Math Game

1. ANALYSIS

1.1 Project Proposal

1.2 Research

1.3 Initial Modelling

1.4 Final Objectives

2. DOCUMENTED DESIGN 2.1 Overall System Design

2.2 Data Dictionary

2.3 Validation

2.4 Files

2.5 Algorithms

2.6 Data Structures

2.7 User Interface

3. TECHNICAL SOLUTION

3.1 Program Code

4. TESTING

4.1 Test Plan

4.2 Test Evidence

5. EVALUATION

5.1 Evaluation of objectives

5.2 Evaluation of whole solution compared with the start

5.3 User Feedback

5.4 Analysis of user feedback

5.5 Extensions and Improvements

**Page 3**: Analysis section heading

**Page 4** Design Section heading

**Page 5** Technical Solution heading

**Page 6** Evaluation Section heading

* 1. Project Proposal

Project Proposal

The user is Winn. She is my mother and I will have a lot of time to access with my user because I see her every day. However, the focus of the program will be towards my little sister because she will be using it a lot however, I have listed my mother as the user due to the fact the program will print out my sister’s progress.

The program will generate random basic math problems and will ask my sister to answer it. The program should also print out a report on how my sisters doing and what she struggles on. The program will be for personal use not for businesses or an organisation.

The problem with the current ones they either require a fee, are too hard or too basic meaning my mother must sit and manually give her questions with pen and paper. Which means my program would save my mother time allowing her to concentrate on other things.

Possible Objectives:

1. The program will generate a couple of numbers between 1-200
2. For the multiplication section the numbers will be randomly picked out from a set numbers depending the option the user picked out. For example, if the user picked out the easy section then the numbers would only be randomly being picked out from 1,2,3,5,10.
3. The program will also have multiplication with the generated numbers moreover it will be up to 20 \* 20
4. In the basic difficulty it will be very easy multiplication so the 1,3,2 ,5 10
5. In the Normal difficulty it will be normal such as 4,6,7,8,9,11,12
6. In the Hard difficulty it will
7. The program will also have division section using the same basis as the multiplication
8. In the basic difficulty it will only go to 20. For example, 20/5
9. In the normal difficulty it will only go up to 99. For example, 99/3
10. In the Hard difficulty it will only go up to 200. For example, 200/50
11. The generated numbers will also be integers because they can’t do decimals yet.
12. The reward system will also be a picture or animation saying “Well Done” with bright colours so it can appeal to the young audience.
13. There will also be a score system to keep track so that the user can see their score
14. I will put in a high scores chart that will work like so:
15. Human Calculator (Got 90% or above)
16. Genius (80% or above)
17. Mathematician (60% or above)
18. A permanent storage to store data about the user such as what their score was, what mistakes they made last time and other high scores
19. Also a leader board system is in place so that the user can know how well they do.
20. For addition the computer would randomly generate numbers
21. For the basic difficulty the addition will be limited to 50
22. For the normal difficulty the addition will be limited to 100
23. For the hard difficulty the addition will be limited to 200
24. Subtraction will follow the same footsteps.
25. A permanent storage to store data about the user such as what there score was, what mistakes they made last time and other high scores
26. Also a leader board system is in place so that the user can know how well they do.

1.2 Research

1.2.1 Interviews

Date: Saturday 2nd February Time: 2PM GMT Duration: 30mins

Interview 1

For the purposes of the project I will call my sister: Daisy (which is not her real name)

Will you have time to see me regularly about the project between February and December?

*Yes I will be able to see you regularly.*

What area needs to be computerised?

*The maths side of learning for a 9yr-11+*

Is there anything that you have a problem with?

*Too many reports must keep them at hand and don’t know exactly what she’s doing.*

Are there other users who could be using my system?

*My daughter, Daisy will be a regular user.*

How confident are you in using computers?

*I am confident enough.*

Does the user have a PC based system? Do they have a printer?

*Yes, I have a laptop and also I have a printer.*

Can I arrange another time so that I can observe how your system currently works in practice?

*Yes, a time that fits.*

How do you keep track of her learning?

*I check her homework regularly and we get end of term reports.*

What problems are there that you think could be solved by my computerised system?

*Would be able to see what specifically what Daisy is struggling with.*

What’s the current system?

*Doodle maths and written reports.*

Is it difficult keeping all these papers?

*Yes*

What does she learn about?

*Maths leading up to 11+ and currently division.*

How difficult is the maths?

*For her it has a range of difficulty.*

Would you like the program to learn about her?

*If necessary.*

Would you like to see what she struggles on?

*Yes, that would be key thing.*

How appealing should it be to a 9yr old?

*It doesn’t have to look that good but I would like it to be easy to use.*

How much time would she use it on the daily?

*10-20mins is what the school recommends.*

Does the current system use money?

*Yes, Doodle Maths is a paid subscription.*

What are the drawbacks of the current system?

*More expensive and don’t quite clearly know exactly what she does.*

Would you like any new features within reason in the program?

*Not that I can think of right now however I may ask for one at a later date.*

Would you be satisfied?

*Yes.*

Conclusion:

I have concluded that the important details are that it needs to be easy for Daisy to use moreover it needs to have a range of maths questions. It’s also key that it reports the findings such as what she gets most incorrect and maybe even a record of her answers.

Interview 2

Will Daisy be doing percentages or divisions?

*No she will not be doing percentages however I would like it if there is division question*

How hard should the multiplication questions be?

*They shouldn’t be too tough so limit it to 20times tables*

Would you like to see her score?

*Yes, if possible*

What kind of multiplication should it be?

*The multiplication should be ordered into sections she finds easy, normal and hard. So for example 2 times tables and 10 are easy. 9 and 3 are normal. 7 11 19 14 are relatively hard.*

How hard should the division be?

*It should be up to 400 and Always equal to a whole number for example 400/25 but not 384/19*

*Which will be too hard.*

How hard should the addition and subtraction be?

*They should be relatively easy so keeping it to three digits and double digits is enough*

How long should she be on the program?

*She should be on for around 10 minutes*

Should the questions be randomly generated?

*Yes, if possible it would be random*

Would you like fractions?

*They haven’t learnt that yet so no.*

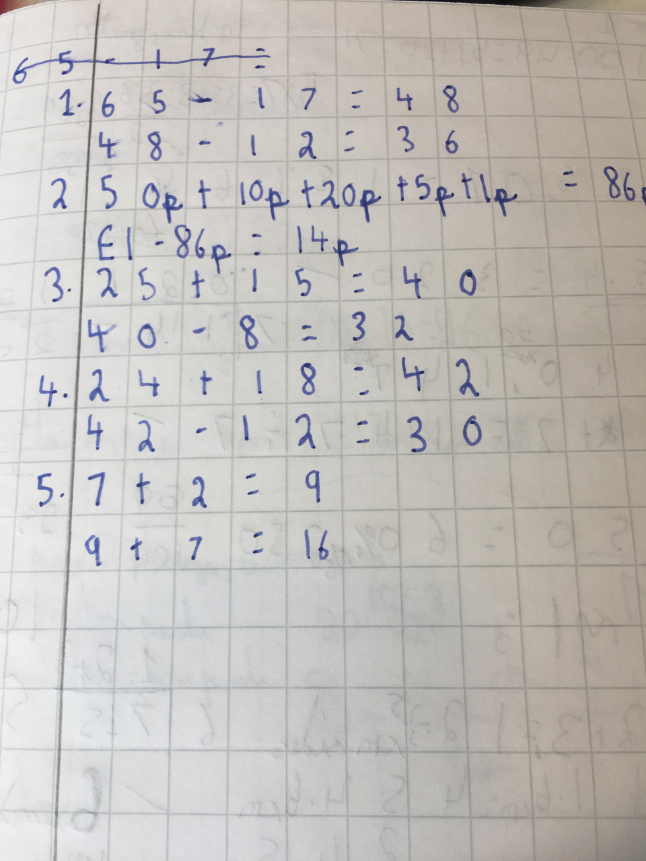
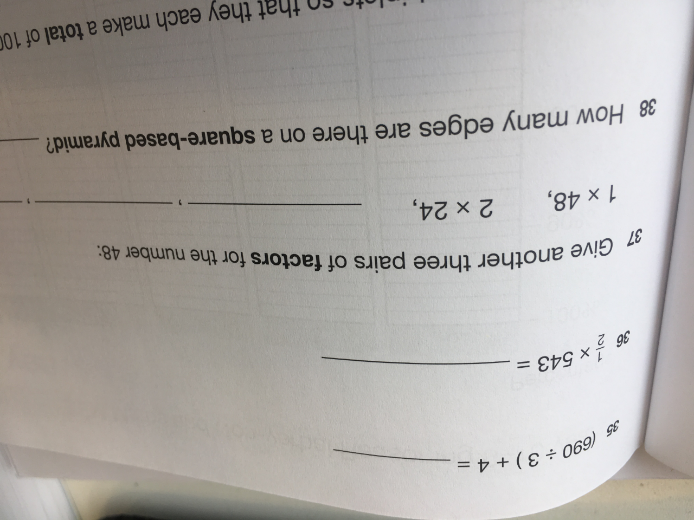
Conclusion:

I should make my program interesting enough to keep Daisy focused for 10 minutes and the difficulty will have to be altered from the way I had it moreover I will have to keep a scoring system so my mum can keep track of everything that’s happening. Also it won’t have factions or percentages.

1.2.2 Source Documents

Source Documents Here:

Interview 09/03/19:



How do you keep these documents?

*I store them in a filing cabinet to keep them safe*

How many of these papers are there?

*Most of them are kept by the school.*

How would you like my program to replace these documents?

*If at best, I would like it to come with some permanent information of what she gets wrong so I can go back and look at it. Like a word document with clear information*

Is there anything else that I should know about the documents?

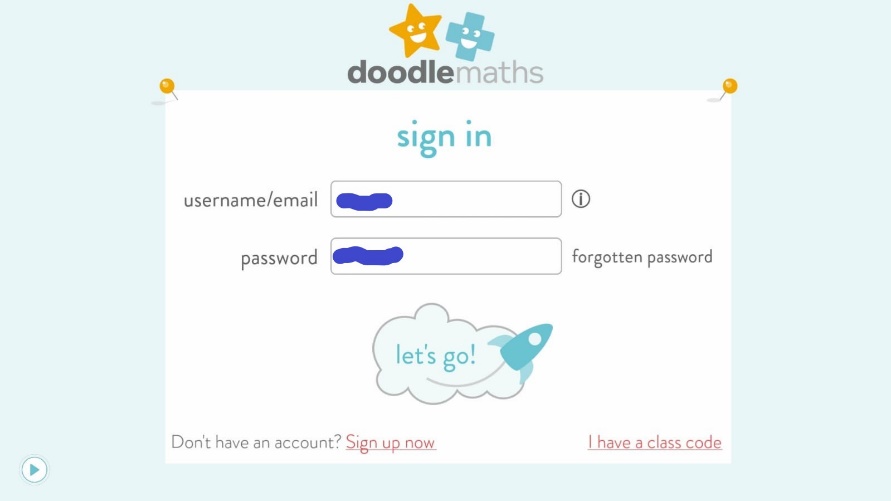
*Sometimes we don’t get a report or any pieces of paper on what Daisy does so it would be amazing if your program would regularly be updated.*

Conclusion:

