**IT PAT**

**BOBs**

**Dian Naude 27 October 2021**

**Table of contents**

[1 SCENARIO EN REIKWYDTE 5](#_Toc86226136)

[2 VLOEIDIAGRAM 6](#_Toc86226137)

[2.1 Start 6](#_Toc86226138)

[2.2 Game1 7](#_Toc86226139)

[2.3 Game2 8](#_Toc86226140)

[2.4 Game3 9](#_Toc86226141)

[3 DATAWOORDEBOEK 10](#_Toc86226142)

[3.1 Text file 10](#_Toc86226143)

[3.2 Text to speech 10](#_Toc86226144)

[3.3 Functions 10](#_Toc86226145)

[3.3.1 Function1 10](#_Toc86226146)

[3.3.2 Function2 11](#_Toc86226147)

[3.4 Array 11](#_Toc86226148)

[3.5 Media Player 11](#_Toc86226149)

[4 PROGRAMONTWERP 12](#_Toc86226150)

[4.1 Entry 12](#_Toc86226151)

[4.1.1 [1] 12](#_Toc86226152)

[4.1.2 [2] 12](#_Toc86226153)

[4.1.3 [3] 12](#_Toc86226154)

[4.1.4 [4] 13](#_Toc86226155)

[4.1.5 [5] 13](#_Toc86226156)

[4.1.6 [6] 13](#_Toc86226157)

[4.1.7 [7] 13](#_Toc86226158)

[4.1.8 [8] 13](#_Toc86226159)

[4.2 Main 13](#_Toc86226160)

[4.2.1 [1] 14](#_Toc86226161)

[4.2.2 [2] 14](#_Toc86226162)

[4.2.3 [3] 14](#_Toc86226163)

[4.2.4 [4] 14](#_Toc86226164)

[4.2.5 [5] 14](#_Toc86226165)

[4.2.6 [6] 15](#_Toc86226166)

[4.2.7 [7] 15](#_Toc86226167)

[4.3 Game1 15](#_Toc86226168)

[4.3.1 Begin 15](#_Toc86226169)

[4.3.2 Play 16](#_Toc86226170)

[4.3.3 Game2 19](#_Toc86226171)

[4.3.4 Begin 19](#_Toc86226172)

[4.3.5 Play 20](#_Toc86226173)

[4.4 Game3 23](#_Toc86226174)

[4.4.1 Begin 23](#_Toc86226175)

[4.4.2 Play 24](#_Toc86226176)

[5 TVA Tabelle 28](#_Toc86226177)

[6 References 38](#_Toc86226178)

[6.1 Images 38](#_Toc86226179)

[6.2 Mr Long 40](#_Toc86226180)

**List of Figures**

[Figure 1. Bob Start 6](#_Toc86210624)

[Figure 2.Game1 7](#_Toc86210625)

[Figure 3.Game2 8](#_Toc86210626)

[Figure 4.Game3 9](#_Toc86210627)

[Figure 5.Entry 12](#_Toc86210628)

[Figure 6.Main 14](#_Toc86210629)

[Figure 7.Game1Begin 15](#_Toc86210630)

[Figure 8.Game1Play 17](#_Toc86210631)

[Figure 9.Game2Begin 19](#_Toc86210632)

[Figure 10.Game2Play 21](#_Toc86210633)

[Figure 11.Game3Begin 23](#_Toc86210634)

[Figure 12.Game3Play 25](#_Toc86210635)

[Figure 13. TestInput1 28](#_Toc86210636)

[Figure 14. TestInput2 29](#_Toc86210637)

[Figure 15. TestInput3 30](#_Toc86210638)

[Figure 16.TestInput3 31](#_Toc86210639)

# SCENARIO EN REIKWYDTE

Fase1 :

Analise Definieer die taak

Opstel

Deur: Dian Naude Gr 10 K6

Bob’s Zoo

Ek gaan ‘n opvoedkundige speletjie skryf gennaamd Bob’s zoo.In vandag se wêreld spandeer kinders al meer tyd op tegnologiese toestelle waar hulle nuttelose speletjies speel sonder om enige iets te leer. Ek is met die taak gerig om ‘n opvoekundige speletjie te programmeer waar kinders kan speel maar ten selfde tyd nuwe dinge leer.

Bob’s zoo se tema is gefokus op ‘n lewe in die wildtuin.Bob is die eienaar en moet die diere help met dag tot dag take.Vir elke korrekte antwoord sal daar bevordering in die spesifieke taak wees.Die program sal uit 3 speletjies bestaan.

Die eerste speletjie sal fokus op die kind se wiskundige vaardighede.’n Som sal gegee word en die kind sal die antwoord moet insleutel vir bevordering

Die tweede speletjie sal fokus op die kind se spelling.Daar sal ’n text to voice component wees wat woorde/sinne hardop vir die gebruiker sal lees.Hierdie woord/sin sal dan deur die gebruiker in ‘n beperkte tyd in die edit ingesleutel moet word.

