

Software Requirements for Hangman

Hangman

Create By: Jacob Glanz

Date: Nov 11 2023

Overview

This document describes the requirements for the software implementation of the Hangman word-guessing game.

How The Game Works

In the Hangman game implementation, the player starts with a column grid of blank boxes representing letters in a randomly chosen word. Below this grid, there's an alphabet grid (A-Z). The player is guessing the word by clicking on individual letters from the alphabet grid. If the guessed letter is part of the word it fills its corresponding blank box otherwise the player loses a life (out of 5 lives) , after 5 wrong attempts the game is over.

Software Implementation

UI Element

- Start Game Button
- Column Grid of blank boxes representing the word
- Alphabet grid (A-Z) for letter selection
- Label for Game Status

Game Process and Rules

- When the program starts
 - all clicks on the alphabet grid are ignored
 - the letters are Gray/Disabled
 - the game status message is "Click Start to Start Game"
- When the Start button is clicked the
 - clicks on the alphabet grid are not ignored
 - the letters are not Gray/Disabled
 - the game status will display the number of lives that still remain, the game starts with 5 lives.
- When a letter is clicked:
 - If it is in the word, than change this letter's BackColor to Green and show letter in the Word-Column-Grid where it belongs in the word

- If it is not in the word, than change this letter's BackColor to Red and lower the lives by one
- Always when the game ends the letters that were not guessed change to Gray/Disabled
- When the game ends with a win
 - Game status will display "Won"
 - The Word Grid Box BackColor will change to green
- When the game ends with a loss
 - Game status will display "Lost"
 - Fill in the missing letters but mark them red
 - The Word Grid Box BackColor will change to red
- If the user clicks Start in the middle of a game, then the game should interrupt and begin a new game.