```
clc;
clear;
close all;

% initial the game engine
Worlde = simpleGameEngine('Alphabet.png', 16, 16, 5, [255, 255, 255]);

% initial letter color set
green_offset = 0;    % green letter start position
yellow_offset = 30;  % yellow letter
red_offset = 60;     % red letter
default_offset = 27; % white

% initial start menue
startMenuMatrix = default_offset * ones(6, 5);  % creat 6*5 matrix

% set Wordle tag
% show WO with green
startMenuMatrix(1, 1) = green_offset + double('W') - double('A') + 1;
startMenuMatrix(2, 2) = green_offset + double('O') - double('A') + 1;
% RD with yellow
startMenuMatrix(3, 3) = yellow_offset + double('R') - double('A') + 1;
startMenuMatrix(4, 3) = yellow_offset + double('D') - double('A') + 1;
% LE with red
startMenuMatrix(5, 4) = red_offset + double('L') - double('A') + 1;
startMenuMatrix(6, 5) = red_offset + double('E') - double('A') + 1;

% show start menue
drawScene(Worlde, startMenuMatrix);
```

```
title('~Simple Wordle Game---Enter 1 to start game, 0 to finish game~');


% get user's input (start game or not)
while true
    GetInput = getKeyboardInput(Worlde);
    if GetInput == '1'  % start the game
        break;  %jump out of the loop
    elseif GetInput == '0'  % user want to exit
        % using end munue function
        displayEndMenu(Worlde, green_offset, yellow_offset, default_offset, red_offset);
        return;  % finish whole game
    end
end


%main loop to repeat the game
while true
    % read word database
    fileID = fopen('words.txt', 'r');
    wordbase = textscan(fileID, '%s');
    fclose(fileID);
    wordbank = wordbase{1};


    % random a word as the target
    correctWord = upper(wordbank{randi(length(wordbank))});
    fprintf('Target Word (for debugging): %s\n', correctWord); % Test


    % initial the matrix
    maxGuesses = 6;
    guessMatrix = default_offset * ones(maxGuesses, 5); % 6 time to guess
```

```
% main loop's set
gameActive = true;
guessCount = 0;
% show start menue
drawScene(Worlde, guessMatrix);
title('~Simple Wordle Game---Enter 5 letters to guess, or 0 to quit, or 2 to delete the last letter');
while gameActive && guessCount < maxGuesses
    % get user's input
    userInput = '';
    while length(userInput) < 5
        GetInput = getKeyboardInput(Worlde);
        if GetInput == '0'  % check user input 0 or not, if yes, finish game
            gameActive = false;
            break;
        elseif GetInput == '2' && ~isempty(userInput)  % check if user input 2, if yes remove last letter
            % remove the last input letter
            userInput(end) = [];
            % renew the screen of the letter and set the last input
            % letter to white
            guessMatrix(guessCount + 1, length(userInput) + 1) = default_offset;
        elseif ismember(GetInput, 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ')
            % add the input to user's input
            userInput = [userInput, upper(GetInput)];
            % renew the screen of the letter
            guessMatrix(guessCount + 1, length(userInput)) = double(upper(GetInput)) - double('A') + 1;
        end

        % show user's input letter
        drawScene(Worlde, guessMatrix);
        title('Enter 5 letters to guess, or 0 to quit, or 2 to delete the last letter');
```

```
end

% if user want to finish game, then finish this loop
if ~gameActive
    break;
end

% check the user's guess in or not in the word database
if ~any(strcmpi(userInput, wordbank))  % if not a real word
    % change title
    title('!!! This is not a word, please try again !!!');
    userInput = '';  % clear user's input
    guessMatrix(guessCount + 1, :) = default_offset;  % clear the matrix
    drawScene(Worlde, guessMatrix);  % draw again imidiately
    continue;  % keep let user play
end

% guesscount++
guessCount = guessCount + 1;

% check the letter and input it in 'guessMatrix'
% check the letter and input it in 'guessMatrix'
for i = 1:5
    if userInput(i) == correctWord(i)
