Pseudocode:

```
Module(s) to import:
    Random to pick random word, to pick random starting_player
make global variables:
     players_started_list to keep track of who’s already started
    wheel_list list of wheel options
Initialize_game() takes no arguments, returns list of 3 player dictionaries
    Create empty player list
    For 1-3:
        Create dictionary {player: #, temp bank: 0, perm bank: 0}
        Append dictionary to player list
    Returns player_list
get_word(wordbank_file): takes a file of words, returns list of words
    use with open…as to open a wordbank_file:
        read wordbank_file into a word_list
    return word_list
starting_player(player list): takes player list, returns random player dictionary
    already_picked = True Boolean for while loop
    while already_picked == True:
        starting_player = random.choice(player_list)
        if starting_player not in players_started_list:
            already_picked = False
        else:
            append starting_player to players_started_list
            break
    return starting_player
get_word(word_list) takes list of words, returns randomly chosen word not in chosen_words_list
    chosen_word_index = random.randrange(0, len(word_list))
    chosen_word = popped word_list at chosen_word_index picks word/removes it from list
    return chosen_word

hidden_word(correct_guessed_list, chosen_word):
    hidden_word = [letter if letter in correct_guessed else ‘-‘ for letter in chosen_word]
    return *hidden_word

round1or2(chosen_word, players_list, starting_player): takes word, player_list, starting_player, returns winner and has side effects on players_list
    initialize hidden word hyphen for letter in chosen_word
    initialize consonant_list list of lowercase consonants
    initialize vowel_list list of lowercase vowels
    initialize incorrect_guessed_list list of incorrect guesses
    initialize correct_guessed_list list of correct guesses
    swap player_list order either using a temporary variable or python swap
    word_guessed = False Boolean for while loop
    print(brief details about word. Player order)
    while word_guessed == False:
        for each player:
            turn_ended = False
            can_pick_vowel = False
            while turn_ended == False:
                print(message with brief instructions)
                prompt user to spin wheel
                wheel_option = random to get item from wheel_list
                if wheel_option is bankrupt:
                    print(message)
                    update player dictionary to reflect money lost
                    turn_ended = True
                elif wheel_option is lose turn:
```

```
    print(message)
    turn_ended = True
else:
    print(message)
    already_picked = True Boolean for while loop
    while already_picked == True:
        guess = prompt for player guess
        if guess in correct_guessed or in incorrect_guessed:
            print(message)
            continue
        elif guess in vowel:
            if can_pick_vowel == False:
                print(message)
                continue
            else:
                already_picked = False
                deduct vowel cost
                check guess
        elif guess in consonant:
            if incorrect:
                already_picked = False
                append incorrect_guessed
                turn_ended = True
            if correct:
                already_picked = False
                append correct_guessed
                update temp bank
                print(hidden_word)
                if ‘-‘ not in hiddenn_word:
```

```
                                    winner = player
                                    word_guessed = True
    update winner’s perm bank to reflect temp bank value
    zero all players’ temp banks
    return winner

richest(player_list): take player_list and return player with highest perm bank
    richest = player_list.copy()
    total = 0
    for each player:
        if richest[i][‘perm_bank’] > total:
            total = richest[i][‘perm_bank’]
        else:
            richest.pop(i)
    return richest should be list of one player, the richest

round3(richest, chosen_word):
    correct_guessed = [r, s, t, l, n, e]
    incorrect_guessed = []
    print(instructions)
    guess = prompt user for guess
    repeat code from round1or2 to flow and check guesses
    if richest guesses word:
        print(congratulatory message)
        update bank
    else:
        print(sorry message)
    return richest
```

```
print(congratulations, {richest}! You won the game with a total
```