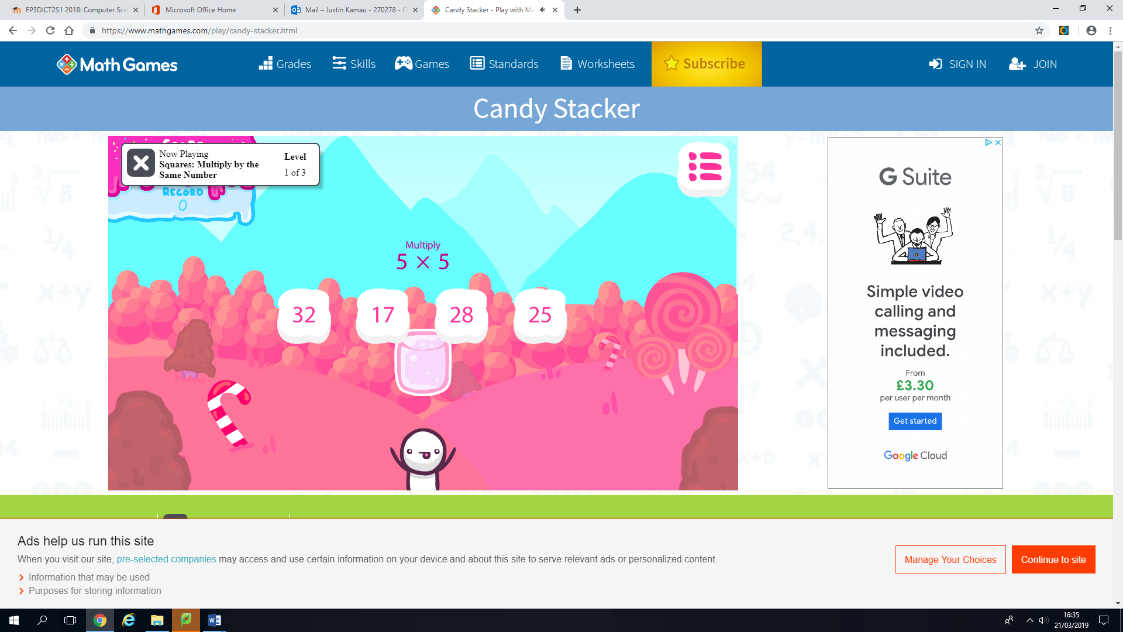
I will make sure that I can keep a permanent record of what Daisy gets as a score in a separate file and even what questions she got wrong. This would make it easy for mum to know exactly what Daisy Needs and how she can improve. Also I will try to make the file where the data is being stored as user-friendly as possible

Doodle Maths (App)

1.2.3 Internet Searches



https://www.mathgames.com/play/candy-stacker.html



As you can see in the Math Games website it has redeeming quality of being very user friendly moreover its free on the contrary it has a lot of ads as you can see to the right therefore the child may even get distracted which isn’t good also it won’t work at all without internet. Also its very buggy when used on a mobile device and it’s not compatible on some computers such as windows XP. Also it doesn’t save anything about the user so you can’t see the repeated mistakes because every time you reload the page it treats you as a different user. This was the best out of the free maths games you can get.

Doodle maths is a paid app and is very expensive however unlike Maths Games it doesn’t support ads. Also you can sign in allowing you to save your score and see what you get wrong. However the problem is its very expensive and is recommended for schools not for individual users meaning its not that personalised.

1.2.4 Questionnaire

1.2.5 Direct Observation

The current system Doodle Maths: *Every learner has their own unique Zone of Proximal Development (ZPD) - the narrow band that is the difference between what a learner can do without help, and what he or she can do with help. Doodle Maths identifies this zone and creates a daily-practice work program based around it, filling in gaps and ensuring continual progression.*

The problems faced with Doodle Maths: The current system unfortunately is a paid app moreover the app also has some glitches with the users.

The Users told me this: *Sometimes when using Doodle maths, it randomly crashes even though the phone is more than capable to handle it,*

After investigating for a bit, I found out that it had an amazing GUI and was very appealing to young children. However, this brings many drawbacks such as because of this amazing GUI it uses a lot of storage. Also, I saw sometimes the app would freeze randomly then then when the child put it the CORRECT answer it answered with “*Sorry you got the problem wrong*”. Although It has those drawbacks it has a huge benefit which would be that it can learn about the child and particularly see where the problem is with their maths.

Time 10/02/18 15:46

Duration: 29mins Location: Home Done By Justin Kamau

1.3Initial Modelling

1.3.4

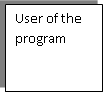
# Data Flow Diagram

Young Children Maths Game 12/16/2019

User details

User Details

User inputs their name



User Preferences

Data store

What questions they got wrong

User of the program

User Preferences

Question details

Question Detail

User

Program shows high scores and where they are on the leader board

User plays the allotted questions

User chooses the difficulty of the level and division or subtraction

Calculations-

The calculations that I will be using for now (these may change):

The most important thing is how the score will be awarded. There will be a score system which will be done as so: In the HARD questions the score will be multiplied by a multiplier of 1.5 so for example if they got 15 right then their score would be 15 \* 1.5 = 22.5 and then that will be multiplied by 1000 and put on the score leader board. The normal questions will have a multiple of 1.3 and easy would have a multiplier of 1.0. This would encourage the user to put more effort and try the harder section and improve their maths.

EXAMPLE FOR SCORE@ : UserCAnswers \* UserDOption = UserHScore

UserCAnswers: Is the users correct answers

UserDOption: Is the user’s difficulty multiplier

UserHScore: This is the user’s High score for that session

For the multiplication section the numbers will be randomly picked out from a set numbers depending the option the user picked out. For example, if the user picked out the easy section then the numbers would only be randomly being picked out from 1,2,3,5,10. Then the computer would print out the two random numbers and multiply them. The computer would have no problem to multiply out these answers and check with the Users answers to see if they’re correct.

EXAMPLE FOR ACTUAL QUESTION:

(UserRNum) \* (UserRNum) = UserQuestion

UserRNum: Will be randomly picked from a pool of number which are chosen from what difficulty the user set it on

UserQuestion= Computers answer to the question

Text and Binary:

I will be using the text file to store permanent data for leader boards mainly to keep the record of the users score. This will be kept permanently and hopefully when the user looks back their score they can see how much they improved.

1.4 Final Objectives

Possible Objectives:

1. The program will generate numbers depending on which questions they’re answering; the difficulty will get harder if they get 90% of 10 questions correct:
2. Multiplication will generate random numbers up to 12 however if the difficulty gets harder then it will generate up to 20
3. Subtraction will generate random numbers up to 99 however if the difficulty gets harder then it will generate up to 499 (Also the program has been set so the first number is always bigger than the latter)
4. Addition will generate random numbers up to 99 however if the difficulty gets harder then it will generate up to 499
5. Division will generate random numbers 99 however if the difficulty gets harder then it will generate up to 299 (Also the program will make sure that the question always has an answer of a positive integer)
6. When the program gets harder the user is told by the program with a message informing them
7. The program will also have multiplication with the generated numbers moreover it will be up to 20 \* 20
8. In the basic difficulty it will be very easy multiplication so the 1,3,2 ,5 10
9. In the Normal difficulty it will be normal such as 4,6,7,8,9,11,12
10. In the Hard difficulty it will
11. The program will also have division section using the same basis as the multiplication
12. In the basic difficulty it will only go to 20. For example, 20/5
13. In the normal difficulty it will only go up to 99. For example, 99/3
14. In the Hard difficulty it will only go up to 200. For example, 200/50
15. The generated numbers will also be integers because they can’t do decimals yet.
16. The reward system will also be badges so it can appeal to the young audience and give them incentive to work.
17. There will also be a score system to keep track so that the user can see their score
18. I will put in a high scores chart that will work like so:
19. Diamond Badge (Got 90% or above)
20. Gold Badge(80% or above)
21. Silver Badge (60% or above)
22. A permanent storage to store data about the user such as what their score was, how many questions they did and
23. Validation to make sure the user doesn’t type in anything that might crash the program
24. Such as the user typing in a decimal will not work and will be informed of this by an error message
25. The user typing in characters that are not are not integers “@,£,<,&,a,E, Seventeen” and is informed by a different error message.
26. The user typing in nothing will be informed and be informed with an error message.
27. A canvas for the following.
28. For a fantasy creature dragon the left hand side about half the size of the wall. The wall should be about 5 bricks on top of each other and takes up half of the scree.
29. Images of badges
30. Messages that will say “Well done” or “Try again”.
31. An animation for when the user successfully is doing well after 10 questions. The animation will be colourful stars flashing brightly.
32. A hint button to help the user for when they’re struggling at a question. The hint button will recognise each question and be able to give different “hints” to each question
33. 20% of the wall should disappear every time they get a question wrong.
34. If they get two questions correct in a row, then they regain any walls they lost
35. Have all 4 different type of questions on one frame (Multiplication, Addition, Subtraction and Division)
36. They can directly switch from one type of question to another and keep their progress
37. As soon as they press submit, they can’t change their answer