Die derde en finale speletjie sal fokus op memorisasie en identifisering van lettertipe.Willekeurige letters sal verkyn vir ‘n beperkte tyd en dan verdwyn.Die gebruiker sal dan die regte woorde weer moet intik.

# VLOEIDIAGRAM

## Start

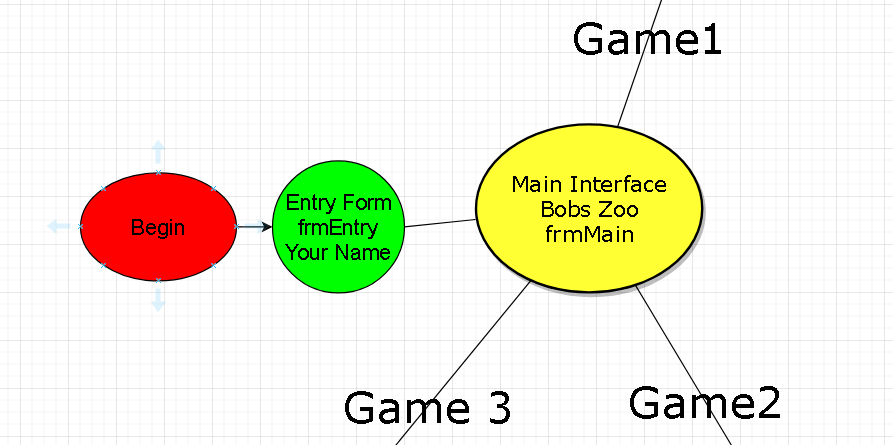


Figure 1. Bob Start

The game Starts and the option of three games are given:

