code.sprint MT

TASK BOOKLET

- Final Round -

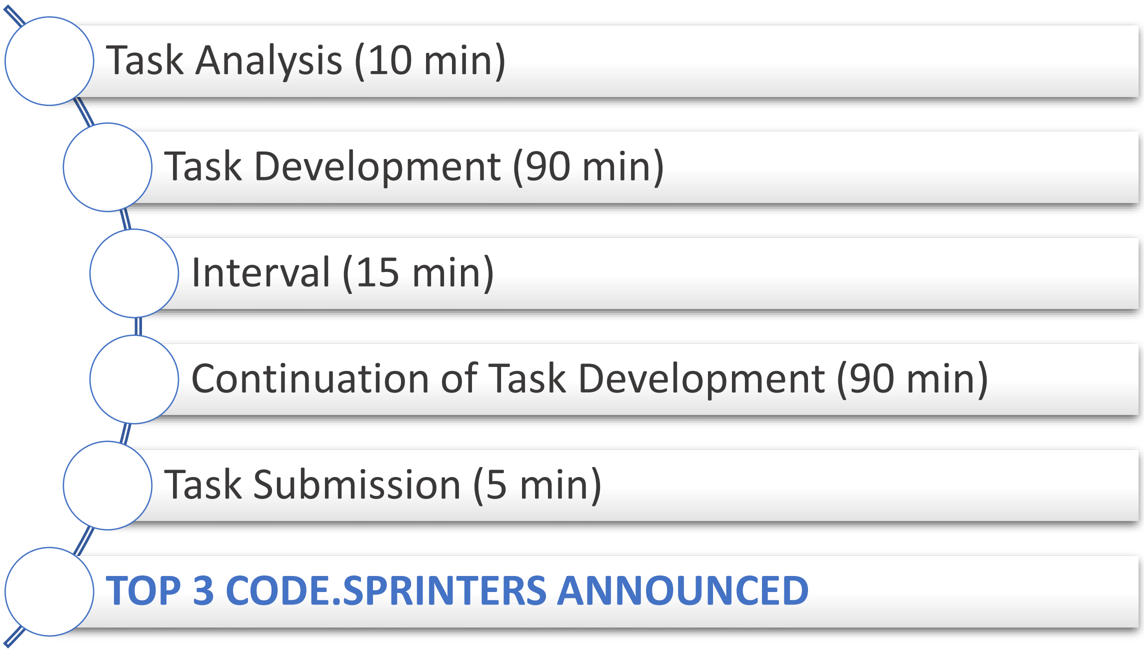
Secondary Category

2023

ICE Malta

GOVERNMENT OF MALTA
MINISTRY FOR EDUCATION, SPORT,
YOUTH, RESEARCH AND INNOVATION
DIRECTORATE FOR LEARNING AND ASSESSMENT PROGRAMMES

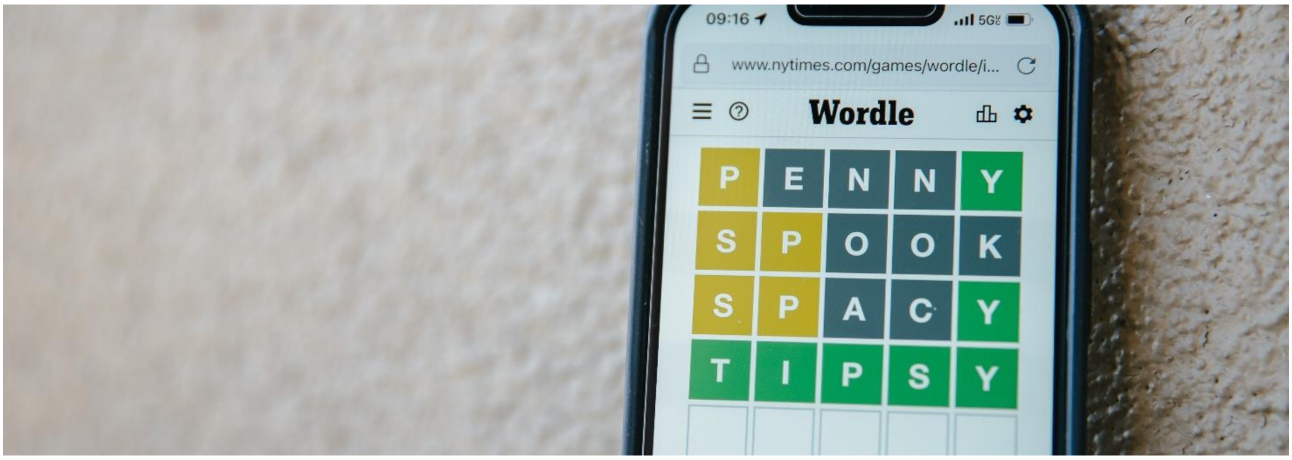
Final Round Schedule



- Task Analysis (10 min)
- Task Development (90 min)
- Interval (15 min)
- Continuation of Task Development (90 min)
- Task Submission (5 min)
- **TOP 3 CODE.SPRINTERS ANNOUNCED**

LingoQuest - Text-based Wordle Game!