2. DOCUMENTED DESIGN 2.1 Overall System Design

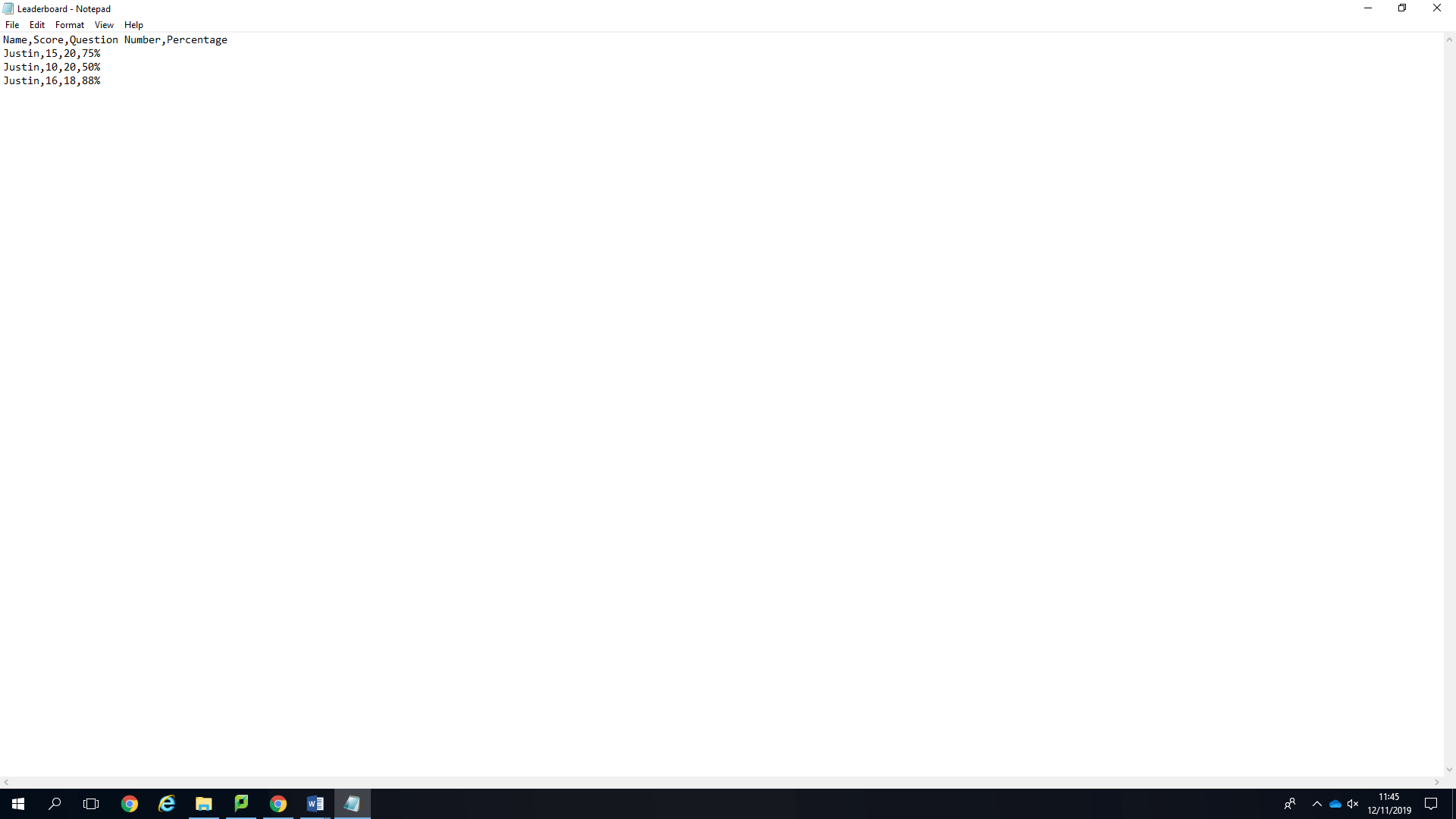
2.2 Data Dictionary

2.3 Validation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field(s)** | **Where** | **Validation Rule** | **Error Message and action(s).** | **When to check for error** |
| User entry | Multiplication, Division, Addition  and Subtraction frame. | When the user enters nothing in the entry field. | “You haven’t typed anything in”. Using try-except it allows them to press next question and try again. | When next question button is clicked. |
| Save button | User pressing save on any frame. | When the user presses save before the tenth question | “You haven’t answered enough questions”; “You need a minimum of 10 questions answered”. The program won’t save their score. | When the save button is clicked. |
| User entry | All Frames | When the user types anything that’s not an integer such as special characters or letters. | “Please type in an integer”. Deletes what they put and they try the next question. | When save button is clicked. |

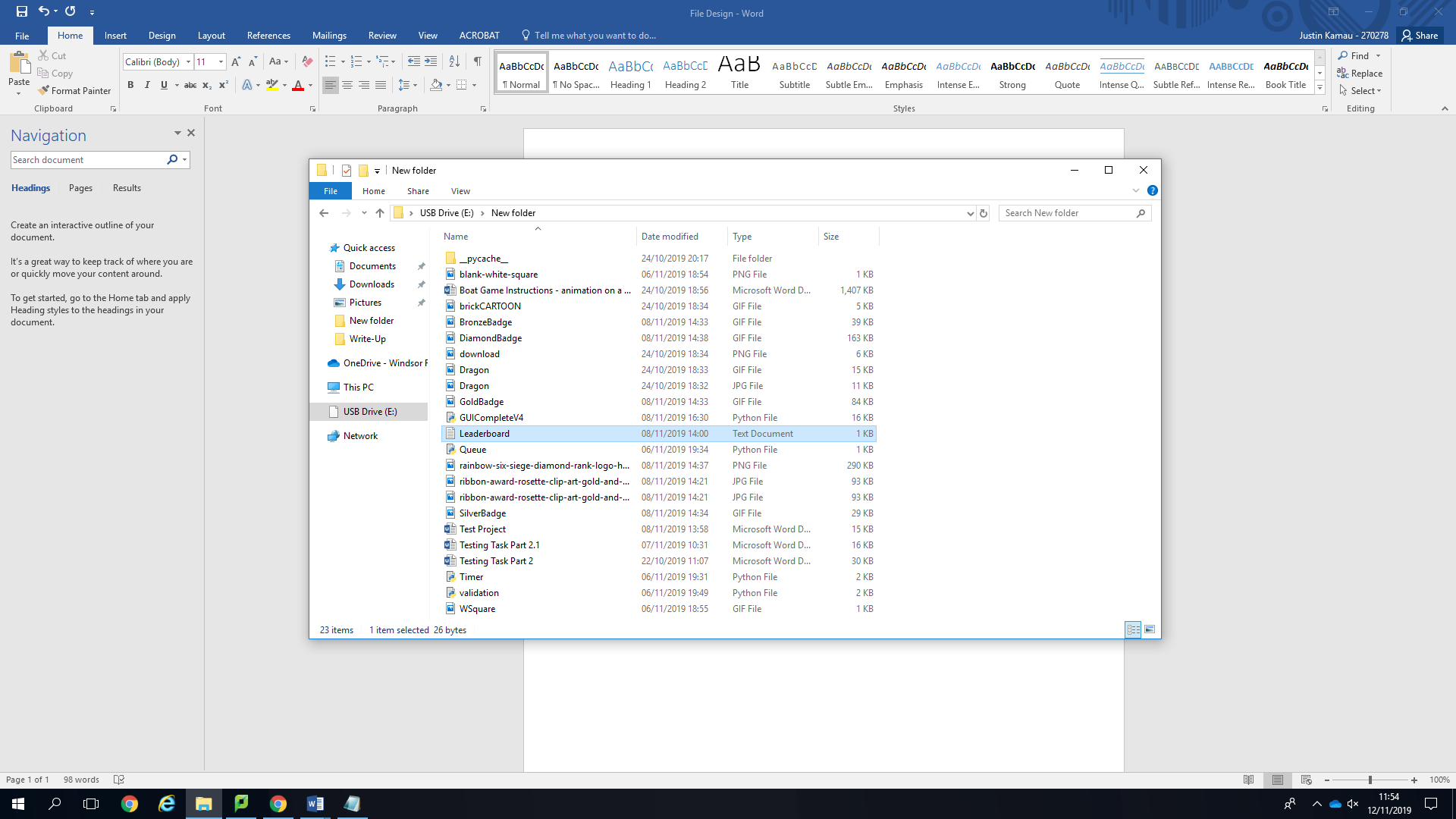
2.4 Files

File Design: FOR THE LEADER BOARD



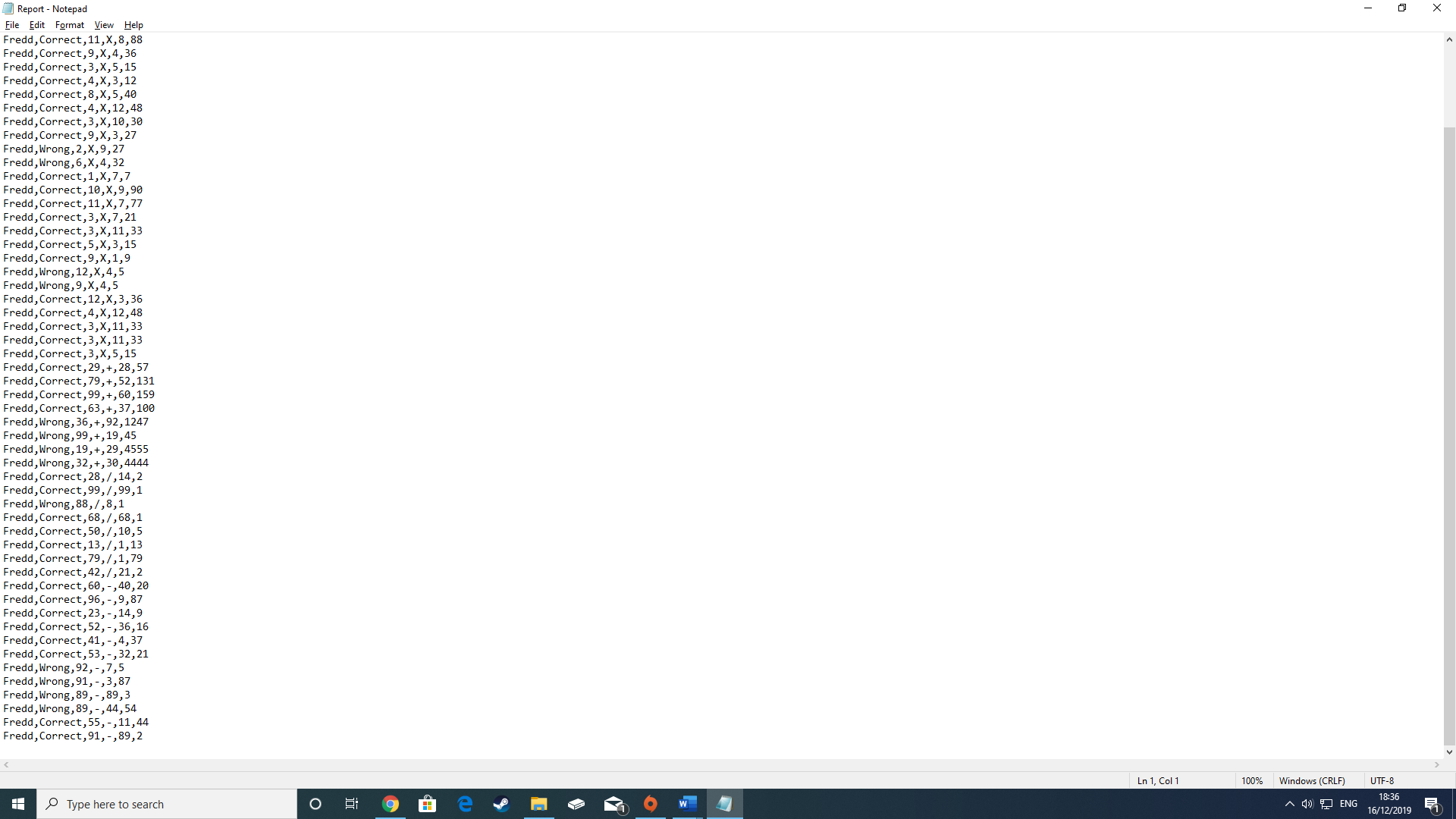
***Brief Description***

The text file is saved in a comma saved format so that the program can read through it if needed. It saves the name first then score then the question number. It saves both so that it can calculate the percentage very easily. Another reason why it’s saved is because someone with the same percentage but lower amount of questions will be equal if the question number wasn’t there.

***Program Details***.

The program prints each data value into the file using a comma to separate each file. The data is saved at the end when the user is finished and presses the save button. Only text will be saved here so the size of the file will be small compared with the main program. The main program will be around 20-30KB whereas the text file should be up to 5KB this helps the user because it uses miniscule amount of storage. The file name is useful “Leaderboard” so that it can’t get confused with anything else.

