Degree is an inventory item.

Global access

Game class. Or gamestate class. This is accessible from within the menu classes.

Game.Character.Health

Automatic generation ….. focus on first, (arbitrary numbers so we can continue with testing)

Let's get a game from start to finish running. I.e. a quick barebone autotype.

Character class… In an external .py file. A dict. Or a class.

game\_state.character.health

or..

game\_state.character.strength / 2 ?

What about having a text box in the menu class? For lots of text like news events....

Er, sure. Well, no text boxes by default pygame. There are some wrapping text box code snippets on the internet. Jesse and Roland each find good ones.

J/k

There could be a game speed mode:

By days, by 1 day, by 7 days, by half-months, by months (full speed)

There could be a game difficulty mode:

Expert mode, Easy mode, etc. …

Completely unneccesary idea, but fun stuff…:

Bar for trumps anger, red.   
Background turns to a blood red the higher his anger gets.

# Some pseudo code…...

#class CharacterTypes:

#name: Bill

#health:

#Jane

#Wanda

#StevenBillyJimBo

class Character:

def \_\_init\_\_ (self, create\_type):

self.attributes = {

'name': 'Default',

'health': 3, #0-3

'strength': 3, #0-3

'gender': 'male',

'age',

}

if create\_type == 'random':

self.random()

pass

def random(self):

num = random(0,3)

if num == 0:

self.attributes.name = 'Bill'

self.attributes.health = 3

self.attributes.strength = 2

self.attributes.age = 86

elif num == 1:

pass

class GameState:

def \_\_init\_\_(self):

self.current\_screen = OpeningMenu()

self.character = Character('random')

self.game = Game()

#def randomCharacter():

# self.ch

# passB

class Game:

def \_\_init\_\_(self):

terms\_to\_play = '' # 1, 2, 999

#self.score = …

# \_\_main\_\_

game\_state = GameState()

#game\_state.current\_screen

#game\_state.game = Game()