This challenge requires the development of a text-based game called "LingoQuest". The game is a variation of the popular Wordle game, where players attempt to guess a five-letter target word within six attempts. You need to implement the core game mechanics, including user input, feedback, and validation.



Functionality #1: Game Play

1. LingoQuest allows the player to play against the ’Computer’, or against another Player. The game should ask the user whether to play in one-player mode or two-player mode or to exit the game.
2. The user should enter [1] to play in one-player mode, [2] to play in two-player mode or [exit] to quit the game.
3. If the player chooses to play in one-player mode, the program should randomly choose a five-letter target word from a list of words. Check table 1 below for the list of words to be hard coded.

apple	**lemon**	**mango**	**dance**	**tiger**
clock	**mouse**	**pizza**	**chair**	**water**

Table 1: List of words

4. If the player chooses to play in two-player mode, the first player should enter the target five-letter word, the screen is then cleared, and the second player should then guess the word.

   Hint: you can use the ” / f ” escape character to clear the screen.

Functionality #2: Game Rules

1. Limit the number of guesses to six chances per game.
2. Provide feedback on each guess using symbols or emojis to indicate correct letters in the correct or wrong positions, or incorrect letters, such as the below:

Emoji or symbol	Description
✅ or ‘ + ‘	letter in the correct position.
🌟 or ‘ ? ‘	letter exists in the target word but not in correct position.
❌ or ‘ – ‘	letter is not present in the target word.

   For example, if the secret word is ‘`lemon`’ and the user enters the word ‘`lingo`’, the feedback might be + – ? – ? or ✅❌🌟❌🌟

   Note: if a player enters a word with two identical letters, and only one of those letters matches a letter in the secret word, prioritize marking the first occurrence and indicate the second occurrence with a '–' or ❌.

   For instance, if the secret word is '`lemon`' and the player enters '`green`', the hint should be – – ? – + or ❌❌🌟❌✅ to signify that there is only one occurrence of the letter 'e'.

3. Provide the player with the option to end the guessing process during gameplay by entering the word 'exit'.
4. Display a proper message when the user guesses the target word.
5. Display a proper message and the target word if the player fails to guess it.

