Maxwell Maia

21236277

#

Assignment 1

CT2109 OOP

#

12/01/2023

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
HEY! Welcome to the game...
Type the alphabet in order!

Hit enter between letters
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

```
k
[k: Correct! Now type l]
l
[l: Correct! Now type m]
m
[m: Correct! Now type n]
n
[n: Correct! Now type o]
```

```
z
[Correct! Well done!]

Time taken: 9.968seconds

Congratulations!!! :)
```

Problem Statement with Analysis and Design Notes

A game where the player chooses to either recite the alphabet forwards or backwards.

Problem Analysis and Pseudocode

PSEUDOCODE START

//An array of chars will be used. 26 elements big.

declare array of characters

//The player is instructed on how to the play the game.

print message

//Then the player is asked to choose forwards or backwards.

print message

//This requires an input from the player. This has to be either f or b. If neither is input put an error message and ask again.

get input

```
int choice = -1
while(choice == -1)
{
        choice = …
        1 if forwards
        0 if backwards
        -1 if neither is chosen AND print message //invalid input
}

if input is f
{
        Start a timer
        bool playingGame = 1
```

```
        //playing forwards
        while playingGame == 1
        {
                input a character
                if input is equal to next character in array: increment array and print message

                //The game ends once the last character has been correctly chosen
                if i = 26: playingGame = 0
        }
        end timer
}

else if input is b
{
        Start a timer
        bool playingGame = 1

        //playing forwards
        while playingGame == 1
        {
                input a character
                if input is equal to next character in array: decrement array and print message

                //The game ends once the last character has been correctly chosen
                if i = -1: playingGame = 0
        }
        end timer
}
display results


PSEUDOCODE END
```

My design notes

How to loop the array backwards? Start with 25 and – until 0.

Maybe use an if statement and have 2 different loops. One that goes from index 0 -> 25 and One that goes from index 25 -> 0

//Once a direction is chosen the first character (the correct character that the player's input will be checked against) must be set. If forwards i = 0 if backwards i = 25.

Scanner will be used for getting the user input

Extra precaution needs to be taken with the inputs. A user may enter multiple characters. In which case I only want the first letter to be taken. I will use the charAt method for this.

A user may press enter twice, entering an empty string. This may cause problems so a check will be in place to check if the length of the input is 0 using the .length() method.

To make a timer:

Get the current time in milliseconds when the timer is "started".

When the timer is ended: get the current time in milliseconds.

The time elapsed is (end - start).

Convert this value to seconds.

And display it.

The program will run in an infinite loop using a while loop. This allows the player to play again if they choose.

The program will be inside a function so the while loop can easily start the game and restart it when it's over.