## Game1

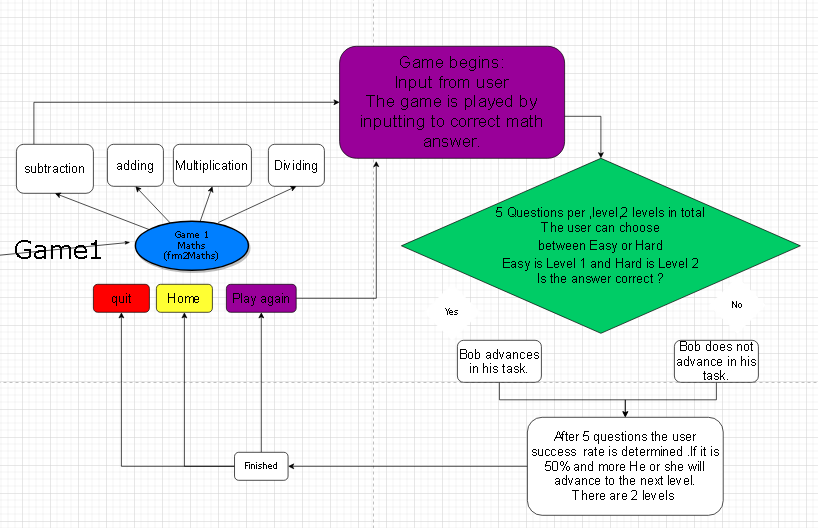


Figure 2.Game1

## Game2

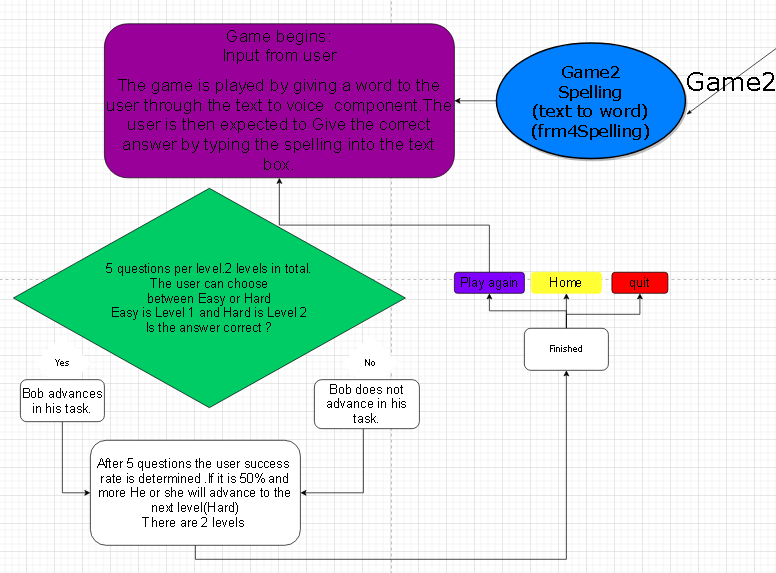


Figure 3.Game2

## Game3

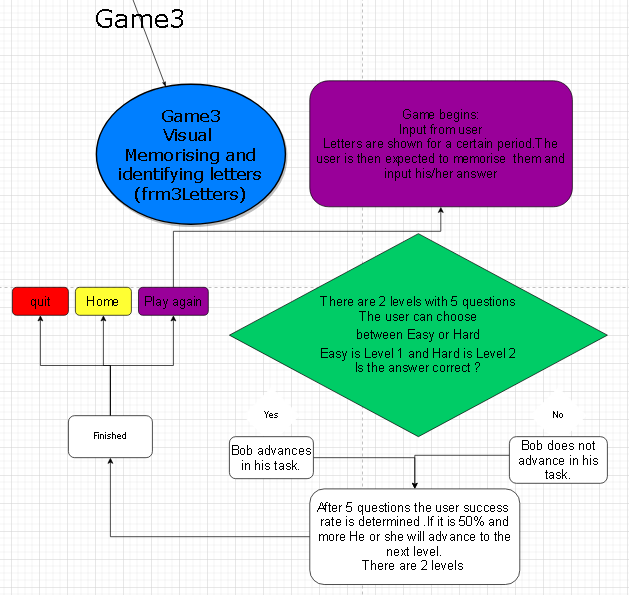


Figure 4.Game3

# DATAWOORDEBOEK

Data dictionary

## Text file

I used my text file in my second game ‘Angus the ape’s spelling game’ .I used it to import words from the text file that is then declared as a string variable .I used the string variable and imported it into the text to speech function which then reads it out loud to the user. After watching Mr. Longs videos I was able to implement the given functions

## Text to speech

I used this in Game 2 as well to convert text to speech and then expect the user to give a input value of the word he or she had heard. I tested this value to see if it was correct and then I showed a message

https://classroom.google.com/u/2/w/MjYzOTE4ODY0MzE1/t/all

https://classroom.google.com/u/2/c/MjYzOTE4ODY0MzE1/m/NDE0NDYzMzM0MDkw/details

Floating division by zero error solved with this code

https://support.medialooks.com/hc/en-us/articles/360000210012--Floating-division-by-zero-error-with-Delphi

## Functions

### Function1

My first function I used in my Math game. The purpose of this function was to give an array of three numbers.1 and 2 always being the sum and 3 being the answer. The input of the function is the level you are at(1 or 2) and if you want to multiply add subtract or divide. Two inputs are given and one array is the output

### Function2

This function is used in game 3 the purpose of the function is to generate random letters .The input is The amount of letters you want(integer) and the output is a string containing 3 or 6 letters depending in which level you are in at present.

## Array

My array is used in Game 1’s Math function to generate 3 numbers that is used in the Math equations

Number 1 is a random number

Number 2 is the second random number

Number 3 is the sum of number one and two .

## Media Player

The component is used to play back round music for the enjoyment of the user. I watched Mr Longs video and was able to implement it into the program.(mpA and mpB)

# PROGRAMONTWERP

## Entry

The user enters his or her name into the edit and it is stored

**Components that are not seen**

MediaPlayerA -Mediaplayer

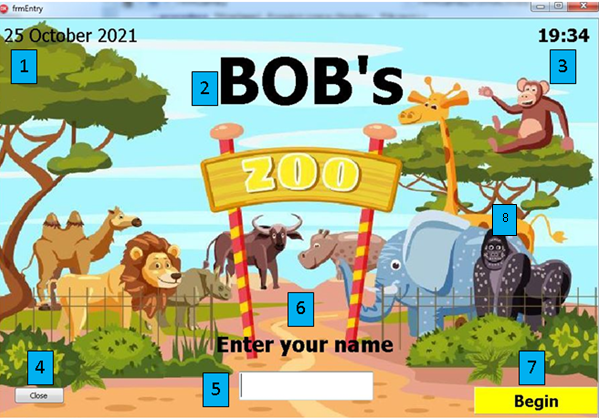


Figure 5.Entry

### [1]

lblDate -Label

### [2]

lblHeading - Label

### [3]

lblTime -Label

### [4]

btnClose-Button

### [5]

edtName-Edit

### [6]

lblName-Label

### [7]

pnlBegin-Panel

### [8]

imgBackround

## Main

The user is given the choice of playing three different games and can choose the game by clicking on the panel with the reference to the game

**Components that are not seen**

MediaPlayerB -Mediaplayer



Figure 6.Main

### [1]

btnCloseX-Button

### [2]

pnlG2-Panel

### [3]

btnBack-Button

### [4]

lblHeadingX-Label

### [5]

imgBackround-Image

### [6]

pnlG3-Panel

### [7]

pnlG4-Panel

## Game1

### Begin

The User is given the opportunity to choose the difficulty of the Game .He/she can then decide whether they want to multiply, divide, add or subtract. This choice will be saved and the user can click on Next



Figure 7.Game1Begin

#### [1]

3 Labels

lblScenarioA1-Label

lblScenarioA2-Label

lblScenarioA3-Label

#### [2]

rgpDifficulty- Radiogroup

#### [3]

rgpInput-Radiogroup

#### [4]

imgJerryA1-Image

#### [5]

imgBobA1-Image

#### [6]

3Grass Images

imgGrassA1-Image

imgGrassA2-Image

imgGrassA3-Image

#### [7]

btnNextA1-Button

#### [8]

bmbHome-Bitmap button

#### [9]

btnClose-Button

### Play

The Math question is given and the user is expected to enter hi/her answer into the edit. A message will then show revealing if it is correct or not .If it is correct the Animation will start. There are 5 questions