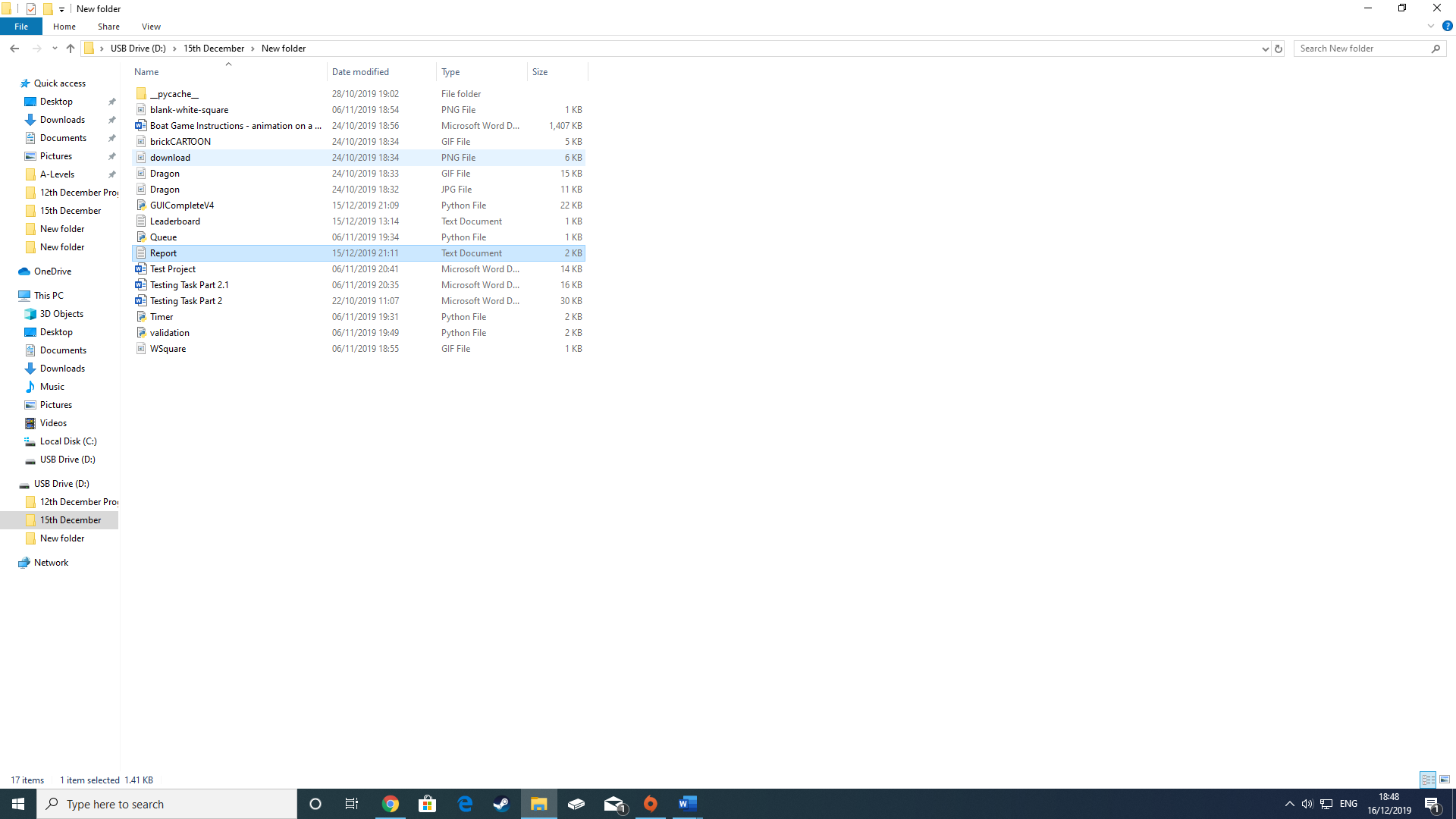
File Design: FOR THE PARENTS REPORT



***Brief Description:***

The program prints into the text file the following in the correct order: The users name, whether they got the question correct, The first integer for the question , the operator , the second integer and lastly the child’s answer. These are all key components which are used in the parents section of the program. They are used to show the parent which questions they get wrong so it can help the child improve.

***Program Details :***



The text file is mainly used in the parents section of the program. Thid utilises the questions they have got wrong and it allows the parent to use this information. This can help them with targeted revision or using the parent button to make the questions harder.

2.5 Algorithms

Algorithms

***Saving Algorithm.***

The standard procedure in this program is add data to a text file. The procedure uses a validation and uses an algorithm to save it using CSV. Th validation is used to stop too many questions being saved.

Procedure Save(self):

Check using a saved validation from the library if its less than 10 saves.

-Percentage 🡨 (self.HScore divided by self.QNumber) \* 100 so that the percentage was calculated

-Open the File called Leaderboard.txt

-Print the following variables into the text file (self.UserName,self.HScore,self.QNumber,Percentage)

-And separate them using a comma

-Close the file

The program is used to save the data into the text file so it can be used later by the program. The procedure starts with an instant validation from a library to make sure the user isn’t saving before they’ve answered 10 questions. If they have answered 10 questions, then the program instantly calculates the percentage of what they got to two decimal places. This makes it easier because it might cause an unforeseeable error without a limit on how many decimal places they can answer. After it opens the file and prints each variable with a comma separating each variable. Them when the program is finished it closes the file.

***Submit.***

The standard procedure in this program is to check if the user answer is correct. The program uses clever ways of “if” statements to make sure that they got everything right. The procedure also informs the user if they got it correct or wrong..

Procedure SubmitAnswer(self, QOne,QTwo,UserAnswer,lblresult,WhichFrame,lblHScore):

Change the state of the users answer box to read-only

Check if the answer is an integer less than 1000

X 🡨 The first number in the question

Y 🡨 The second number in the question

Z 🡨 Get the users answer

if the frame is the multiplication frame:

The operator is “\*”

Also if the frame is the subtraction frame:

The operator is “–“

Also if the frame is the division frame:

The operator is /

also the frame is addition frame:

The operator is +

End IF

If the right answer is equal to Z:

Change the label to say well done

Add one to the score

Or else:

Change the result label to say try again

Delete a brick from the canvas

End IF

Call the procedure badges

The program is used to do a lot of things in this submit procedure. The first thing it does is to change the answer box into a read only state so that the user can’t change their answer. Then after it calls the user answer and assigns it to the variable “Z”. Then it must determine which operator to use so it can calculate the answer using the WhichFrame variable which is passed in as a parameter. After doing so it checks if the User’s answer is correct. If they’re correct, then the label is changed to inform them they got it correct then one is added to the score. However, if they were to get it incorrect then label is instead changed to try again and then a brick is deleted from the canvas. This means that they’ve lost some of their wall health and if they lose all then the game ends.

***Badges:***

The standard procedure in this program is used to check what badge they deserve to be on then the quality of badge they deserve to get. With the diamond badge the best to earn then the bronze badge being the easiest to earn. The badges are used as an incentive for the child to do better on the game and learn.

Procedure Badges(self):

Grid the canvas

Difference 🡨 Number of questions – Score

If the Difference is below 2 and the number of questions is above 15:

Delete all the other badges

Get the Diamond badge image file

Create the image at the top right

Also if the Difference is below 2 and the number of questions is above 10:

Delete all the other badges

Get the Gold badge image file

Create the image at the top right

Also if the Difference is below 5 and the number of questions is above 10:

Delete all the other badges

Get the Silver badge image file

Create the image at the top right

Also if the number of questions is above 5:

Delete all the other badges

Get the Bronze badge image file

Create the image at the top right

End IF

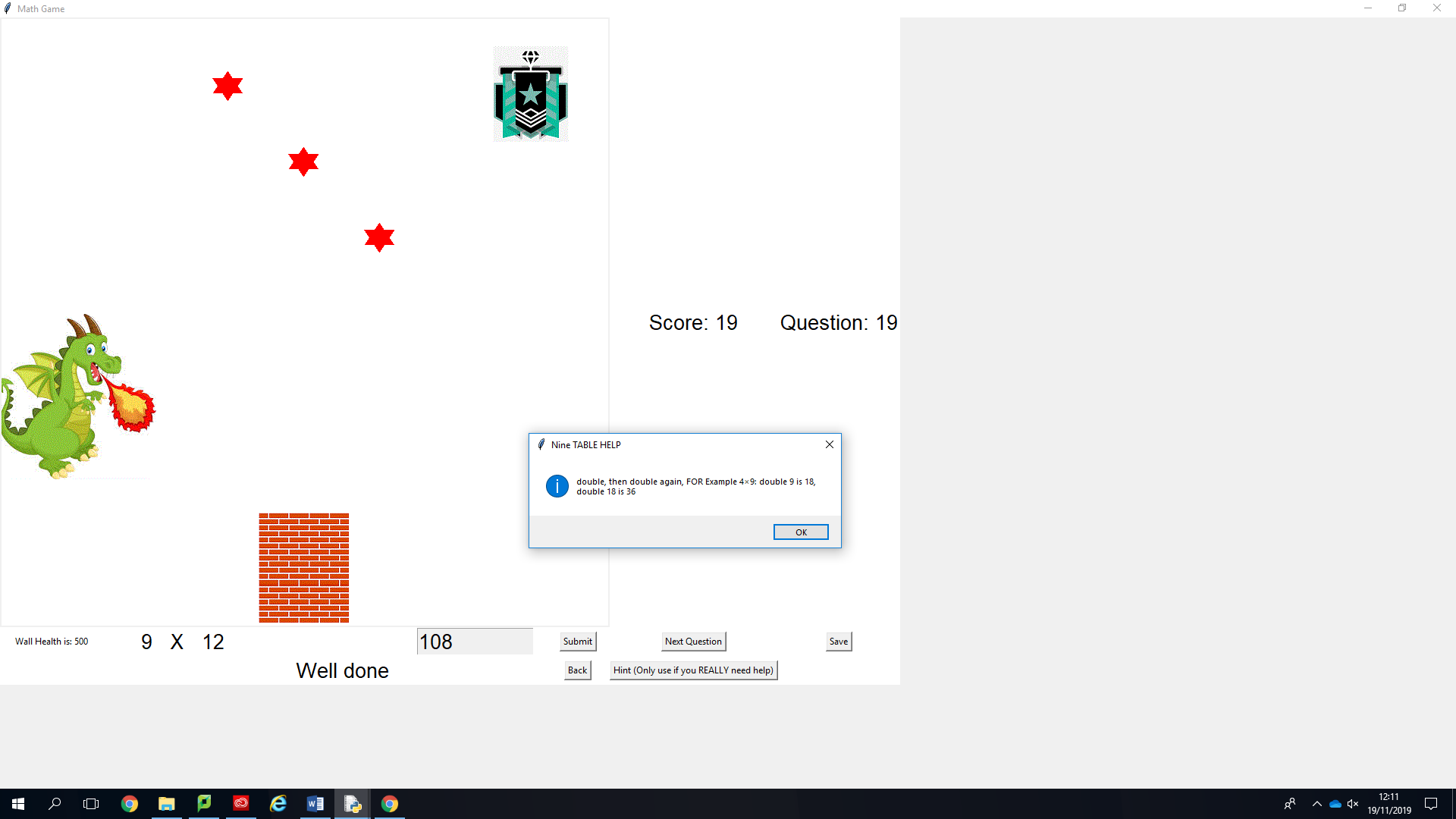
The procedure does not call any parameters in because all this procedure is doing is creating the badges. The first thing in the program is gridding the operation frame because that’s where the badges are going to be placed. The difference is basically the amount of answers they have wrong. The diamond badge means you have to have had less than two questions wrong and more than 15 questions answered. Then it deletes all the other badges because then it would have the badge stuck on top of another badge. It’s very tough to get the diamond badge however I had to make it hard or else it would be too easy moreover the user would get bored if they got the best prize all the time .Then it gets the diamond badge in the file which has already been resized and the coordinates are put in. It goes in the top right so it doesn’t obstruct the users view however they can see its there and be proud of their badge. The other badges have easier and lenient “Difference” because they’re not as good as the diamond badge

2.6 Data Structures

**Arrays/Lists**

**ScoreLists**. A huge part of the program. This is used to calculate the users highest all time score by adding all of their scores into one list. The program uses a csv reader to find the users name in the list and then it checks with the name they entered at the start of program. If the names are exactly the same then the program adds it to the list. To limit too many saves, the program requires you to have answered 10 questions before anything is added the list.

2.7 User Interface

Screenshots

5

8

10

9

7

6

4

2

1

3

1.[Question] The whole question is in a label of two integers named ‘QOne’ and ‘QTwo’ which are randomly generated. The integer is generated depending on which selection they picked on the menu screen. In the multiplication screen the integer will be up to 12 whereas in the addition section where it’s up to 100. There’s also the operator included in the Question label which is generated depending on which button they pressed.

