Wordle

Main Code

```
% WORDLE GAME
clc
clear

% use for play again to manage loop
play = 1;

%import data from spreadsheets
answerWords = readlines('Wordle Christmas.txt');
legalWords = readlines('valid-wordle-words.txt');

% initialize the letters as integers
Q = 11; W = 12; E = 13; R = 14; T = 15; Y = 16; U = 17; I = 18; O = 19; P = 20; %keyboard row 1
A = 21; S =22; D = 23; F = 24; G = 25; H = 26; J = 27; K = 28; L = 29; %keyboard row 2
Z = 30; X = 31; C = 32; V = 33; B = 34; N = 35; M = 36; DEL = 37; ENT = 38; %keyboard row 3

% same with colors
green = 4;
yellow = 5;
gray = 3;
lgray = 2;

% initialize scene specifics and scene
spriteSize = 16;
```

```
zoom = 5;
BGColor = [255,255,240];

wordle_scene = simpleGameEngine("wordleSprites.png",spriteSize,spriteSize,zoom,BGColor);
win_scene = simpleGameEngine("wordleSprites.png",spriteSize,spriteSize,zoom,BGColor);

while(play)
    %initialize the board
    bottomScene = makeBottomBoard();
    topScene = makeTopBoard();
    % draw scene
    drawScene(wordle_scene, bottomScene, topScene)

    % get correct word
    randomIDX = randi(height(answerWords));
    correctWord = lower(char(answerWords(randomIDX)));
    %initialize guess count
    guesses = 0;
    win = 0;

    %execute loop of gettng guesses while the game isnt finished
    while(win == 0 && guesses < 6)

        legalGuess = 0;
        userGuess = '     '; %initial guess filled with spaces
        guessLength = 0;
```

```
%get a  legal guess from the user
while(legalGuess == 0)
    input = 'a'; %initialize input to enter loop
    while(~strcmp(input, 'return'))
        input = getInput(wordle_scene); %keyboard only for now
        %process input (add to array or remove a letter) unless
        if(strcmp(input, 'backspace'))
            if(guessLength > 0)
                userGuess(guessLength) = ' ';
                guessLength = guessLength - 1;
            end
        elseif(strcmp(input, 'return'))
            %do nothing = will exit loop
        else
            %if space is available, add the letter to the user
            %guess array. if space isnt available, do nothing
            if(guessLength < 5)
                guessLength = guessLength + 1;
                userGuess(guessLength) = input;
            end
        end
        %add the status of the guesses and current guess into scene
        %in correct rows
        topScene = addLetter(topScene, guesses, userGuess);
        drawScene(wordle_scene, bottomScene, topScene);
    end
    %initialize user guess as a string
```

```
        userGuessString = convertCharsToStrings(userGuess);
        if(sum(ismember(legalWords, userGuessString)) == 1)
            %if the guess is legal, move onto processing the guess
            legalGuess = 1;

        else

        end
    end
    %process the legal guess (update background scene)
    bottomScene = processSceneGuess(correctWord, userGuess, bottomScene, topScene,
guesses);
    %draw updated scene
    drawScene(wordle_scene, bottomScene, topScene)
    guesses = guesses + 1; %increment guesses
    if(userGuess == correctWord)
        win = 1;
    end

end
pause(.25);

%draw the end scene
if win == 1 %display win screen
    makeEndScene(win_scene, correctWord, 1);
elseif win == 0 %display lose screen
    makeEndScene(win_scene, correctWord, 0);
end
```

```
    % get mouse input
    playInput = 0;
    %get inputs from user until a valid spot is clicked (Y/N)
    while(playInput == 0)
        [r, c] = getMouseInput(win_scene);
        %if the user clicks yes, play again
        if(r == 8 && (c == 5 || c == 6 || c == 7))
            play = 1;
            playInput = 1;
        %if the user clicks no, don't play again
        elseif(r == 8 && (c == 9 || c == 10))
            play = 0;
            playInput = 1;
        end
    end
    close all;
end
```

Making the foreground screen

```
function topBoard = makeTopBoard()
    % initialize the letters as integers
    DEL = 37; ENT = 38; %keyboard row 3

    keyRow1 = [11:20]; %sprite sheet numbers for letter on top row
    keyRow2 = [21:29]; %sprite sheet numbers for letter on middle row
    keyRow3 = [30:36]; %sprite sheet numbers for letter on bottom row
```

```
    topBoard = ones(13);
    topBoard(9,2:11) = keyRow1; %make 1st row keyboard layout
    topBoard(10,3:11) = keyRow2; %make 2nd row keyboard layout
    topBoard(11,3:11) = [ENT, keyRow3, DEL]; %amke 3rd row keyboard layout
end\
```