Code

```
import java.util.Scanner;
/**
 * Write a description of class GameTest here.
 *
 * @Maxwell Maia
 * @Alphabet Game
 */
public class GameTest
{

    public GameTest()
    {

    }

    public static void main(String[] args)
    {
        GameTest p = new GameTest(); //object to test the game with

        System.out.println("Play? (y/n): ");
        //Infinitely ask the player to play the game
        //Variables to get user input
        String userInputString = "";
        char userInputChar;
        //Scanner for user input
        Scanner inputScanner = new Scanner(System.in);

        //Loop that plays the game if user types 'y'.
        while(true)
        {
```

```
        //Get user input as a String from Scanner
        userInputString = inputScanner.nextLine();
        //Get the first character of the String
        if(userInputString.length() == 0)
        {
            //In case someone presses enter twice
            userInputChar = 's'; //'s' will be counted as invalid. This will count as invalid and the program will continue without crashing
        }
        else
        {
            userInputChar = userInputString.charAt(0);
        }

        //Did the user press f, b or an invalid input?
        if(userInputChar == 'y')
        {
            //Start the game
            p.play();

            System.out.println("\n**************************************************");
            System.out.println("\nPlay again? (y/n): ");
        }
        else if(userInputChar == 'n')
        {
            System.out.println("Ok bye ;)");
            break;
        }
        else
        {
            //invalid
            System.out.println("Invalid. enter only 'y' or 'n'");
```

```
        }
    }

    //p.play(); //// start the game
  }

  public void play()
  {
    //Instructing player

System.out.println("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~");
    System.out.println("HEY! Welcome to the game... \nType the alphabet in order!\n\nHit enter between letters");

System.out.println("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\n***");

    //initilizing char array with alphabet loaded
    char[] alphabet = {'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z'};


    //scanner for user input
    Scanner inputScanner = new Scanner(System.in);

    //Getting the choice of forwards or backwards from user
    //Initilize the variables needed
    int playerChoice = -1;
    String userInputString = "";
    char userInputChar;

    while(playerChoice == -1)
    {
      System.out.println("Hmm, do you want to play forwards or backwards?\n(f/b)\nEnter your answer:");
```

```
            //Get user input as a String from Scanner
            userInputString = inputScanner.nextLine();
            //Get the first character of the String
            if(userInputString.length() == 0)
            {
               //In case someone presses enter twice
               userInputChar = 's'; //'s' will be counted as invalid. This will count as invalid and the program will continue without crashing
            }
            else
            {
               userInputChar = userInputString.charAt(0);
            }

            //Did the user press f, b or an invalid input?
            if(userInputChar == 'f')
            {
               playerChoice = 0;
            }
            else if(userInputChar == 'b')
            {
               playerChoice = 1;
            }
            else
            {
               //invalid
               System.out.println("Invalid. You must enter either 'f' or 'b' to start.");
            }
         }

         int gameInProgress = 0;
         //Gamemode has been chosen.
```

```
        //Time to start the game based on the gamemode
        //note: inputScanner and String + char variables will be reused to save on memory
        if(playerChoice == 0)
        {
            //playing forwards
            System.out.println("\nPlaying forwards");
            System.out.println("************************************************");
            System.out.println("Get ready and have fun :D \nIt's not a competition.\n");
            System.out.println("\nEnter the first letter of the alphabet and press enter to start: ");

            //get the first input
            //Get user input as a String from Scanner
            userInputString = inputScanner.nextLine();
            //Get the first character of the String
            if(userInputString.length() == 0)
            {
                //In case someone presses enter twice
                userInputChar = 's'; //'s' will be counted as invalid. This will count as invalid and
the program will continue without crashing
            }
            else
            {
                userInputChar = userInputString.charAt(0);
            }

            gameInProgress = 0;
            char nextChar = 'G'; //initialized to G to help with debugging
            int i = 0;
            long startTime = 0;
            long elapsedTime = 0;
            //Loop until the user gets the last letter correct
            while(true)
            {
```

```
//Find the value of the next correct character
if(i <= 25)
{
   nextChar = alphabet[i];
}

//Check if input was correct
if(userInputChar == nextChar)
{
   //Display output message saying you were correct
   if(i < 25)
   {
      System.out.println("["+nextChar+": Correct! Now type "+ alphabet[i+1] + "]");
   }
   else
   {
      System.out.println("[Correct! Well done!] \n");
   }


   //Also,if it was the first one he got right,
   //start the timer.
   if(gameInProgress == 0)
   {
      //Start timer
      startTime = System.currentTimeMillis();

      gameInProgress = 1;
   }

   //Increment i only if a correct answer was given
   i++;
```

```
            }

            //Exit the loop if he has gotten the last letter correct
            if(i == 26)
            {
               //Stop timer
               //elapsedTime is currentTime - startTime.
               elapsedTime = System.currentTimeMillis() - startTime;

               gameInProgress = 0;
               break;
            }

            //Get next user input as a String from Scanner
            userInputString = inputScanner.nextLine();
            //Get the first character of the String
            if(userInputString.length() == 0)
            {
               //In case someone presses enter twice
               userInputChar = 's'; //'s' will be counted as invalid. This will count as invalid and
the program will continue without crashing
            }
            else
            {
               userInputChar = userInputString.charAt(0);
            }



         }

         //Display timer result
         System.out.println("Time taken: " + (float)elapsedTime / 1000 +
"seconds\n\nCongratulations!!! :)");
```

```
    }
    else if (playerChoice == 1)
    {
        //playing backwards
        //All the same code, except when the alphabet array is referenced "25 - " is added in the index e.g. alphabet[25 - i]
        //This allows the game to be played backwards.
        System.out.println("\nPlaying backwards");
        System.out.println("**************************************************");
        System.out.println("Get ready and have fun :D \nIt's not a competition.\n");
        System.out.println("\nEnter the last letter of the alphabet and press enter to start: ");

        //get the first input
        //Get user input as a String from Scanner
        userInputString = inputScanner.nextLine();
        //Get the first character of the String
        if(userInputString.length() == 0)
        {
            //In case someone presses enter twice
            userInputChar = 's'; //'s' will be counted as invalid. This will count as invalid and the program will continue without crashing
        }
        else
        {
            userInputChar = userInputString.charAt(0);
        }

        gameInProgress = 0;
        char nextChar = 'G'; //initialized to G to help with debugging
        int i = 0;
        long startTime = 0;
        long elapsedTime = 0;
        //Loop until the user gets the last letter correct
```