2.[Hint Button] The hint button is put in place to help the user when they’re struggling and need a bit of help. When the hint button is pressed the user sees a message box with text helping them depending on the question. The hint button recognises what button they’ve pressed and gives a different hint for most questions for example; if they have the 9times table question it would be a different hint to an addition question.

3.[Badge] The badge is placed at the top is placed so the user can see their achievements moreover there are 4 badges. The user starts on the bronze badge however if they do very well and get less than 3 questions wrong they get the Diamond badge. The badge is very important for them to have good incentives.

4.[Stars] The stars will appear when the user gets an achievement of doing more than 10 questions answered with 60% of the questions correct. They are randomly coloured and flash vibrant colours.

5.[Dragon] This is supposed to be the antagonist of the game so the dragon wins if all the bricks are deleted. Therefore the dragon is placed to kind of show there’s something to work against.

6.The bricks are an integral key part of the game because they get deleted each time the user gets a question wrong and when the user gets two questions correct then they get their brick back.

7.[Submit] The submit button works to check whether the answer is correct. This is inter linked with the score because it has to add one to the score. The entry box is also locked once the user has pressed the submit button so they can’t change their answer.

8.[Score] The score is calculated after they press submit and it’s used to work a lot of variables in the program. It’s used to work out what badge the user is on for example. Also the score is put in a text file. The score is very important and is yet another thing that the user will try hard to improve.

9.[Save] The save button does a lot. The save button saves the users score, question number they were on when they saved, their name and the percentage of the score divided by the question number to two decimal places. The text file is read into the program and is used all over the program. The users all time high score is used and is shown as a label at the bottom.

10.[Next Question] This clears the user’s entry when pressed so that they can answer the next question. The next question button regenerates the whole of the question and then clears the result label.

3. TECHNICAL SOLUTION

3.1 Program Code

4. TESTING

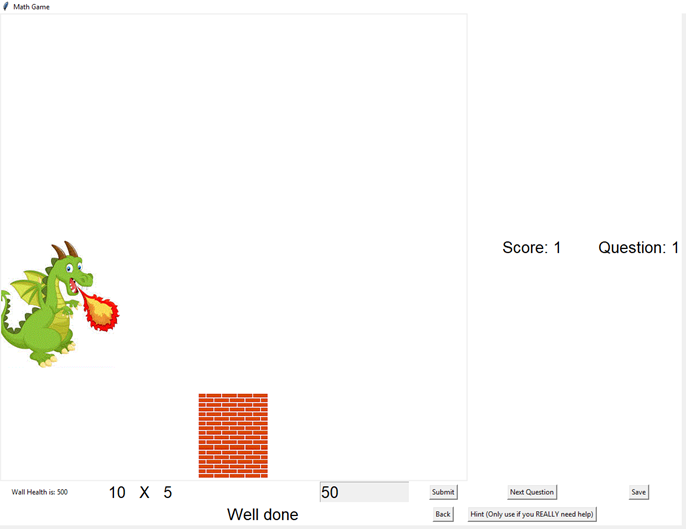
4.1 Test Plan

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test No** | **Objective tested** | **Test Data and description** | **Test Type** | **Expected Results** | **Actual Results** | **Comments** |
| 1 | User answers a question correct | The program should allow it to be correct. Program asks 5\*10. | **Normal data** | User Answer Should be 50 | 50 |  |
| 2 | User put in an incorrect answer | P2.1 The program should notice its incorrect .Program asks 6 \* 2. .Also 20% of the wall should disappear. | **Erroneous data** | User types in 0 which should be wrong. Wall should have been broken | 12 |  |
| 3 | User types in a decimal. | The program asks 5\*6 however the user types in 56.0 instead of 56 . | **Erroneous data** | The program should validate it and say it’s incorrect with an error message saying type in an integer. | 56 |  |
| 4 | **User** types in correct answer then an integer. | The program asks 7\*10 and the user types in “*70 2*” | **Erroneous data** | The program will instantly recognise its wrong instead of seeing the correct answer and allowing it. | 70 |  |
| 5 | The user tries to save on the 9th Question | The user tries to save on the 9th Question (the minimum being 10) | Boundary data – Data is | The user is greeted with a message telling of them they need to answer more questions before saving |  |  |
| 6 | Testing the score | The score starts at 0 and when the user gets answer correct, added by one. So start score should be 0 then with a correct answer it should be 1. | Calculations Test | Score should go up by one.  First Score: 0  After correct answer: 1 | It does. |  |
| 7 | P5.2 Testing the percentage just before the save. | When the user saves their score, they also get a percentage of how well they did. Printing the percentage and rechecking with a calculator to see if it’s correct. Also, its in 2 decimal places. | Calculations Test | The percentage should be correct to 2 decimal places.  If the percentage of their score/Total question answered is 16/19. Then the answer should be 84.210526  However, the program should round it to 84.21% | Correct  Answer = 16/19  Percentage= 84.210526%  Percentage in program = 84.21% |  |

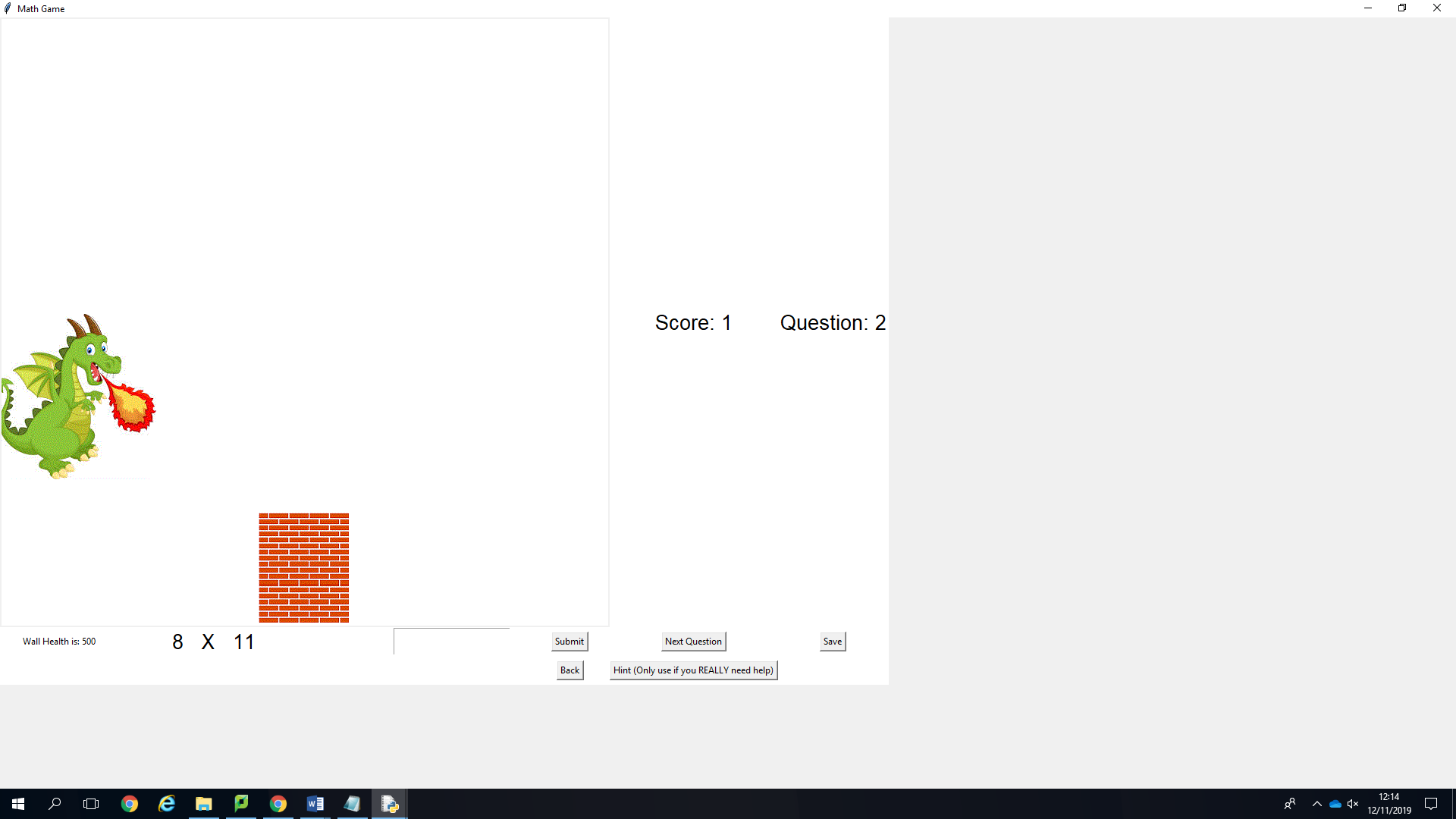
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test** | **Objective tested** | **Test Data and desciption** | **Test Type** | | **Expected Results** | **Actual Results** | **Comments** |
| 8 | Saving to a file | I can save to a file successfully with the User’s name and other details | **Objectives Test** | | Expected to save without any problems | Worked as expected |  |
| 9 | Generate random numbers in the label | When the user answers different question; they won’t be the same one every single time.  Example [10 \* 5]  And  Example [8\*11] |  | | Expected to see two different integers each time | Worked as expected |  |
| 10 | Have four different options for user | User can choose between Multiplication, Subtraction, Addition and division. |  | | Expected for each button the user pressed then the corresponding question will appear | Worked as expected |  |
| 11 | Have a hint button | User can press the hint button to help them. When pressed it will be able to tell what type of question it is such as multiplication, subtraction etc… It will also be able to tell what the question says. So for example if it’s 5\* 5 then the hint button should come with a hint for the 5 times table. | |  | The hint button worked perfectly and came as an info dialog box | Worked as expected |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 13 | Multiplication Button | I can press the multiplication button and the multiplication | **Functional Test** | Expected the button to work perfectly | Worked as expected |  |
| 14 | Submit Button | Tried using the submit button and the answer will be checked |  | Expected the submit button to take all the values and the answer will be checked | Worked as expected |  |
| 15 | Next Question Button | When next question is pressed then a new question will be made |  | Expected the Next Question button to work perfectly and create the question | Worked as expected |  |
| 16 | Back Button | When the back button is pressed then it will take the user back to the menu |  | Expected the back button to take the user back to the menu | Worked as expected |  |
| 17 | Submitting with no answer  submitting | When the user submits without an answer | Robustness Test | The program has validation to stop this; so, the user will be met with an error message telling them to enter something | Worked as expected |  |
| 18 | Switching Questions | When the user switches from one frame to another; for example, the user could be answering a question in the division section however they switch to a multiplication question |  | Expected to change the question with no problem | Worked as expected |  |
| 19 | Entering special characters when | The user enters characters such as: £@a |  | Expected to be met with an error message telling the user to type in an integer | Worked as expected |  |
| 20 | Going back to the menu to reset your score | The user backs out the frame to the menu to reset their score because its bad. |  | The score stays the same | Worked as expected |  |
| 21 | Do as many saves to the program as possible | The user presses save as much as they can | Volume test | The program should not allow the user to keep doing this and you can only save once you’ve answered 10 questions | Worked as expected |  |
| 22 | Spam Next Question as much as possible | The user presses Next Question as much as they can until they get an easy question |  | The program should allow this. | The program allows this however; it adds it as a question answered wrong because the question number goes up however the score does not. |  |