Making the background screen

```
function botBoard = makeBottomBoard()
    % initialize the letters as integers
    W = 12; O = 19; R = 14; E = 13; D = 23; L = 29;%initialize letters in the title as numbers

    wordleTitle = [W, O, R, D, L, E];

    botBoard = ones(13); %initialize a 13x13 matrix for the game screen
    botBoard(1, 5:10) = wordleTitle;

    botBoard(2:7,5:9) = 6; %change center to wordle board

    botBoard(9,2:11) = 2; %make 1st row keyboard layout
    botBoard(10,3:11) = 2; %make 2nd row keyboard layout
    botBoard(11,3:11) = 2; %make 3rd row keyboard layout
end
```

Processing keyboard input

```
function input = getInput(scene)
    %make valid inputs to check for
    validInputs = ["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", "return", "backspace"];
    input = ','; %invalid input to enter loop
    while(~ismember(convertCharsToStrings(input), validInputs))
        input = getKeyboardInput(scene);
    end
end
```

Adding letters to the screen

```
function newTopScene = addLetter(topScene, guesses, userGuess)
    gameBoard = topScene(2:7, 5:9); %initalize game board
    %updates the game board with the status of the current guess
    for(i = 1:length(userGuess))
        %sets the current row of the game board equal to the current status
        %of the users guess
        gameBoard(guesses+1, i) = lettertoNum(userGuess(i));
    end
    topScene(2:7,5:9) = gameBoard; %update top scene
    newTopScene = topScene; %return different variable than the input
end
```

Translating letters to numbers for sprites

```
function num = lettertoNum(letter)
    Q = 11; W = 12; E = 13; R = 14; T = 15; Y = 16; U = 17; I = 18; O = 19; P = 20; %keyboard
row 1
    A = 21; S =22; D = 23; F = 24; G = 25; H = 26; J = 27; K = 28; L = 29; %keyboard row 2
    Z = 30; X = 31; C = 32; V = 33; B = 34; N = 35; M = 36; %keyboard row 3
    num = 1; %have to initialize to get it to work
    %match character with corresponding number value
    if(letter == 'a')
        num = A;
```

```
elseif(letter == 'b')
    num = B;
elseif(letter == 'c')
    num = C;
elseif(letter == 'd')
    num = D;
elseif(letter == 'e')
    num = E;
elseif(letter == 'f')
    num = F;
elseif(letter == 'g')
    num = G;
elseif(letter == 'h')
    num = H;
elseif(letter == 'i')
    num = I;
elseif(letter == 'j')
    num = J;
elseif(letter == 'k')
    num = K;
elseif(letter == 'l')
    num = L;
elseif(letter == 'm')
    num = M;
elseif(letter == 'n')
    num = N;
elseif(letter == 'o')
```

```
        num = O;
    elseif(letter == 'p')
        num = P;
    elseif(letter == 'q')
        num = Q;
    elseif(letter == 'r')
        num = R;
    elseif(letter == 's')
        num = S;
    elseif(letter == 't')
        num = T;
    elseif(letter == 'u')
        num = U;
    elseif(letter == 'v')
        num = V;
    elseif(letter == 'w')
        num = W;
    elseif(letter == 'x')
        num = X;
    elseif(letter == 'y')
        num = Y;
    elseif(letter == 'z')
        num = Z;
    end
end
```