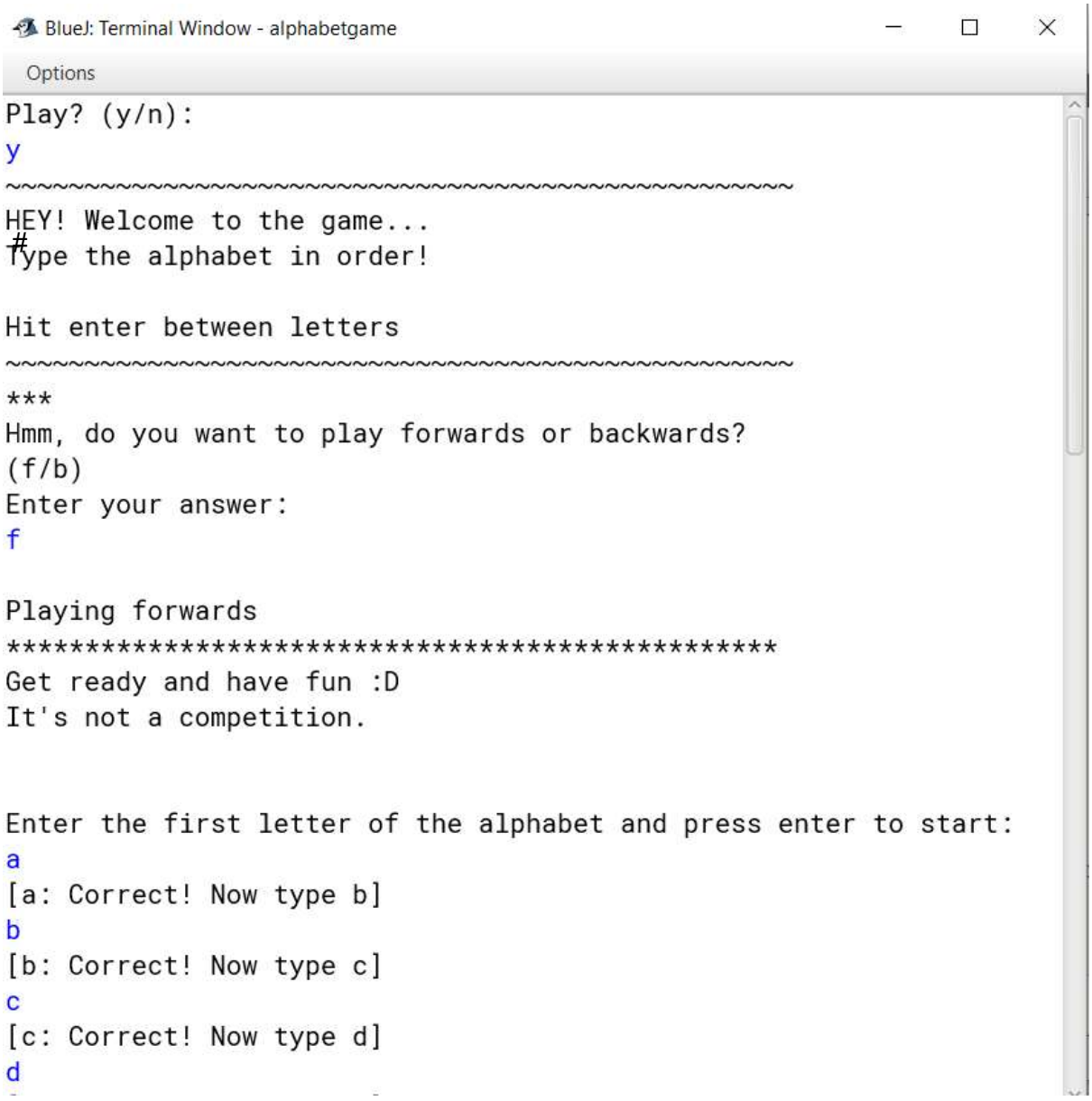
```
        while(true)
        {
            //Find the value of the next correct character
            if(i <= 25)
            {
                nextChar = alphabet[25-i]; //25 - i because we are going backwards
            }

            //Check if input was correct
            if(userInputChar == nextChar)
            {
                //Display output message saying you were correct
                if(i < 25)
                {
                    System.out.println("["+nextChar+": Correct! Now type "+ alphabet[25-i-1] +
"]");
                }
                else
                {
                    System.out.println("[Correct! Well done!] \n");
                }


                //Also,if it was the first one he got right,
                //start the timer.
                if(gameInProgress == 0)
                {
                    //Start timer
                    startTime = System.currentTimeMillis();

                    gameInProgress = 1;
                }
```

```
            //Increment i only if a correct answer was given
            i++;
        }

        //Exit the loop if he has gotten the last letter correct
        if(i == 26)
        {
            //Stop timer
            //elapsedTime is currentTime - startTime.
            elapsedTime = System.currentTimeMillis() - startTime;

            gameInProgress = 0;
            break;
        }

        //Get next user input as a String from Scanner
        userInputString = inputScanner.nextLine();
        //Get the first character of the String
        if(userInputString.length() == 0)
        {
            //In case someone presses enter twice
            userInputChar = 's'; //'s' will be counted as invalid. This will count as invalid and the program will continue without crashing
        }
        else
        {
            userInputChar = userInputString.charAt(0);
        }


    }

    //Display timer result
```

```
            System.out.println("Time taken: " + (float)elapsedTime / 1000 +
"seconds\n\nCongratulations!!! :)");
        }
    }
}
```

