Class Struggle Design Document

Apt word for this project: “class struggle.” Also known as “class warfare,” this conflict of interest is a defining feature of human society. It also makes for a quirky board game! This game was developed by NYU professor Bertell Ollman in 1978, and it disappeared from distribution around 1994. It’s now a collectors item for niche board game enthusiasts.

It has an extensive list of rules and features, and for scoping purposes for this project, I will simulate the simplest conditions to reach the game’s end state: nuclear war or revolution.

In this abbreviated version of this game, there will be two players who alternate taking turns rolling the dice, and will continue to gain assets or debits as they traverse the spaces until either player lands on square 84 (Revolution) or the Capitalist player lands on square 81 (Nuclear War). For this version, I eliminated the GREEN chance cards and instead am using RED/BLUE chance cards, so there is a card draw at every turn.

As this is developed further, it would be great to program more than two players into gameplay, the alliance strategy, better visualization, and advanced board actions from the original game rules.

Classes

Board

Methods

- create_board - will generate and print an ascii version of the board, spaces labeled 1 to 84
- map_labels - would love to do something cool and visual here, but will endeavor to create a list of tuples (square number, label) that will show if a player has landed on a RED, BLUE, or CONFRONTATION square

Game

Attributes

- __end_state = False
- __worker_chance_cards = {} Card text here
- __captialist_chance_cards = {} Card text here

Methods

- start_map
- create_players
- shuffle_card_decks
- play_game - while __end_state == False

Capitalist

Attributes

- assets = #
- debits = #
- board_position = #

Methods

- Setters
    - set_assets(self, change)
    - set_debits(self, change)
- Getters
    - get_assets(self)
    - get_debits(self)

Worker

Attributes

- assets = #
- debits = #
- board_position = # (1-84)

Methods

- Setters
    - set_assets(self, change)
    - set_debits(self, change)
- Getters
    - get_assets(self)
    - get_debits(self)

Turn

Methods

- roll_dice
- move_player - this will be the method that checks whether we've reached end state, setting __end_state = True (Player lands exactly on "84" or Capitalist landing on "81"), otherwise updates board_position
- get_square_label - determines whether the player does nothing (turn ends), pulls a card (turn ends after gaining assets/debits, or begin confrontation (whoever has the most points gets three dice throws in a row)
- draw_card - updates assets/debits based on card prompt
- confrontation - total both players' points and loop through three dice rolls for whoever wins