define main():

keepGoing gets True

while keepGoing:

user choice gets menu choice

user choice gets converted to integer

If userChoice = 0

keepGoing gets False

tell the player thank you for playing

Elif userChoice = 1

getDefaultGame()

Elif userChoice = 2

Elif userChoice = 3

Elif userChoice = 4

Elif userChoice = 5

Elif userChoice = 6

Else:

Tell the user we need 0-5

Define getMenuChoice():

Print menu

userChoice = input()

Return userChoice

define playGame():

parameter of game

keepGoing gets True

currentNode gets “start”

while keepGoing:

if currentNode == “quit”:

keepGoing gets False

else:

currentNode gets PlayNode with game and currentNode as parameters

Define playNode():

If currentNode is within the keys of the game dictionary:

Assign variables to the items in each node of the dictionary and have game sub currentNode receive them

returns the next node

define getDefaultGame():

create dictionary to hold nodes

return dictionary

define editNode():

given the current game structure

list all the current node names

get a node name

if that node exists

copy that node to newNode

otherwise:

create newNode with empty data

use editField() to allow user to edit each node

assign the new inputs

define editField():

get a field name

print the field's current value

if the user presses 'enter' immediately

retain the current value

otherwise...

use the new value

define saveGame():

save the game to a data file

print the current game dictionary in human-readable format

Save the file in JSON format

Define loadGame():

Locate the file that was saved

open that file

load the data into the game object

return that game object