Speed Typing Test Planning:

**Zara’s Games Compendium Plan**

* I will store various short stories in text files
* every time the game is opened the text will change
* I will use the GetFullPath function to pick up the path of the text files.
* There’ll be a text box for the user to copy the text into This will be managed by a random number generator which generators numbers corresponding to those in the file names.
* The user will have 60 seconds controlled by a timer and event which stops input into the user’s text box by setting its properties to read only.
* A score button will be displayed and when it’s clicked the user’s typing will be put into a text file. Then the two files will be spilt into arrays where they will be compared.
* I will be checking this word for word so if one mistake is made everything after that should be wrong.
* I will display the speed of the user and the percentage of completion. This will be calculated using a counter and a for loop which checks each character and adds the correct ones to a list. At the end the elements of the results list will be counted for the score.
* After the score is displayed the user can exit the game and return to the menu with a button click event and then they can choose to replay the game or play a different one.
* Form closing event on click of a button to return to menu. Closes the form and displays the menu form.

Anagrams Planning:

* Display letters in a text box and have a separate text box that is read only (to prevent copy pasting) for the user to type in as many words they can make with those letters.
* Ignore case.
* 60 second timer
* User’s Answers are stored in a list
* Correct Answers stored in a text file which gets converted into a list.
* GetFullPath to get the file path of the correct answer file and adds to a string variable with the file name.
* Distinct function for checking with for loop and counter. Makes sure there’s no double ups of words.
* Comparing it to a list with the correct answers.
* Loop checks each element using if statements check if the correct answers list **Contains** each element of the users list of answers. For each word that corresponds with an element in the correct answers the counter gets incremented to tally up a score.
* Display the score in a label.
* Button to return to menu at any time
* Form closing event on click of a button to return to menu. Closes the form and displays the menu form.

Memory Game:

* Store some famous tongue twisters in a list.
* Random number generator that generates numbers in a specific range. The random number will makeup the elements of the list so that a random phrase is chosen each time,
* Phrases will be displayed in a text box for a few seconds and then the text will be changed to blank.
* Then the user must type the phrase from memory and if its not exactly the same (not including case) then they will lose otherwise they win.
* The user can take their own time to type the phrase and there’ll be button to click when they’re done.
* On the click event the phrase will be displayed so the user can see where they went wrong as well as a label saying either “you win!” or “you lose!”
* Button to return to menu at any time
* Form closing event on click of a button to return to menu. Closes the form and displays the menu form.

Menu Planning:

* Big Title “Zara’s Games Compendium”
* Nice background
* Two categories centred on the landing page. One to exit and one that displays the names of the three games when you hover on it.
* When each game is clicked, they have a click event which opens their own form and hides the menu.

Components

* Each game will have their own form
* And they will be accessed from the menu