementation working on creating		Figure out a way to update and	

	the three difficulties and inserting the grid		validate	
14/02/17	Trying to figure out how to insert numbers into the grid as I was able to enter numbers by just clicking but now I can't with this game.		Find a way to get numbers into the game.	
15/02/17	Made buttons to insert numbers into grid so what I have done is created the file handling and created validation for the buttons		The buttons won't insert the numbers at that time.	
22/02/17	Resourced a program that is Knots and Crosses that have grid code so I used that and warped it so that it will work with my code and also finished up on the validation for the difficulties		Should've done this earlier	
23/02/17	Created validation for some buttons so that the same number can't be on the same row or column and will do so for the rest of the buttons.		Could've created it for all the buttons at that time.	
28/02/17	Still updating the buttons going through it slowly to make sure I don't mess it up.			
02/03/17	Finished the A1-D4 buttons and now went onto to change the file handling as it doesn't work and I will need to check what is wrong with it			
06/03/17	Found out why it wasn't working which was a variable was a local variable instead of a global variable which wouldn't allow the data flow through the program. The only			

	thing I need to work on now is the complete portion of actually checking the game if it is complete.			
07/03/17	Found a way to check the game by checking that everything isn't blank and that every row and column is filled with numbers. Also finished most of the final testing and the only thing I need to finish is getting people to play test my game			
08/03/17	Got people to play test my game and found out some problems I didn't see before at all but using this feedback I started my evaluation for my project talking about the program and what I could've done etc.		After realising the problems that were present. I kind of wished I saw them in the first place	
09/03/17	Finished my evaluation and went back to do some changes and updates for parts of my project telling what has changed due to the implementation.			
13/03/17	Finished all of the changes and also done all of the reflective commentary for all of the parts of the project talking about what succeeded and what didn't and what got changed etc.		On time but barely. Could've gone quicker.	
14/03/17	Added more onto the program/data structure talking about the data flow through the program and describing what global variables are being created and used in			

	each section of the program and also talked about accessibility in the interface design and conclusion			
15/03/17	Went back to the project plan and added on talking about the different feasibility's and relating it to my game, added more onto the reflective commentary and moved my survey for the game that was blank up to the test plan. I also talked about the results for my end user survey and my systematic user survey as well.			