**Not seen in Screenshot**

imgLeavesB1 –Image

tmrB1-Timer

tmrB2-Timer



Figure 8.Game1Play

#### [1]

pnlLetsGoB-Panel

#### [2]

imgTreeB1-Image

#### [3]

imgJerryB1-Image

imgJerrybendingB1-Image

#### [4]

imgBobB1-Image

imgbobaxe-Image

#### [5]

lblOutputB1-Label

#### [6]

bmbRetryB1-Label

#### [7]

lblQuestionB1-Label

#### [8]

edtInputAnswerB1-Edit

#### [9]

BtnGoB1-Button

#### [10]

btnLevel2B1-Button

#### [11]

btnBackB1-Button

#### [12]

btnNextB1-Button

#### [13]

bmbHome(same button on Play)-Bitmap button

#### [14]

btnClose-Button

#### [15]

3Grass Images

imgGrassB1-Image

imgGrassB2-Image

imgGrassB3-Image

### Game2

### Begin

The user can choose the difficulty and then click Next to move on

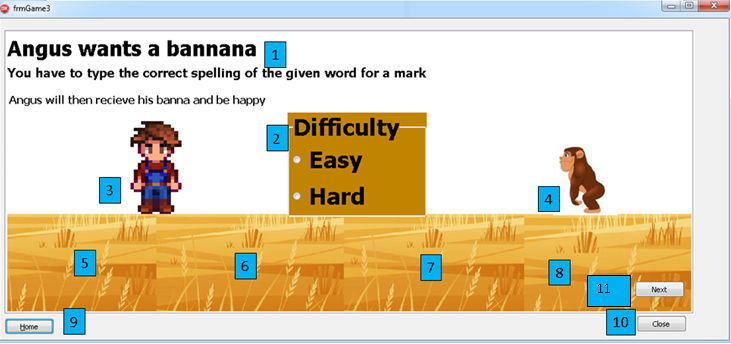


Figure 9.Game2Begin

#### [1]

3labels

lblScenarioA1

lblScenarioA2

lblScenarioA3

#### [2]

rgpDifficultyA1-Radiogroup

#### [3]

imgBob-Image

#### [4]

imgApeA-Image

#### [5]

imgGrassA1-Image

#### [6]

imgGrassA2-Image

#### [7]

imgGrassA3-Image

#### [8]

imgGrassA4-Image

#### [9]

bmbHome-BitMap Button

#### [10]

btnClose-Button

#### [11]

btnNext-Button

### Play

When the user starts the game a sound will play which the user is then expected to give the correct spelling of the word into the edit. A message will then show revealing if it is correct or not .If it is correct the animation will start.

