Use Cases

for

Alphabet Tiles

Version 1.0 approved

Prepared by <Batoul Haidar>

<UNIVAQ>

<07/01/2024>

Revision History

Name	Date	Reason For Changes	Version
Batoul Haidar	07/01/2024	Creating Use Cases for Alphabet Tiles project	1.0
Batoul Haidar	10/01/2024	Merging UC2 “Form Word” & UC3 “Calculate Score” in UC2 “Play Round”	1.0
Batoul Haidar	12/01/2024	Expanding in UC2 & Edit end game conditions	1.0

Use Case Template

Use Case ID:	UC1		
Use Case Name:	Start New Game		
Created By:	Batoul Haidar	Last Updated By:	Batoul Haidar
Date Created:	07/01/2024	Date Last Updated:	07/01/2024

Actor:	Player
Description:	The player starts a new game session. Upon initiation, the system automatically and randomly assigns an initial set of tiles to both the player and the system as an opponent..
Preconditions:	The game application is installed, opened, and functioning correctly
Postconditions:	The game board is initialized, and both the player and the system have their initial set of tiles distributed to them. And first word is formed and its score is calculated.
Priority:	High
Frequency of Use:	Each time a player starts a new game
Normal Course of Events:	• The player opens the game application. • The player selects the option to start a new game. • The game prompts the player to choose a difficulty level. The options presented are Easy, Medium, and Hard. • The player selects the desired difficulty level. • The system initializes the game board. • Each of the player and the system will randomly supplied by seven tiles from Tiles Bag. • Who goes first is determined randomly. • The player or computer, who is determined to go first, places the first word on the board. • The game validates that the first word is placed on the center square and adheres to the rules of valid word formation. • If valid, calculate score. Then, gameplay continues with the next turn; if not, the first player re-attempts to place a valid word.
Alternative Courses:	None.
Exceptions:	• If the game fails to start or load correctly, the player is presented with an error message and options to retry or exit.
Includes:	UC3: Calculate Score.
Special Requirements:	• The game interface should be user-friendly and adhere to usability standards for accessibility. • The game must ensure fair and random tile distribution. • The game must have a valid dictionary to check the formed words. • The game's dictionary must be comprehensive and up-to-date to ensure proper validation of words. • The game must have method to randomly determine who goes first.
Assumptions:	The player understands the basic rules of the game, including word formation and the significance of the center square.
Notes and Issues:	

Use Case ID:	UC2		
Use Case Name:	Play Round		
Created By:	Batoul Haidar	Last Updated By:	Batoul Haidar
Date Created:	07/01/2024	Date Last Updated:	12/01/2024

Actor:	Player
Description:	This use case describes the actions of forming a word on the game board by the player or the computer during their respective turns, check validity of the formed word and calculate the score
Preconditions:	A game is in progress, and it is the actor's turn.
Postconditions:	A word has been formed on the board, and the game has validated it and updated the scores accordingly.
Priority:	High
Frequency of Use:	Occurs every turn throughout the game.
Normal Course of Events:	• The system indicates whose turn it is. • Each player in each round should has 7 tiles in his TileRack, so system ensures that player has 7 tiles. • If tile counts in tile rack aren't 7, then tileBag should give tiles to tileRack to be 7. And remove them from the tileBag. • The actor (player or computer) selects tiles from their hand. • The actor places the selected tiles on the board to form a word "new word should be connected to other tiles exists on the board, player can't make new word without connection with other previous tiles". • The actor submits the word for validation. • The game validates the word against the dictionary. • Check if the formed word is valid. • If the word is invalid, the actor is prompted to make a new word or choose another action (if available, such as exchanging tiles or skipping a turn). • If the word is valid "successfully formed on the board", the system calculates and updates the score, as below: ➢ The game system retrieves the value of each tile used in the word. ➢ The system calculates the score based on the tile values, word length, and any special board spaces covered (like double word score). ➢ If any additional scoring rules apply (e.g., bonus points for using all tiles), these are also factored into the calculation. ➢ The total score for the word is computed. ➢ The game system updates the scoreboard with the new score. ➢ The used tiles in forming this word should be removed from TileRack. • TileBag should check tiles count it it. • The turn ends, and play passes to the opponent.
Alternative Courses:	The player may choose to use a hint, skip their turn, or exchange tiles if unable to form a word.
Exceptions:	• The game may not accept the word if it doesn't adhere to the game's rules, or if there's an error in the system's dictionary. • If there's a calculation error or a system error in updating the score.
Includes:	None
Special Requirements:	• The game must have a valid dictionary to check the formed words. • The game's dictionary must be comprehensive and up-to-date to ensure proper

	validation of words. • The scoring mechanism must accurately reflect the game's scoring rules.
Assumptions:	• The actor knows how to form words and is aware of the game's rules regarding word placement. • The scoring system is understood and accepted by the players.
Notes and Issues:	

Use Case ID:	UC3		
Use Case Name:	End game & determine the winner		
Created By:	Batoul Haidar	Last Updated By:	Batoul Haidar
Date Created:	07/01/2024	Date Last Updated:	12/01/2024

Actor:	System
Description:	This use case describes the process of ending the game session and determining the winner based on the scores accumulated by the player and the computer.
Preconditions:	The game is in progress, and an end game condition is met (e.g., all tiles are used, a score limit is reached, etc.).
Postconditions:	The game session is concluded, and the winner is determined and displayed.
Priority:	High
Frequency of Use:	Once at the end of each game.
Normal Course of Events:	• An end game condition is triggered (e.g., all tiles in TileBag have been used, or no more empty squares on the board.). • The game system calculates the final scores of both the player and the computer. • The game system compares the final scores. • The winner is determined based on the highest score. • The game displays the result, showing the winner and the final scores. • The player is presented with options like starting a new game, viewing the scoreboard, or exiting the game. • The game session is concluded, and the player can choose to start a new game, exit, or perform another action.
Alternative Courses:	If the player chooses to exit the game prematurely: 1. The player initiates an action to end the game. 2. The game prompts the player to confirm their decision to exit. 3. If confirmed, the game ends immediately.
Exceptions:	Errors in final score calculation or result display.
Includes:	None
Special Requirements:	The game must have a clear and fair method for determining the winner.
Assumptions:	Player knows the winning conditions and the process to exit the game
Notes and Issues:	