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*

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\_words(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

append starting\_player to players\_started\_list

else:

continue

return starting\_player

choose\_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)+1)

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 wheel\_list *list of wheel options*

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

blindly pick 2 consonants and vowel

see results

guess word

if richest guesses word:

print(congratulatory message)

update bank

else:

print(sorry message)

return richest

print(congratulations, {richest}! You won the game!