**Can not see components**

tmrB1-Timer

imgBannanaB1-Image

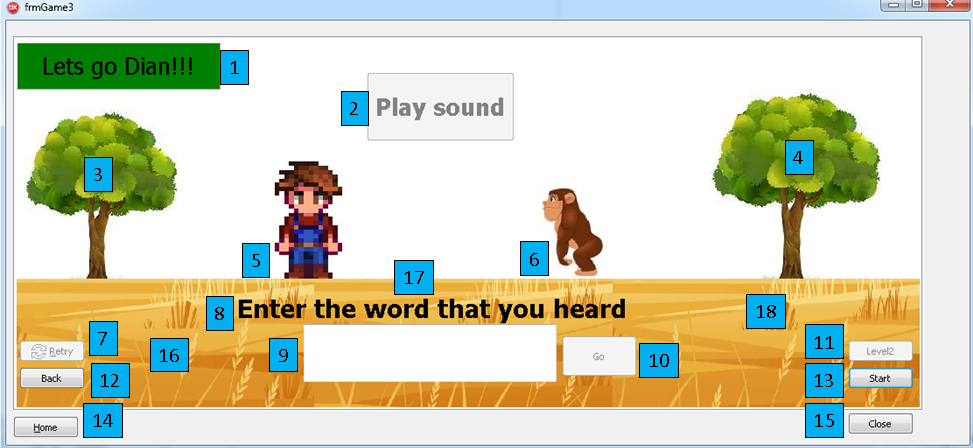


Figure 10.Game2Play

#### [1]

pnlLetsGoB-Panel

#### [2]

btnPaysound1-Button

#### [3]

imgTreeXB1-Image

#### [4]

imgTreeXB2-Image

#### [5]

imgBobB1-Image

ImgBobgivingB1-Image

#### [6]

imgApeB1-Image

imgApewalkingB1-Image

#### [7]

bmbRetry1-BitMap Button

#### [8]

lblQuestionInputB-Label

#### [9]

edtInput-Edit

#### [10]

btnGo1-Button

#### [11]

btnLevelB1-Button

#### [12]

btnBackB1-Button

#### [13]

btnNextA1-Button

#### [14]

bmbHome-BitMap Button

#### [15]

btnClose-Button

#### [16]

imgGrassB1-Image

#### [17]

imgGrassB2-Image

#### [18]

imgGrassB3-Image

## Game3

### Begin

The user can choose the difficulty and then click Next to move on

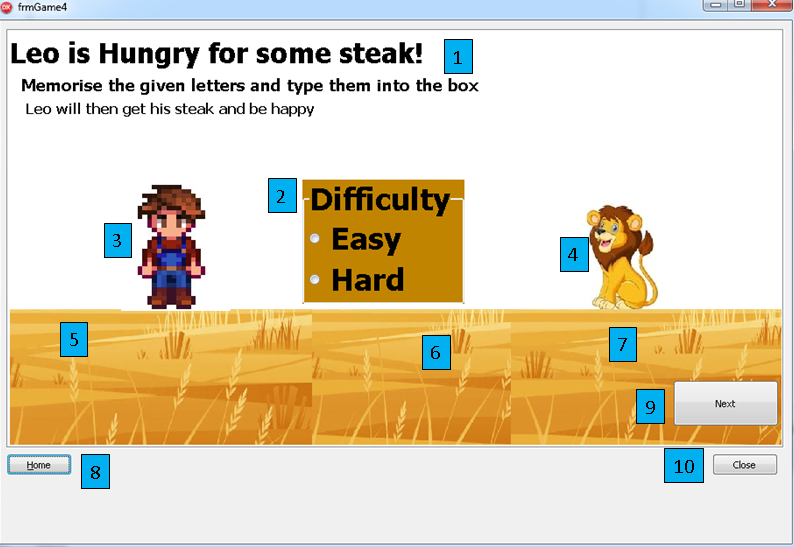


Figure 11.Game3Begin

#### [1]

lblScenarioA1-Label

lblScenarioA2-Label

lblScenarioA3-Label

#### [2]

rgpDifficultyA-Radiogroup

#### [3]

imgBob-Image

#### [4]

imgLion1-Image

#### [5]

imgGrassA1-Image

#### [6]

imgGrassA2-Image

#### [7]

imgGrassA3-Image

#### [8]

bmbHome-BitMap Button

#### [9]

BtnNextA-Button

#### [10]

btnClose-Button

### Play

When the game has stated a 3 or 6 letter word will show and the user is then expected to memorise it and then type it into the given edit. A message will show stating if the answer is correct or not. If the answer is correct the animation will start.

**Components that are not seen**

imgSteak-Image

tmr1-Timer

tmr2-Timer



Figure 12.Game3Play

#### [1]

pnlLetsGoB-panel

#### [2]

imgTreeB1-image

#### [3]

imgBobA1-Image

#### [4]

imgLionA1-Image

imgLionA2-Image

#### [5]

imgTreeB2-Image

#### [6]

btnBack-Button

#### [7]

edtInput-Edit

#### [8]

btnGo1-Button

#### [9]

btnLevel2-Button

#### [10]

bmbRetry-BitMap Button

#### [11]

btnNext-Button

#### [12]

bmbHome-BitMap Button

#### [13]

btnClose-Button

#### [14]

imgGrassB1-Image

#### [15]

imgGrassB2-Image

#### [16]

imgGrassB3-Image

#### 

# TVA Tabelle

|  |  |  |
| --- | --- | --- |
| Input Keyboard and Mouse | Processing | Output |

|  |  |  |
| --- | --- | --- |
| Radio group (rgpDifficulty) is used and  The itemindex is stored and used to determine the difficulty that should be given to the user. Your mouse is used to indicate the index of the radio group. | If rgp.ItemIndex = -1  Then begin  End  Validation by checking that there is a Input | If the itemindex is -1 a message will show indicating that the radiogroup has not been selected |