4.2 Test Evidence

****

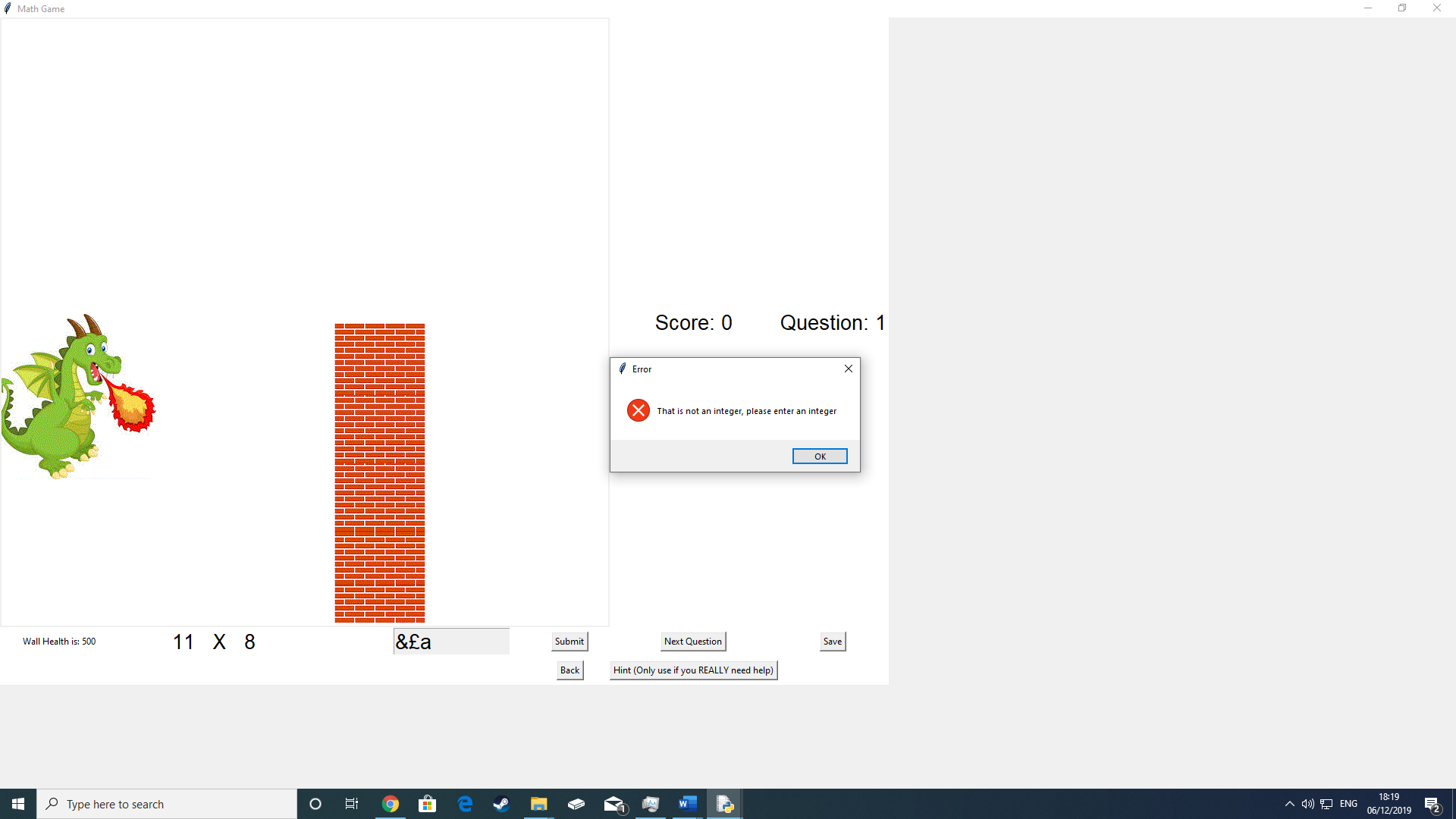
Test No 1. The program recognised that is was a correct answer



Test 9. The program generated random numbers successfully

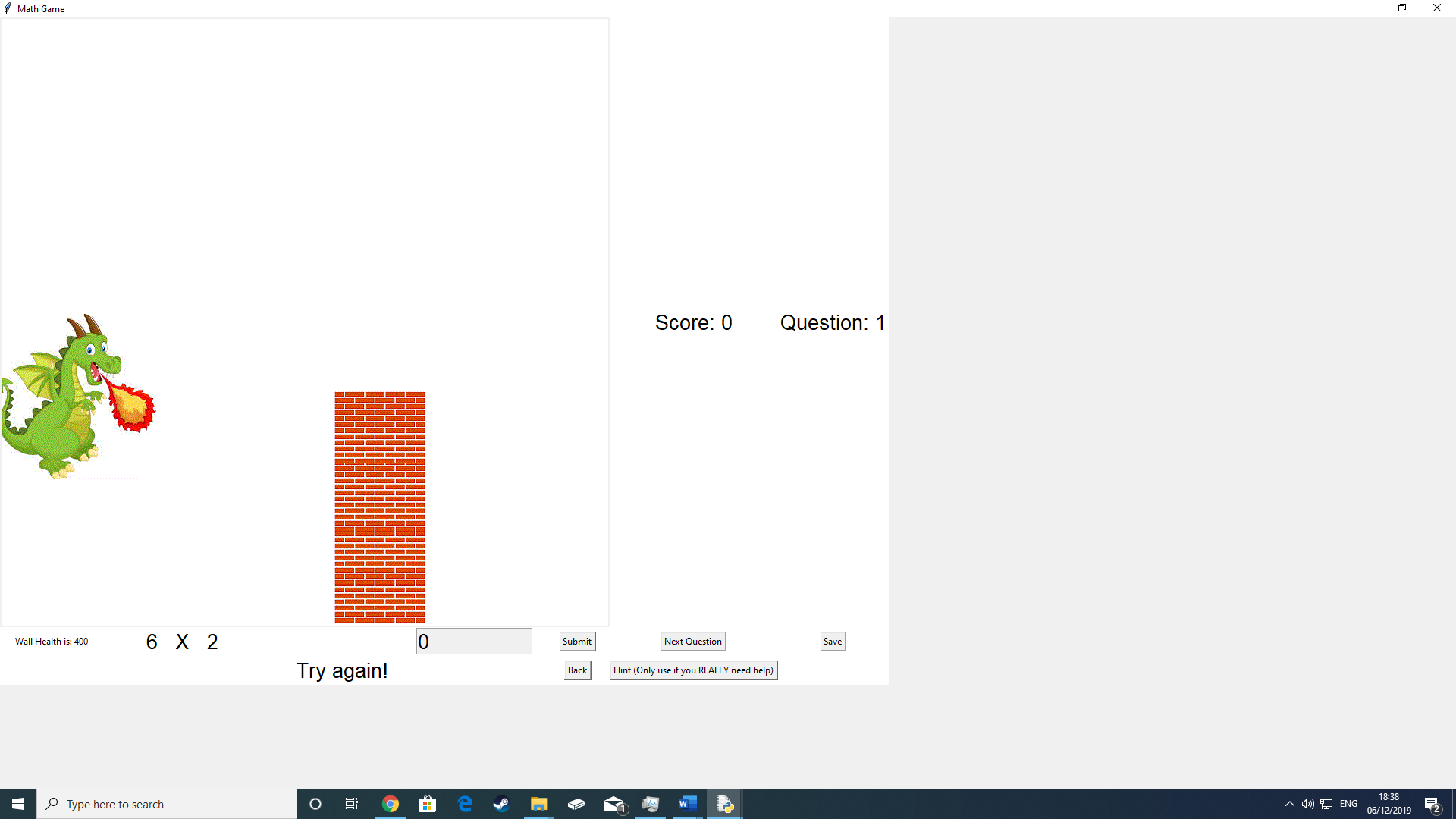
Test No 14. The submit button was pressed and worked to check the user answers correct.