Functionality #3: Validation

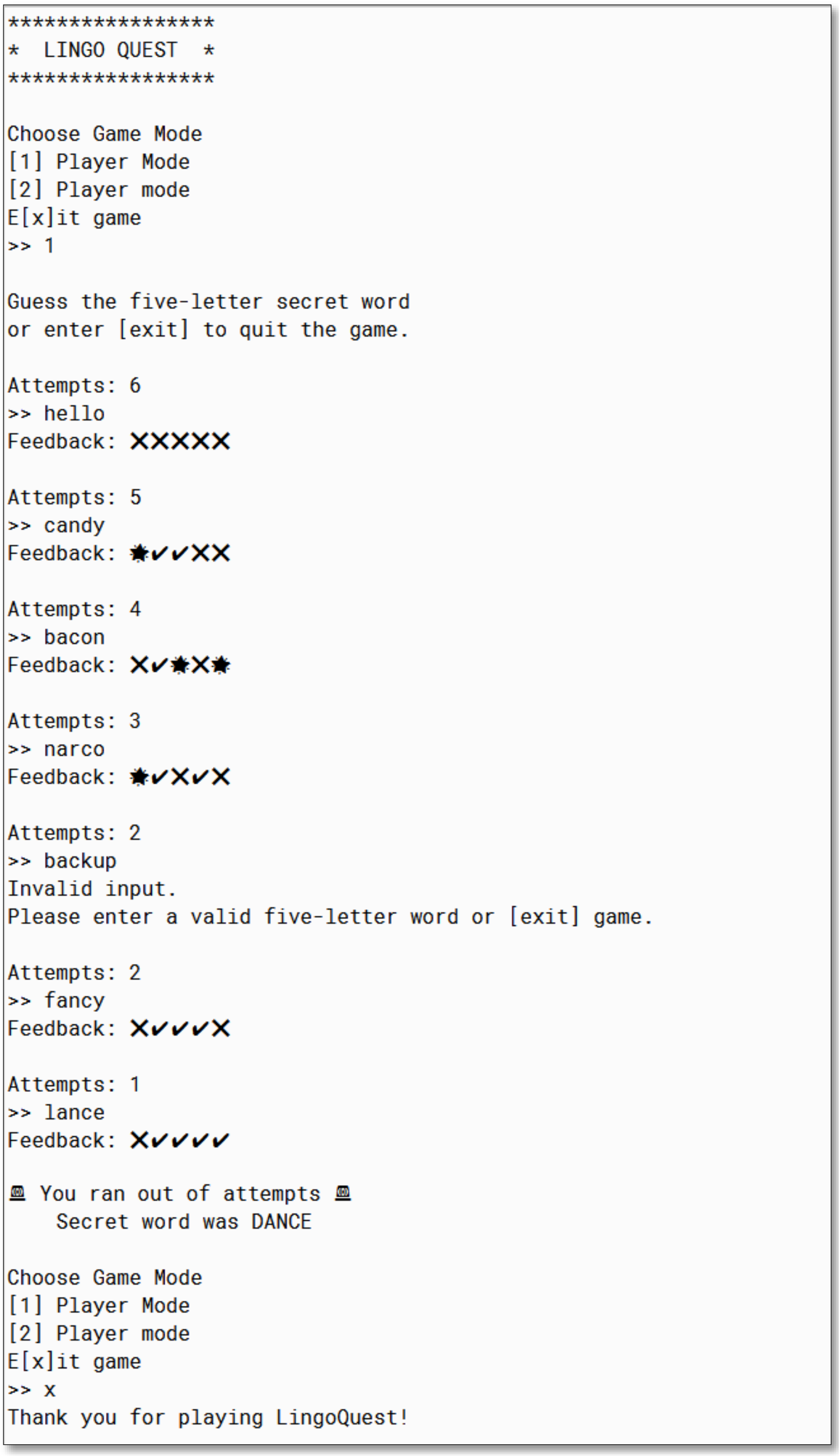
1. Ensure that the user enters [1], [2] or [X] to choose player mode or to quit the game.
2. During gameplay, ensure that the user input is a five-letter word consisting of alphabetical characters only, except when the user enters ‘exit’ to stop guessing the target word.
3. User input is not case sensitive.
4. Validation is required to avoid any possible runtime error.
5. Provide appropriate error messages and allow users to re-enter their input.
6. The player's chances to guess the target word is unaffected by any invalid user input.

Functionality #4: User Interface

Implement a text-based user interface that:

1. allows players to input game mode and their guesses through the command line or console.
2. displays the feedback for each guess after the player submits it.
3. provides an option for the player to play again or exit the game.

**Refer to screenshot 1 below for a comprehensive sample user interface.*



```
*****************
*  LINGO QUEST  *
*****************

Choose Game Mode
[1] Player Mode
[2] Player mode
E[x]it game
>> 1

Guess the five-letter secret word
or enter [exit] to quit the game.

Attempts: 6
>> hello
Feedback: ✘✘✘✘✘

Attempts: 5
>> candy
Feedback: ✸✔✔✘✘

Attempts: 4
>> bacon
Feedback: ✘✔✸✘✸

Attempts: 3
>> narco
Feedback: ✸✔✘✔✘

Attempts: 2
>> backup
Invalid input.
Please enter a valid five-letter word or [exit] game.

Attempts: 2
>> fancy
Feedback: ✘✔✔✔✘

Attempts: 1
>> lance
Feedback: ✘✔✔✔✔

▥ You ran out of attempts ▥
    Secret word was DANCE

Choose Game Mode
[1] Player Mode
[2] Player mode
E[x]it game
>> x
Thank you for playing LingoQuest!
```

Screenshot 1: Sample user-interface

Name the class containing the main method **RunApp.**

Submit your program in a folder named **LingoQuest**

Assessment Rubric

Overall Program Functionality	User-Friendly Interface	Proper use of Comments	Proper Conventions (Camel case, meaningful var names etc.)	Name of Folder & Class/es	User Input	Suitable Prompts / Messages displayed
One or Two player game modes	Random choice of Target Word	Update user attempts remaining	Show the clue code according to rules	Proper use of data structures and/or files	**Maximum Score: 38** **+ 2 for every extra feature.**	
Validations					Modular Code	Code Efficiency
Game mode or Exit	Guesses are five-letter alphabetical words	Invalid guess does not affect attempts	Instruction Case Sensitivity	Avoid Runtime errors		
0 – Not Satisfactorily | 1- Partly Satisfactorily | 2- Entirely Satisfactorily						

eSkills

eSkills Malta Foundation

LeoVegas

me direct