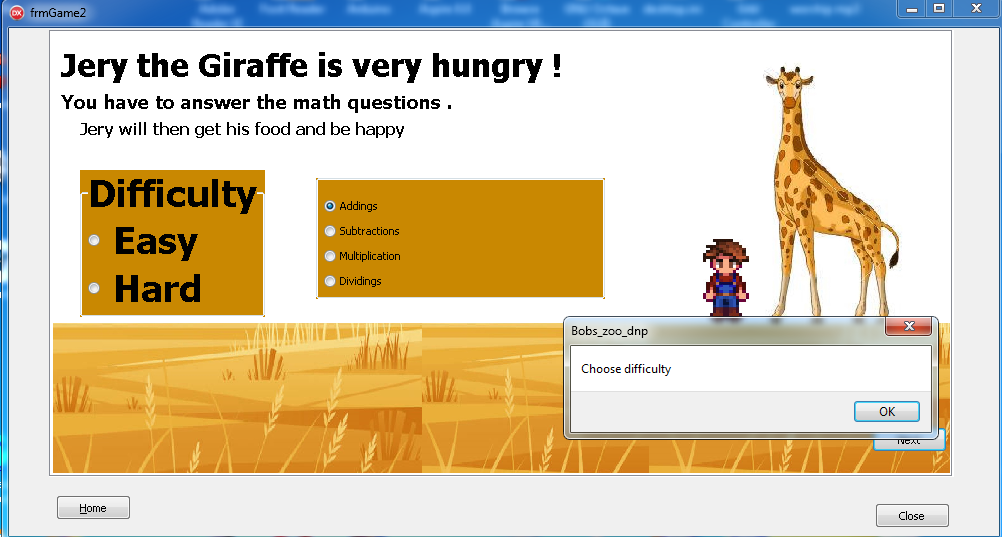


Figure 13. TestInput1

|  |  |  |
| --- | --- | --- |
| Radio group (rgpDifficulty) is used and  The itemindex(Integer) is stored and used to determine whether the user wants to multiply, divide, add or subtract . Your mouse is used to indicate the index of the radio group. | If rgp.ItemIndex = -1  Then begin  End  Valditation by checking that there is a Input  If there is a Input  iInputNum🡨rgpInput.ItemIndex ;  iInputNum is then used in the funtion | If the itemindex is -1 a message will show indicating that the radiogroup has not been selected  else  tbsPlay will show  and tbsBegin will hide |

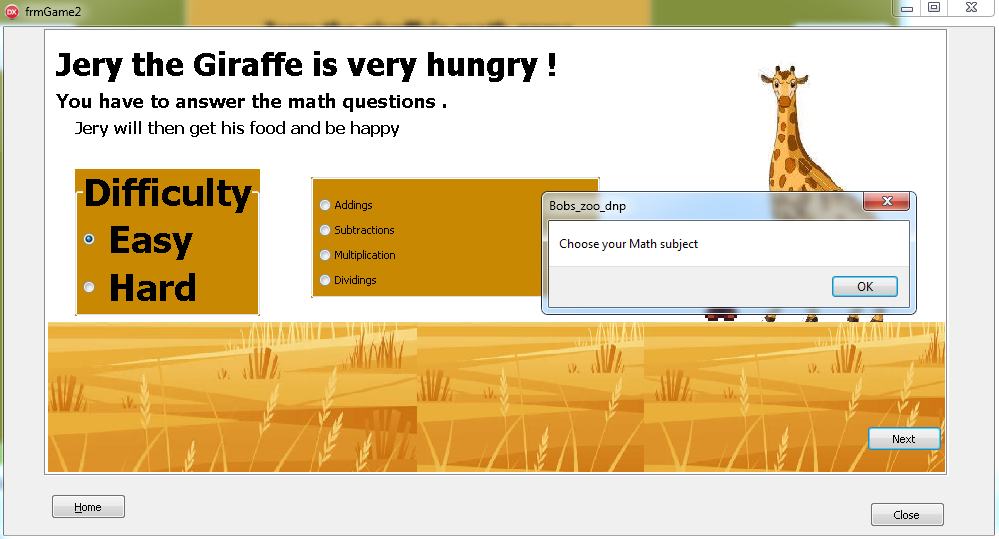


Figure 14. TestInput2

|  |  |  |
| --- | --- | --- |
| The User Enters his or her name(string) This is done with the Keyboard | If there is nothing  A message will show  If there is text the sGivenName will be asigned to it | Showmessage if no text  Else  (sGivenName  has a string value) |

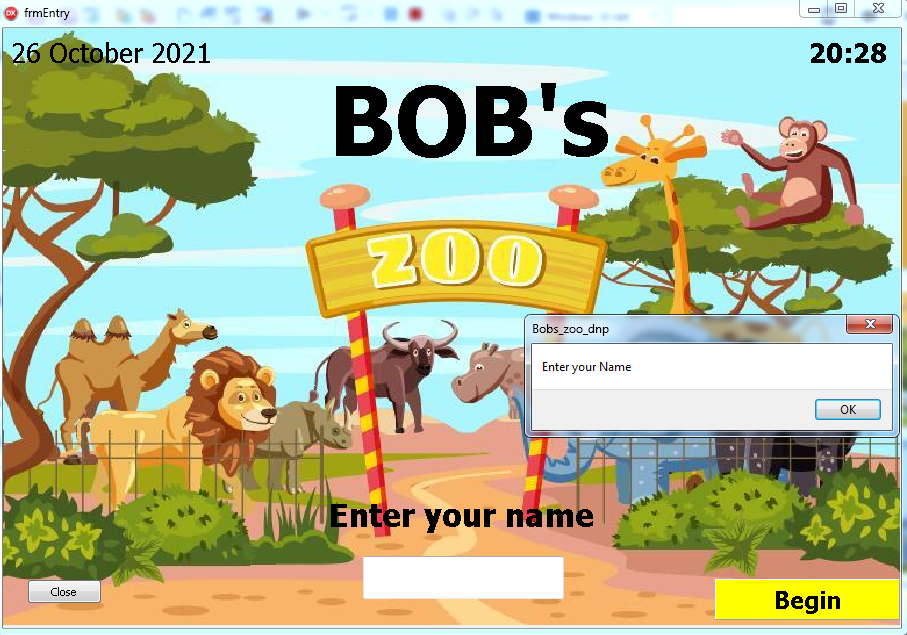


Figure 15. TestInput3

|  |  |  |
| --- | --- | --- |
| The answer of the Math equation is entered as a Input into a edit | If a non integer value is entered  The following commandments will not execute | Message will show ‘’ is not a valid integer value (This is a build in function but still ensures validation to the code) |