Testing

Normal operation



```
[d: Correct! Now type e]
e
[e: Correct! Now type f]
f
[f: Correct! Now type g]
g
[g: Correct! Now type h]
h
[h: Correct! Now type i]
i
[i: Correct! Now type j]
j
[j: Correct! Now type k]
k
[k: Correct! Now type l]
l
[l: Correct! Now type m]
m
[m: Correct! Now type n]
n
[n: Correct! Now type o]
o
[o: Correct! Now type p]
p
[p: Correct! Now type q]
q
[q: Correct! Now type r]
r
[r: Correct! Now type s]
s
[s: Correct! Now type t]
t
[t: Correct! Now type u]
u
[u: Correct! Now type v]
v
[v: Correct! Now type w]
w
[w: Correct! Now type x]
x
[x: Correct! Now type y]
y
[y: Correct! Now type z]
z
[Correct! Well done!]

Time taken: 10.847seconds

Congratulations!!! :)

************************************************

Play again? (y/n):
```

Type input and press Enter to send to program

Invalid inputs

```
Play again? (y/n):
g
Invalid. enter only 'y' or 'n'
g
Invalid. enter only 'y' or 'n'

Invalid. enter only 'y' or 'n'

Invalid. enter only 'y' or 'n'
y
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
HEY! Welcome to the game...
```

Letters other than y and n are correctly detected as invalid.

Pressing Enter is also detected as invalid.

Numbers are also detected as invalid [see picture 4].

When a long string in entered, only the first letter is taken into account. This is intentional. I want it so that if the player types: ‘yz’ at least the ‘y’ will be correct.

```
Hmm, do you want to play forwards or backwards?
(f/b)
Enter your answer:
hmmm, nah I'm okay thanks
Invalid. You must enter either 'f' or 'b' to start.
Hmm, do you want to play forwards or backwards?
(f/b)
Enter your answer:
fabulous

Playing forwards
**********************************************
Get ready and have fun :D
```

These validity checks work for the actual game as well. Only the next character is valid and displays a message when entered correctly.

```
Enter the first letter of the alphabet and press enter to start:
a
[a: Correct! Now type b]
4
twenty
a
b
[b: Correct! Now type c]
cdefg
[c: Correct! Now type d]
```

After you type z, the next letter is not displayed as this would take the array out of bounds.

```
y
[y: Correct! Now type z]
z
[Correct! Well done!]

Time taken: 168.551seconds

Congratulations!!! :)

**********************************
```

Everything works with the backwards gamemode too.

Backwards normal operation

BlueJ: Terminal Window - alphabetgame

Options

```
Play again? (y/n):
y
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
HEY! Welcome to the game...
Type the alphabet in order!

Hit enter between letters
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
***
Hmm, do you want to play forwards or backwards?
(f/b)
Enter your answer:
b

Playing backwards
*************************************************
Get ready and have fun :D
It's not a competition.


Enter the last letter of the alphabet and press enter to start:
z
[z: Correct! Now type y]
y
[y: Correct! Now type x]
x
[x: Correct! Now type w]
w
```

Type input and press Enter to send to program

```
c
[c: Correct! Now type b]
b
[b: Correct! Now type a]
a
[Correct! Well done!]

Time taken: 22.806seconds

Congratulations!!! :)

*************************************************

Play again? (y/n):
```

Type input and press Enter to send to program