        % green--- correct
        guessMatrix(guessCount, i) = green_offset + double(userInput(i)) - double('A') + 1;
    elseif contains(correctWord, userInput(i))
        % yellow--- letter appear in this word but not this position
        guessMatrix(guessCount, i) = yellow_offset + double(userInput(i)) - double('A') + 1;
```

```
        else
            % red--- this letter is wrong
            guessMatrix(guessCount, i) = red_offset + double(userInput(i)) - double('A') + 1;
        end
    end

    drawScene(Worlde, guessMatrix);
    title(['Your Guess: ', userInput]);

    % check does user guess out the correct word
    if strcmp(userInput, correctWord)
        title('Congratulations! You guessed the correct word!');
        drawScene(Worlde, guessMatrix);
        title('U Win! Congratulations!  :D');
        pause(3);
        gameActive = false;
    end
end

% adjust fail or not
if guessCount == maxGuesses && ~strcmp(userInput, correctWord)
    disp('Sorry, you have used all your guesses. Game over.');
    drawScene(Worlde, guessMatrix);
    title(['The correct word was: ', correctWord, '. Better luck next time!']);
    pause(3);
end

% ask for user's input, does user want to play again or not
drawScene(Worlde, guessMatrix);
title('Do you want to play again? Enter 1 for YES, 0 for NO');
```

```
    while true
        GetInput = getKeyboardInput(Worlde);
        if GetInput == '1'
            % if input 1 mean keep play
            break; %jump out of this small loop and do main loop again
        elseif GetInput == '0'
            % if input 0 mean finish game show end menue function
            displayEndMenu(Worlde, green_offset, yellow_offset, default_offset, red_offset);
            return; % return nothing to finish loop
        end
    end
end


% End menue function
function displayEndMenu(Worlde, green_offset, yellow_offset, default_offset, red_offset)
    % intial a 5*5 screen for end menu
    endMenuMatrix = default_offset * ones(5, 5);
    % set THANK YOU FOR PLAYING
    % THANK
    endMenuMatrix(1, 1) = green_offset + double('T') - double('A') + 1;
    endMenuMatrix(1, 2) = green_offset + double('H') - double('A') + 1;
    endMenuMatrix(1, 3) = green_offset + double('A') - double('A') + 1;
    endMenuMatrix(1, 4) = green_offset + double('N') - double('A') + 1;
    endMenuMatrix(1, 5) = green_offset + double('K') - double('A') + 1;
    % YOU
    endMenuMatrix(2, 1) = yellow_offset + double('Y') - double('A') + 1;
    endMenuMatrix(2, 2) = yellow_offset + double('O') - double('A') + 1;
    endMenuMatrix(2, 3) = yellow_offset + double('U') - double('A') + 1;
```

```
    % FOR
    endMenuMatrix(3, 1) = red_offset + double('F') - double('A') + 1;
    endMenuMatrix(3, 2) = red_offset + double('O') - double('A') + 1;
    endMenuMatrix(3, 3) = red_offset + double('R') - double('A') + 1;
    % PLAY
    endMenuMatrix(4, 1) = green_offset + double('P') - double('A') + 1;
    endMenuMatrix(4, 2) = green_offset + double('L') - double('A') + 1;
    endMenuMatrix(4, 3) = green_offset + double('A') - double('A') + 1;
    endMenuMatrix(4, 4) = green_offset + double('Y') - double('A') + 1;
    % ING
    endMenuMatrix(5, 1) = yellow_offset + double('I') - double('A') + 1;
    endMenuMatrix(5, 2) = yellow_offset + double('N') - double('A') + 1;
    endMenuMatrix(5, 3) = yellow_offset + double('G') - double('A') + 1;
    % refresh and show
    drawScene(Worlde, endMenuMatrix);
    title('Thank you for playing!');

    % pause 5 second to show
    pause(5);
    close all; %close all
end
```

```
Error using waitforbuttonpress
waitforbuttonpress exit because target figure has been deleted

Error in simpleGameEngine/getKeyboardInput (line 205)
                keydown = waitforbuttonpress;
                          ^^^^^^^^^^^^^^^^^^
Error in Final_code (line 67)
    GetInput = getKeyboardInput(Worlde);
               ^^^^^^^^^^^^^^^^^^^^^^^^
```

Published with MATLAB® R2024b