Figure 16.TestInput3

|  |  |  |
| --- | --- | --- |
| Input Keyboard and Mouse | Processing | Output |

|  |  |  |
| --- | --- | --- |
| Keybord input is given as the answer of the Math question | The input Integer is compared to the real of the array[3] and if it is the same we know that the correct answer is given to us  The user gives iMathAnsInput which takes the value of the edit  iMathAnsInput 🡨 strtoint(edtInputAnswerB1.Text);  Outputarray[3]; is the answer of the Math equation  rMathAns 🡨 Outputarray[3];  if rMathAns 🡨 iMathAnsInput then  begin  ShowMessage('Correct!');    end  else  begin  ShowMessage('Incorrect'); | ShowMessage('Correct!')  Or  ShowMessage('Incorrect');  (String Output is given) |

|  |  |  |
| --- | --- | --- |
| Input Keyboard and Mouse | Processing | Output |

|  |  |  |
| --- | --- | --- |
| Input from the Keyboard is given as the answer of the spoken word | The input string is compared to the string taken out of the given textFile and if it is the same we know that the correct answer is given to us  The user gives sInputword which takes the value of the edit  sInputword🡨lowercase(edtInput.Text);  if sWord = sInputword then  begin  ShowMessage('Correct!');    end  else  begin  ShowMessage('Incorrect');  When now know if it is correct or incorrect by comparing sWord and sInputword | ShowMessage('Correct!')  Or  ShowMessage('Incorrect');  (String Output is given) |

|  |  |  |
| --- | --- | --- |
| Input Keyboard and Mouse | Processing | Output |

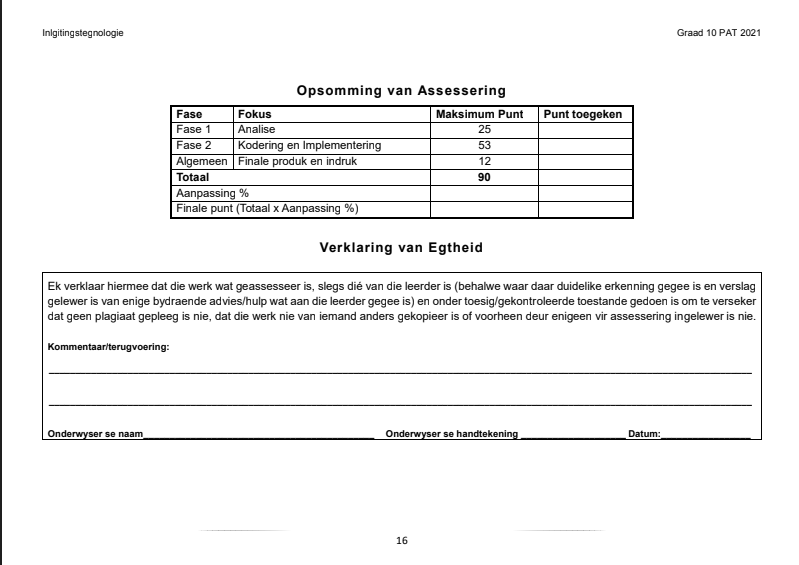
|  |  |  |
| --- | --- | --- |
| Input from the keyboard is given as the answer of the spoken word | The input string is compared to the string taken out of the given Function(Wordgiven) and if it is the same we know that the correct answer is given to us  The user gives sInput which takes the value of the edit  if sInput = sWord then  begin  ShowMessage('Correct!');    end  else  begin  ShowMessage('Incorrect');  When now know if it is correct or incorrect by comparing sWord and sInput | ShowMessage('Correct!')  Or  ShowMessage('Incorrect');  (String Output is given) |

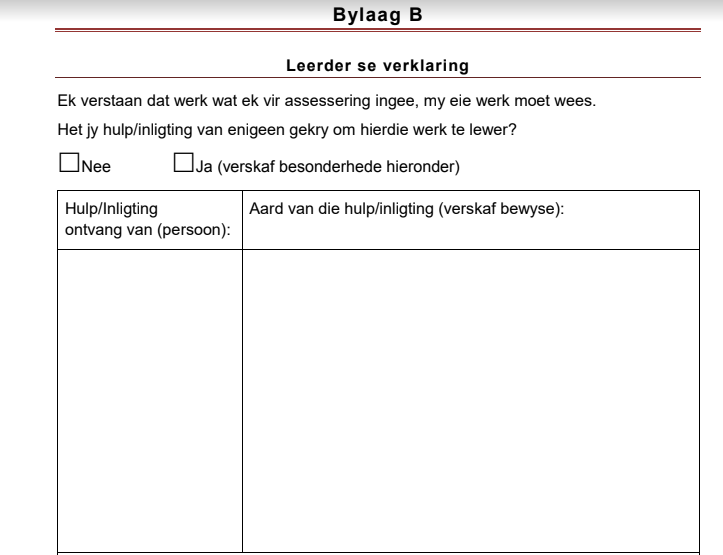
|  |  |  |
| --- | --- | --- |
| Input Keyboard and Mouse | Processing | Output  Speech |