Test No 15. After next question was pressed a new question has been generated

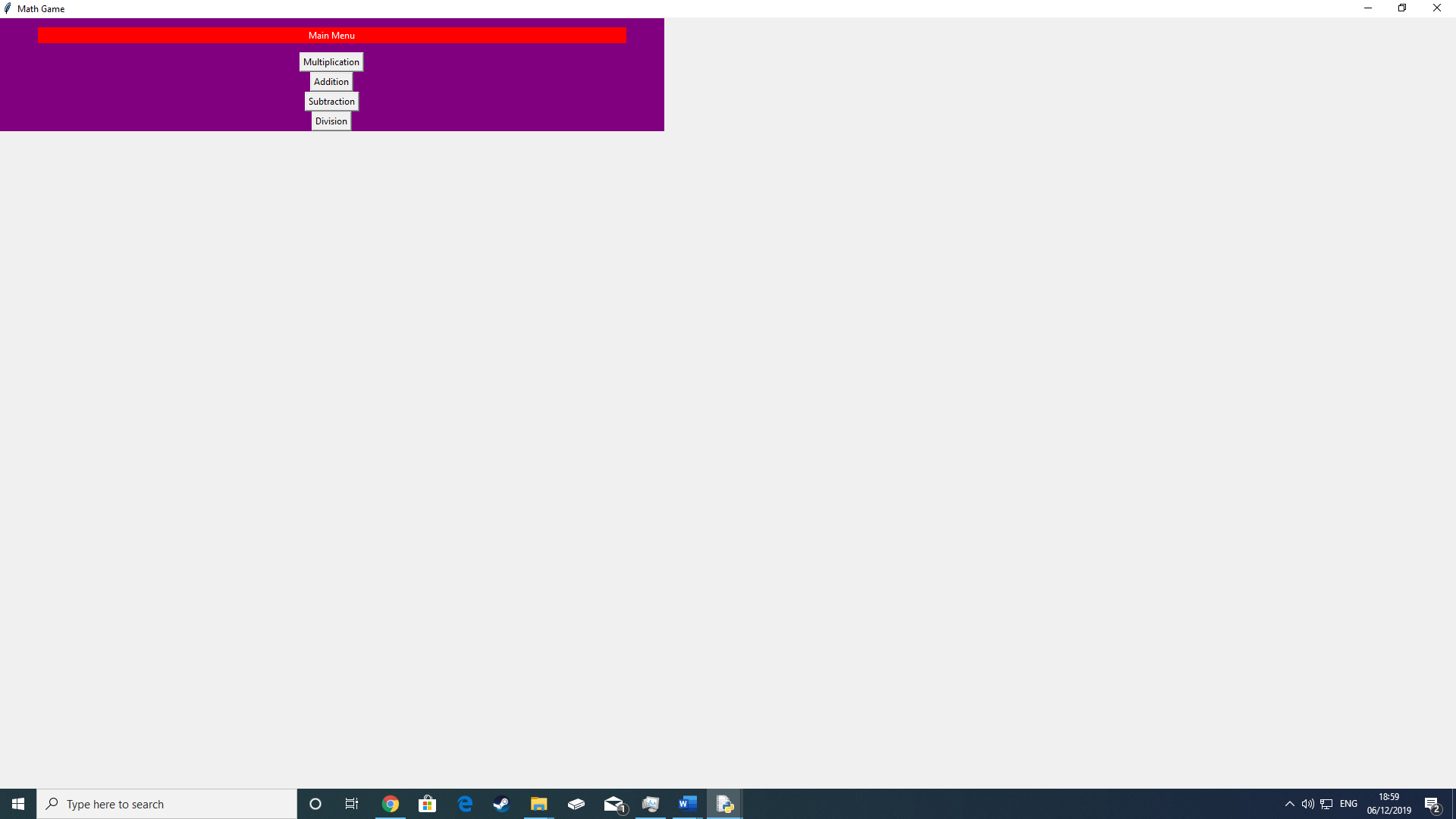
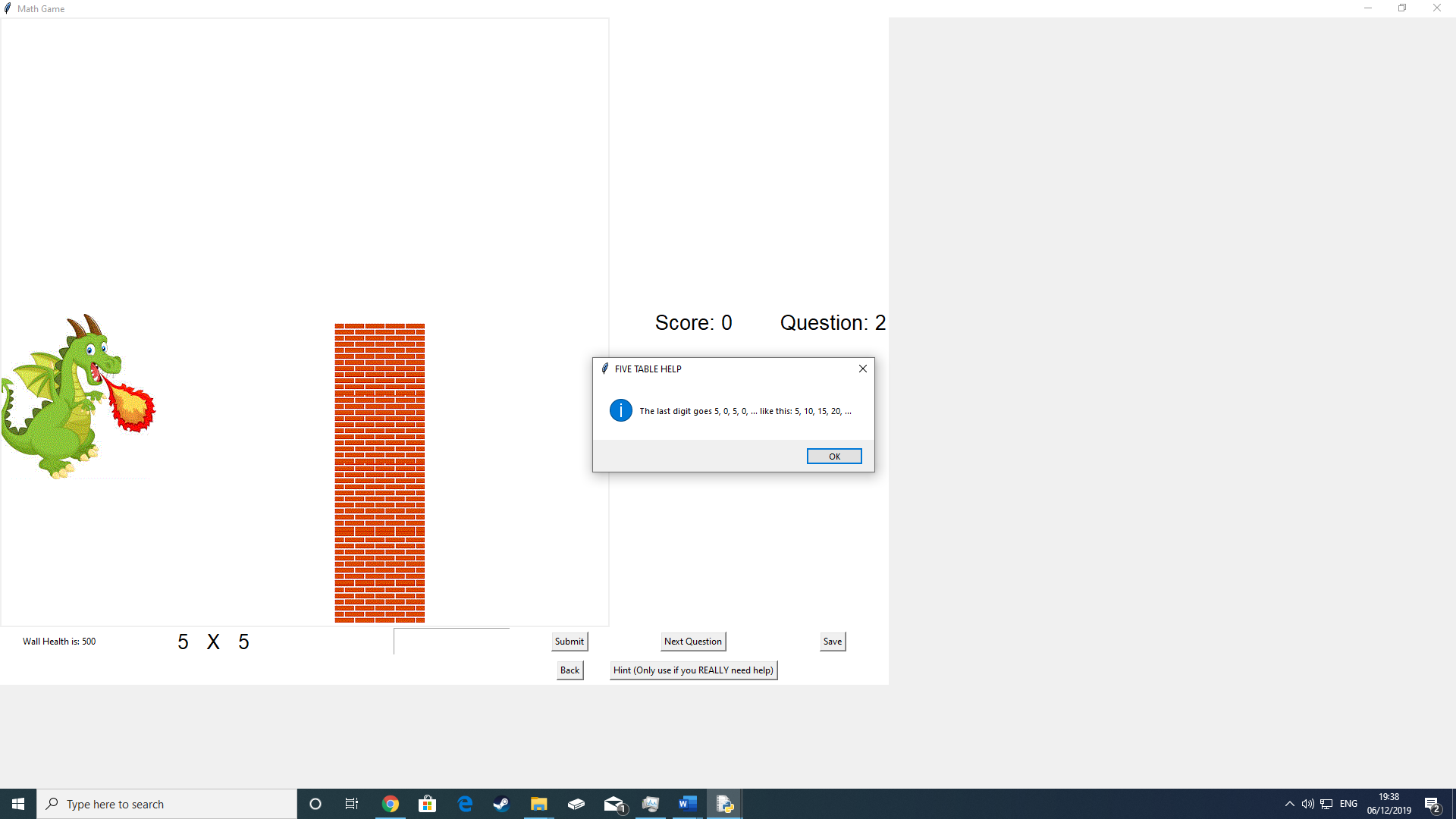


Test 2A. The wall breaks when the user gets a question wrong.

Test No 19. This is the error message that pops up when the user types in special characters.

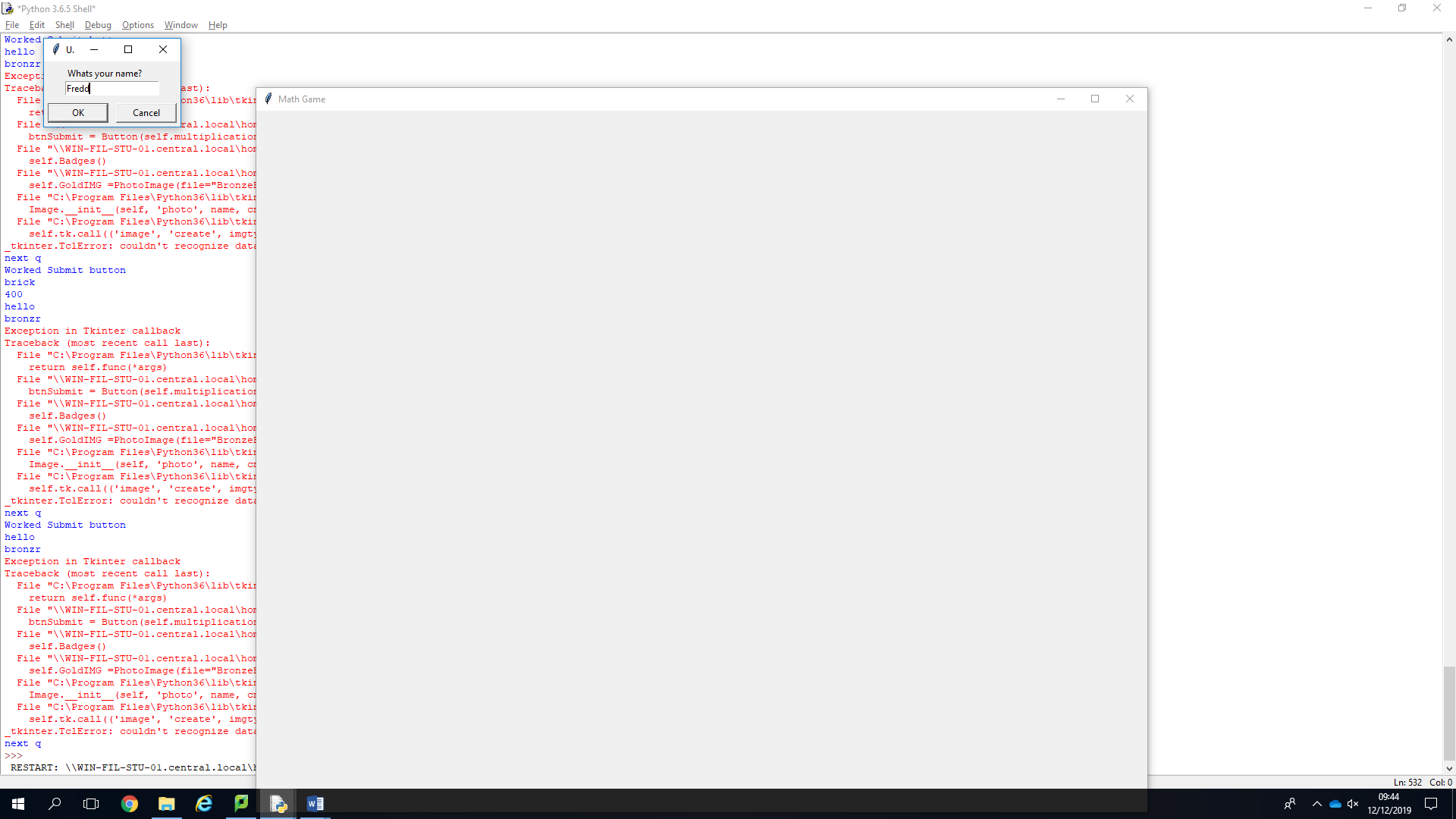
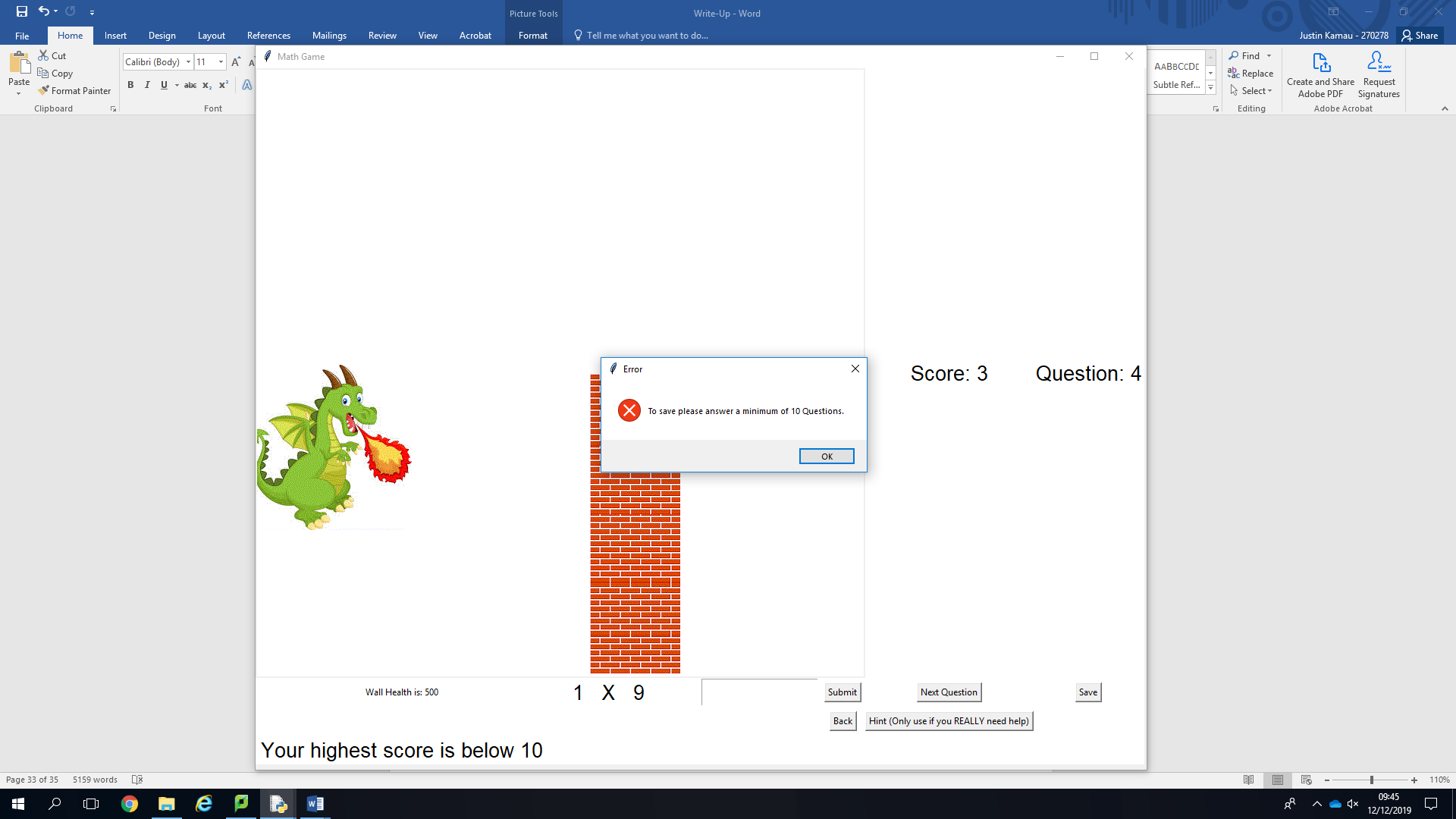


