PITCH

Working Title: Drop
Premise: You are broke. You own a laptop and have an internet connection.

GAME MECHANICS

ACTIONS

these are all the buttons that players can interact with, i.e. the click part of our clicker game

COLLECT DATA

- +1 to Data Files

CASH-IN DATA

- requires 10 Data Files
- -10 to Data Files
- add +n to Cash, based on this:

```
n = 0
      success_chance = round(random.uniform(0.45, 0.55)) +
database_modifier
      for _ in range(data_files_count):
            c = round(random.random())
            if c <= success_chance:
                  n += 2500
```

HIRE COLLECTOR

- +1 to Collectors
- -100 to Cash

HIRE LAUNDROMAT

- -50 to Cash

BUY BITCOIN

- -5000 to Cash
- +1 to Bitcoin

SELL BITCOIN

- -1 to Bitcoin
- +5000 to Cash

COLLECTABLES

these are resources that the player can interact with/manage.

DATA FILES

- main collectable of the game
- each is worth 2500 Cash
- each Data File has a 45% - 55% chance to fail at converting to Cash
- base cap for data files is 50, this can be expanded by purchasing Databases

CASH

- used to buy upgrades/hire services
- can further deplete with the purchase of various upgrades (each upgrade will describe its Cash requirement)

BITCOIN

- 1 Bitcoin = 5000 Cash
- unlocked when the player reaches 5000 in cash

UPGRADES

things the player can purchase with Cash to access various effects (automatically perform Actions, increase capacity of Collectibles, etc.). actions may be accessible through their related upgrade (e.g. COLLECTORS has a drop down menu when the action HIRE COLLECTOR is located)

COLLECTORS

- collects Data Files for the player
- +1 to Data Files per day
- -50 to Cash per 7 days

DATABASE

- +50 to Data File total capacity
- decreases the likelihood of a Data File failing to convert to Cash by 0.1 per Database

LAUNDROMAT

- reduces the chance of Arrest
- -10 to Cash per day

EVENTS

ARREST

- “You’ve been arrested for tax fraud.”
- signals a game-over for the player
- occurs only after the player has reached a cumulative 100 Data Files
- has a 1% chance of occuring on any given day, modified by the following
    - each Laundromat decreases the chance of Arrest happening by .5, diminishing returns as described:

```
laundromat_modifier = 0
for _ in range(laundromat_count):
     laundromat_modifier += 0.5 * (1/(1.25**(i - 1))
```

    - for every 500 Cash above 6000 Cash, the chance of Arrest increases by 4%

```
cash_modifier = 0
excess = cash - 6000
if excess > 0:
     x = excess // 500
     for i in range(x):
          cash_modifier += 0.04
```

    - for each month past 2 months, the chance for arrest increases by 10%

```
month_modifier = (month - 2) * .1
```

- the chance for Arrest is updated daily, is the first to be calculated each day

PROGRAM MODULES

gametime.py

gameTimeData(*day, week, month, year*)

gameTimeData is a class used to track and display in-game time. Time in this game updates at a speed of one day/second. There are 7 days in a week, 4 weeks in a month, 28 days in a month, and 13 months in a year. These values are arbitrary, but were chosen to simplify coding the time system. (Also canonically, this is how time works in the universe of the game.)

.addYear()

adds 1 to self.year.

.addMonth()

adds 1 to self.month, as well as resets the month count to 1 and calls addYear() when self.month is 13.

.addWeek()

adds 1 to self.week, as well as resets the week count to 1 when self.week is 4.

.addDay()

adds 1 to self.day, as well as:

- calls addWeek() when self.day % 7 == 1
- calls addMonth() when self.day is 28, resets self.day to 1

gamestate.py

GameState(saveFile = None)

is a class that performs the 'actions' of each buttons in the game (as stated above).

.calcScore()

computes the player's current score which is based on the cumulative cash count.

gui.py

The following are the functions used in this module:

labelUpdate()

updates the following:

- number of data files
- current cash
- collection rate
- number of bitcoins
- arrest risk (percentage)

update()

updates the time (in terms of: Day, Month, and Year).

Button(name, x, y)

is a class used to display the buttons such as ‘COLLECT DATA’, ‘CASH-IN DATA’, ‘HIRE COLLECTOR’, ‘HIRE LAUNDROMAT’, ‘BUY BITCOIN’, and ‘SELL BITCOIN’.

menugameover.py

The following are the functions used in this module:

compareHighScores(score)

compares current score with the five highest scores saved in the ‘highscore.txt’ file.

gameOverScreen(score)

displays ‘GAME OVER’ together with the total score once the player got arrested.

highScoreScreen()

displays the five highest scores.

mainMenuScreen()

displays the ‘NEW GAME’, ‘HIGH SCORE’, and ‘EXIT GAME’ buttons.

start.py

initiates the program by calling mainMenuScreen() function in menugameover.py

References:

https://darknetdiaries.com/episode/26
https://www.businessinsider.com/tax-refund-average-by-state-2018-1
http://mentalfloss.com/article/94623/numbers-how-americans-spend-more-their-money
https://klasing-associates.com/2016-statistics-tax-crime-convictions-prison-sentencing/
https://www.financialsamurai.com/average-net-worth-is-huge/
https://thecollegeinvestor.com/14611/average-net-worth-millennials/