Processing guesses

```
function newBottomScene = processSceneGuess(correctWord, userGuess, bottomScene,
topScene, guesses)
```

```
%initialize background colors
green = 4;
yellow = 5;
gray = 3;

bottomKeyboard = bottomScene(9:11, 2:11); %change the bottom of the game board
topKeyboard = topScene(9:11, 2:11); %used to find what letter is what

bottomGameBoard = bottomScene(2:7, 5:9); %change the bottom of the game board

%convert correct word string to char array
correctNums = [1:5]; %initialize for speed
guessNums = [1:5]; %initialize for speed

%generate number arrays for sprites
for(i = 1:length(userGuess))
   correctNums(i) = lettertoNum(correctWord(i));
   guessNums(i) = lettertoNum(userGuess(i));
end

%current letter in the guess that is being processed
currentPos = 1;

%initialize current row to adjust guesses
%for each letter, check if it's the right letter for that position and if it is, make the background at that spot green. If
%its not, check if its in the word and make background yellow.
%Otherwise make the background gray
```

```
for(i = 1:length(userGuess))
    matchingIdxs = ismember(correctWord, userGuess(i));
    if(guessNums(i) == correctNums(i))
        %make all the letters on keyboard that are correct green
        bottomKeyboard(find(topKeyboard == guessNums(i))) = green;
        %same with the current row on game board
        bottomGameBoard(guesses+1, i) = green;
    elseif(sum(matchingIdxs) > 0)
        %if its in the word, make yellow
        found = 0; %the amount letters in the guess that would be green
        foundYellow = 0; %the amount of letters in the guess that would be yellow
        %for the letter, update the amount of potentia greens and yellows for
        %the letter
        for(j = 1:length(userGuess))
            if(userGuess(i) == userGuess(j) && userGuess(j) == correctWord(j))
                found = found + 1;
                %if there are already greens in the word, update found
            elseif(userGuess(i) == userGuess(j) && sum(matchingIdxs) > 0)
                foundYellow = foundYellow + 1;
                %update the amount of potential yellows
            end
        end

        %if there is more than one yellow, make the first one yellow
        %and the other gray
        if(foundYellow == 2)
            %check if the yellow has already occured
```

```
        alreadyFound = 0;
        for(k = 1:currentPos)
            if(userGuess(k) == userGuess(i) && bottomGameBoard(guesses+1, k) == yellow)
                %if the yellow has already occurred, say it has been found
                alreadyFound = 1;
            end
        end
        %if the yellow hasnt been found, make the square yellow
        if(alreadyFound == 0)
            bottomGameBoard(guesses+1, i) = yellow;
        %if it has, make the current square gray
        else
            bottomGameBoard(guesses+1, i) = gray;
        end
    elseif(found < sum(matchingIdxs))
        bottomGameBoard(guesses+1, i) = yellow;
    else
        bottomGameBoard(guesses+1, i) = gray;
    end
    if(bottomKeyboard(find(topKeyboard == guessNums(i))) ~= green)
        %if the keyboard square isnt already green, make it yellow
        bottomKeyboard(find(topKeyboard == guessNums(i))) = yellow;
    end
else
    %otherwise, make it gray
    bottomKeyboard(find(topKeyboard == guessNums(i))) = gray;
    bottomGameBoard(guesses+1, i) = gray;
```

```
    end
    currentPos = currentPos + 1; %update current position
  end
  %upadte bottom scene with new keyboard and game board
  bottomScene(9:11, 2:11) = bottomKeyboard;
  bottomScene(2:7, 5:9) = bottomGameBoard;

  newBottomScene = bottomScene;

end
```

Making the ending screen

```
function endScene = makeEndScene(win_scene, correctWord, win)
  Q = 11; W = 12; E = 13; R = 14; T = 15; Y = 16; U = 17; I = 18; O = 19; P = 20; %keyboard row 1
  A = 21; S =22; D = 23; F = 24; G = 25; H = 26; J = 27; K = 28; L = 29; %keyboard row 2
  Z = 30; X = 31; C = 32; V = 33; B = 34; N = 35; M = 36; %keyboard row 3

  % same with colors
  green = 4;

  topWinScene = ones(9, 14); %initialize top and bottom scenes and sizes
  botWinScene = ones(9, 14);

  %initialize  title letters
  if(win == 1)
    winTitle = [Y, O, U, 1, W, I, N];
  else
    winTitle = [Y, O, U, 1, L, O, S, E];
```

```
end

%translate the correct guess into numbers to use for sprites
correctNums = ones(1, 5);
for(i = 1:length(correctWord))
   correctNums(i) = lettertoNum(correctWord(i));
end

correctText = [C, O, R, R, E, C, T, 1, W, O, R, D]; %initialize correct word letters

playAgainQ = [P, L, A, Y, 1, A, G, A, I, N]; %initializeplay again question letters
playAgainTopText = [1, 1, Y, E, S, 1, N, O, 1, 1]; %initialize yes / no letters
playAgainBotText = [1, 1, 7, 8, 9, 1, 7, 9, 1, 1]; %initialize yes / no backgrounds

%adjust the top scene according to the new letters and placement
if(win == 1)
   topWinScene(2, 4:10) = winTitle;
else
   topWinScene(2, 4:11) = winTitle;
end
topWinScene(4, 2:(1+(size(correctText, 2)))) = correctText;
topWinScene(5, 5:9) = correctNums;
topWinScene(7, 3:12) = playAgainQ;
topWinScene(8, 3:12) = playAgainTopText;

%adjust the bot scene according to the new letters and placement
botWinScene(5, 5:9) = ones(1, 5) * green;
```

```
    botWinScene(8, 3:12) = playAgainBotText;


    drawScene(win_scene, botWinScene, topWinScene)
end
```