|  |  |  |
| --- | --- | --- |
| Input from the textFile is given | It is then stored in sWord .The function of text to speech is used and we are able to use it at a given time to output text in a different manner  sWord🡨 sLine  ovRead 🡨 CreateOleObject('SAPI.SPVoice') ;  ovRead.speak(sWord) ; | sWord is spoken and heard by the user through the speaker of the computer.It is a sound output |

|  |  |  |
| --- | --- | --- |
| Input | Processing | Output |

|  |  |  |
| --- | --- | --- |
| The user enters his or her name into the given Edit(edtName) | sGivenName🡨edtName.text  sName 🡨 sGivenName | In the panel the caption ‘Lets go’ + sName ‘!!!’ is shown to the user as a output  (String Output is given) |





# References

## Images

1. <https://freedesignfile.com/443411-cartoon-zoo-design-vectors-01/>

2 <https://depositphotos.com/vector-images/animal-sprite.html>

3 <https://graphicriver.net/lion+run-and-lion-graphics>

4 <https://www.123rf.com/photo_61410224_set-of-african-giraffes-separate-images-digital-painting-full-color-cartoon-style-illustration-isola.html>

5 <https://www.123rf.com/photo_59739778_set-of-antelope-images-digital-painting-full-color-cartoon-style-illustration-isolated-on-white-back.html>

<https://www.google.com/search?q=tree+animation&tbm=isch&hl=en&chips=q:tree+animation,g_1:transparent:UXam0O7XOUQ%3D&client=opera&hs=9FP&sa=X&ved=2ahUKEwj6n_PhhebzAhVCYBoKHSTSDo4Q4lYoAHoECAEQEQ&biw=1929&bih=949#imgrc=RDnX7tCmdlE0IM&imgdii=KKqUKnKwzeIC0M>

<https://www.google.com/search?sa=G&hl=en&tbs=simg:CAQS8wEJSkRngbxjV-4a5wELELCMpwgaOgo4CAQSFMMEzSuvCq00ryPcGpkPkDyUOvITGhp9yMjKPl2Dw5D7ExbFaGI-QhmbfMBknzmPQSAFMAQMCxCOrv4IGgoKCAgBEgTNc7GODAsQne3BCRqHAQoWCgR0cmVl2qWI9gMKCggvbS8wN2o3cgofCgxpbGx1c3RyYXRpb27apYj2AwsKCS9tLzAxa3I4ZgoZCgdkcmF3aW5n2qWI9gMKCggvbS8wMmNzZgoUCgNhcnTapYj2AwkKBy9tLzBqancKGwoIY2xpcCBhcnTapYj2AwsKCS9tLzAzZzA5dAw&sxsrf=AOaemvKVpsFSlMAgyyDx1yTgDcp-pFyYMw:1635278507978&q=transparent+background+tree+cartoon+png&tbm=isch&ved=2ahUKEwjcm86U7-jzAhUzQUEAHbI_BEQQwg4oAHoECAEQMg&biw=1950&bih=948&dpr=0.8>

Programming

Delphi

<http://www.delphibasics.co.uk/RTL.asp?Name=function>

**Stackoverflow**

<https://stackoverflow.com/questions/31250419/hide-tabsheets-by-tabsheet-caption-or-tabsheet-name>

<https://stackoverflow.com/questions/16116398/how-to-share-functions-in-delphi>

## Mr Long

1. <https://classroom.google.com/u/2/w/MjYzOTE4ODY0MzE1/t/all>

2 <https://classroom.google.com/u/2/c/MjYzOTE4ODY0MzE1/m/NDE0NDYzMzM0MDk>

3 <https://classroom.google.com/u/0/w/MjYzOTE4ODY0MzE1/t/all>

**Code used exactly**

Floating division by zero error solved with this code

<https://support.medialooks.com/hc/en-us/articles/360000210012--Floating-division-by-zero-error-with-Delphi>

**Father Jan Naude**

I would like to credit my father who taught me a lot during this PAT.

My father helped me to understand arrays and how to use them.

He also showed me how functions work and how one could use them.

I took this insight and was able to write functions and use array in my own work

**Mr Long**

I watched three of Mr Long’s video as can be seen above

To use:

Textfiles

Text to speech

Mediaplayer

**Music**

Zootycoon

<https://www.youtube.com/watch?v=SrDqC9bFAMc&t=5s>

Teacher M. Dunn

Gave me insights to many problems and issues I had during the PAT