Test 2B. A portion of the wall disappeared after the incorrect answer

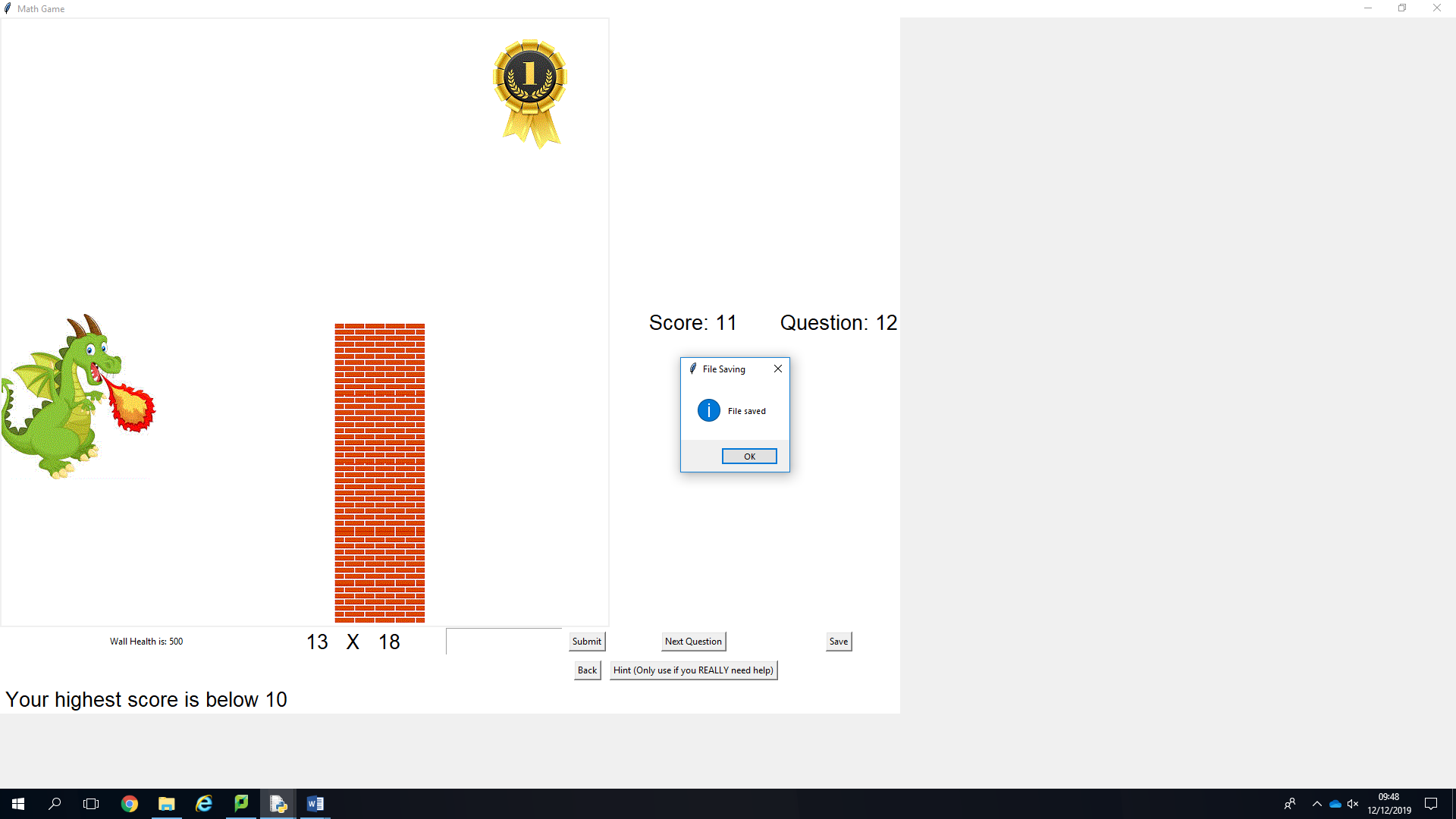


Test No 11.The hint button when pressed has a pop up box giving a helpful hint to the user when they’re stuck

10. This is the menu frame where they get to choose which type of question they want to do.



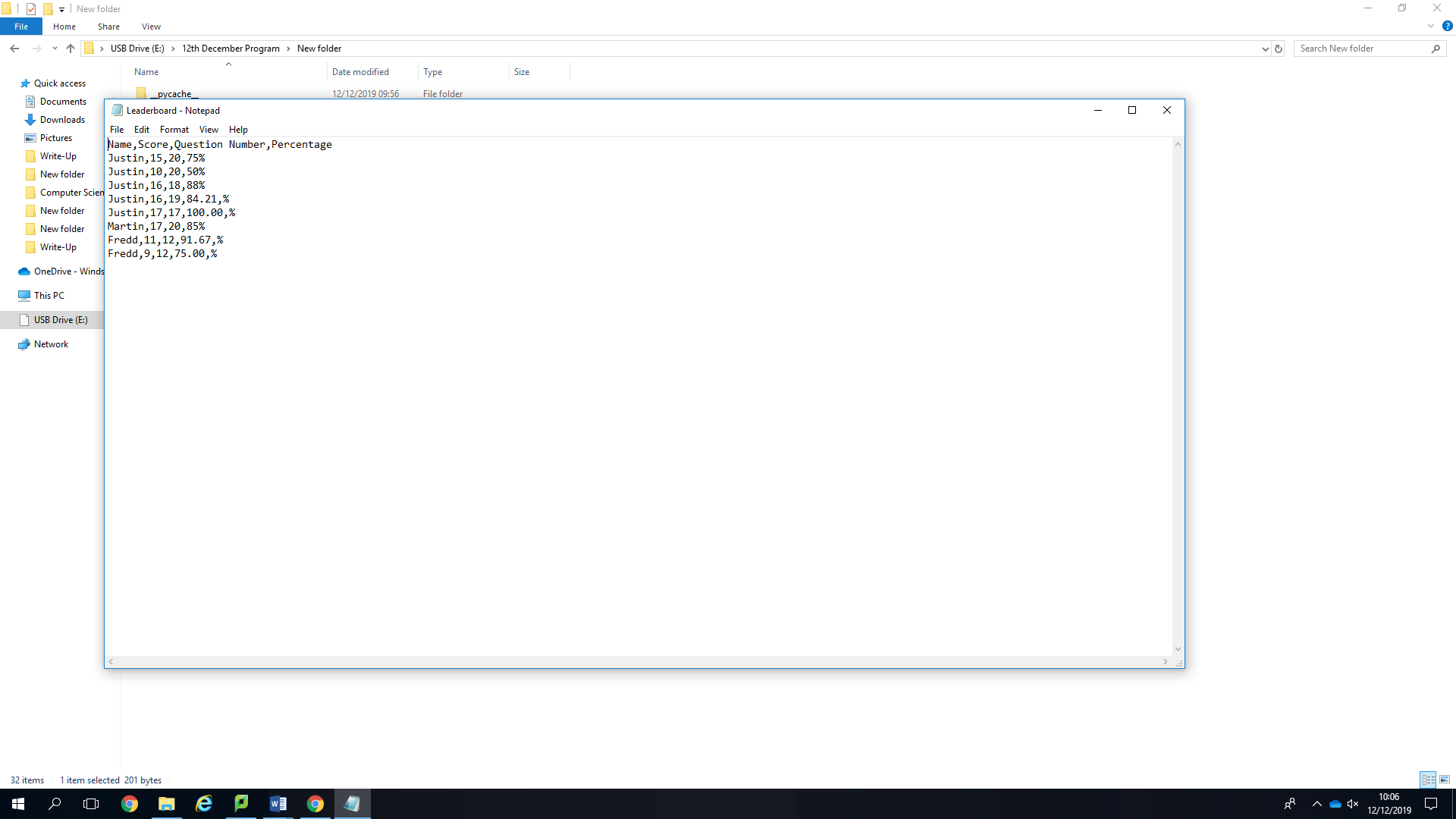
This entry pop up box asks for the users name at the start of the program



Test No 8A. After the users pressed the save button then they are informed that the file is saved. It is then saved as shown here.

Test No 5. The program uses validation to make sure the user has answered a minimum of 10 questions before saving.

Test No 8B.The user’s details and information is saved here.



Test No 7. The percentage is calculated to two decimal places and are all correct.

5.1 Objective Evaluation

1. The program will generate numbers depending on which questions they’re answering; the difficulty will get harder if they get 90% of 10 questions correct:
2. Multiplication will generate random numbers up to 12 however if the difficulty gets harder then it will generate up to 20. Fully achieved with no problems.
3. Subtraction will generate random numbers up to 99 however if the difficulty gets harder then it will generate up to 499 (Also the program has been set so the first number is always bigger than the latter). Fully achieved
4. Addition will generate random numbers up to 99 however if the difficulty gets harder then it will generate up to 499. Fully achived
5. Division will generate random numbers 99 however if the difficulty gets harder then it will generate up to 299 (Also the program will make sure that the question always has an answer of a positive integer). Fully achieved
6. When the program gets harder the user is told by the program with a message informing them. Fully achieved.
7. The program will also have multiplication with the generated numbers moreover it will be up to 20 \* 20
8. In the basic difficulty it will be very easy multiplication so the 1,3,2 ,5 10
9. In the Normal difficulty it will be normal such as 4,6,7,8,9,11,12
10. In the Hard difficulty it will
11. The program will also have division section using the same basis as the multiplication
12. In the basic difficulty, it will only go to 20. For example, 20/5
13. In the normal difficulty, it will only go up to 99. For example, 99/3
14. In the Hard difficulty, it will only go up to 200. For example, 200/50
15. The generated numbers will also be integers because they can’t do decimals yet.
16. The reward system will also be badges so it can appeal to the young audience and give them incentive to work.
17. There will also be a score system to keep track so that the user can see their score
18. I will put in a high scores chart that will work like so:
19. Diamond Badge (Got 90% or above)
20. Gold Badge(80% or above)
21. Silver Badge (60% or above)
22. A permanent storage to store data about the user such as what their score was, how many questions they did and
23. Validation to make sure the user doesn’t type in anything that might crash the program
24. Such as the user typing in a decimal will not work and will be informed of this by an error message
25. The user typing in characters that are not are not integers “@,£,<,&,a,E, Seventeen” and is informed by a different error message.
26. The user typing in nothing will be informed and be informed with an error message.
27. A canvas for the following.
28. The images of the dragon and the wall so the user can see how they’re doing
29. Images of badges
30. Messages that will say “Well done” or “Try again”.
31. An animation for when the user successfully is doing well after 10 questions. The animation will be colourful stars flashing brightly.
32. A hint button to help the user for when they’re struggling at a question. The hint button will recognise each question and be able to give different “hints” to each question
33. 20% of the wall should disappear every time they get a question wrong.
34. If they get two questions correct in a row then they regain any walls they lost
35. Have all 4 different type of questions on one frame (Multiplication, Addition , Subtraction and Division)
36. They can directly switch from one type of question to another and keep their progress
37. As soon as they press submit they can’t change their answer

e