Object Oriented Programming

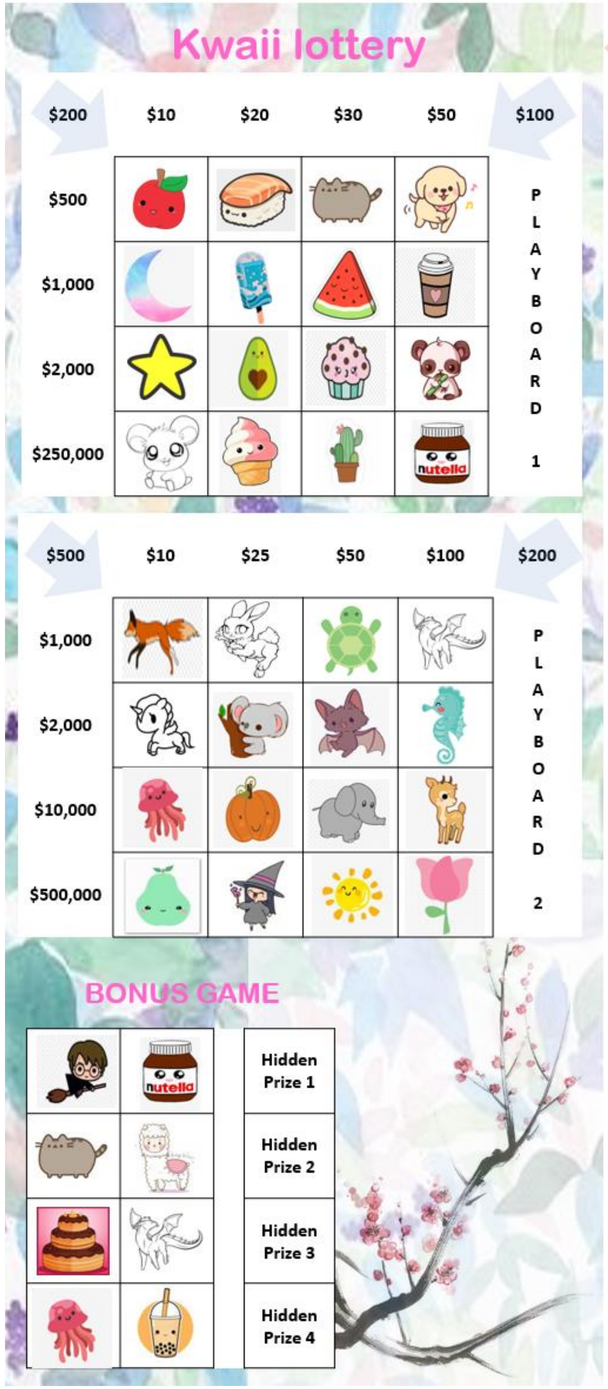
Kwaii Lottery

March 10, 2021

By: Mai La

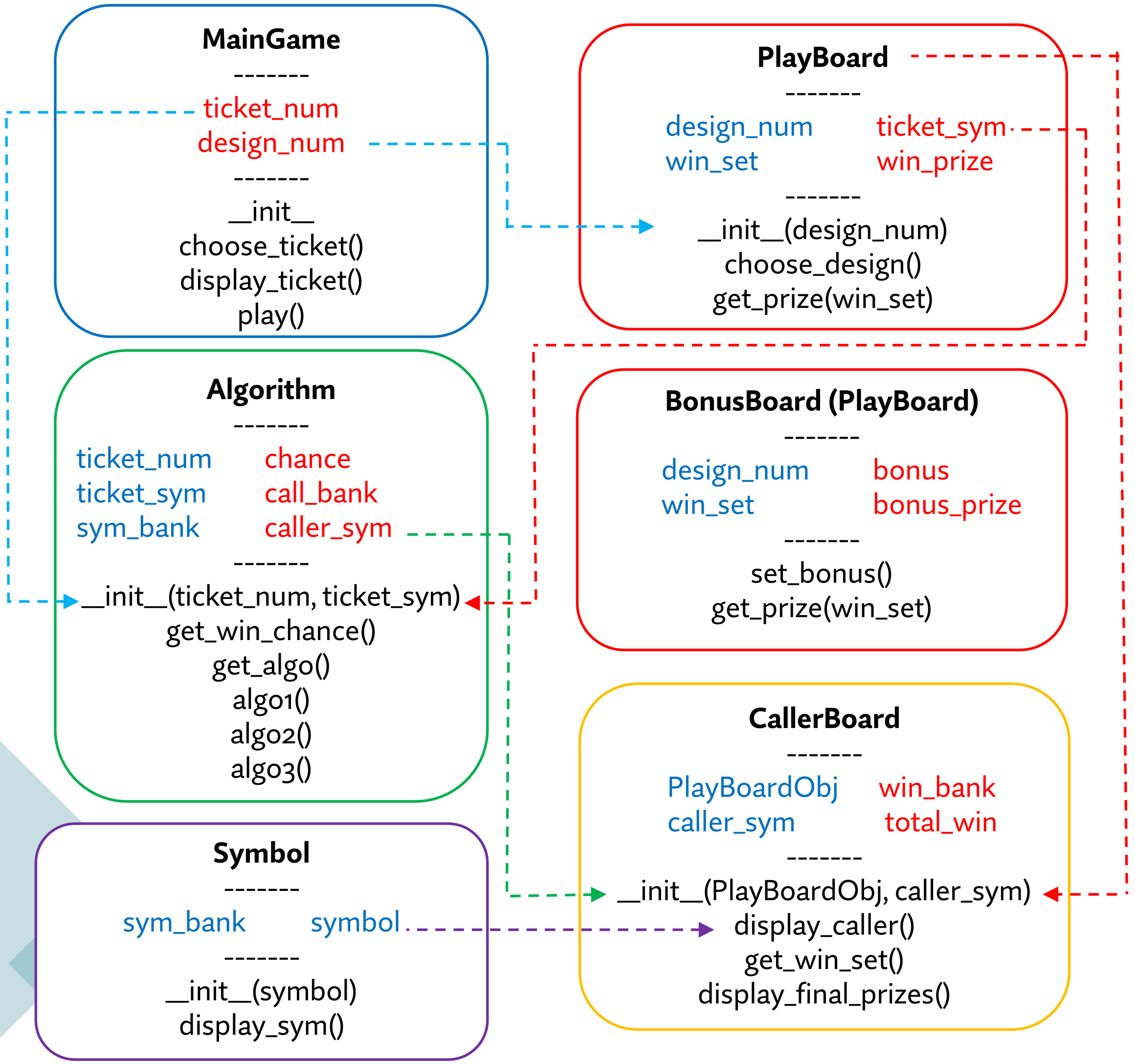
Game design

- HOW TO PLAY?
  - Buy a ticket for $5 and have a chance to win up to $500,000 by matching symbols
  - 3 ticket designs, 10 ticket numbers
  - 28 symbols to be revealed, scratch out any that matched on the ticket
  - Win the prize for any matching row, column, diagonal
  - Hidden prize is revealed at the end of the game
- CHANCE TO WIN
  - 50% no win
  - 40% win up to $2,000
  - 10% win up to $500,000



Objects

Note - color coding:
class: Bold
Input attributes: blue
Output attributes: red
Methods: black()



MainGame
----
ticket_num
design_num

__init__
choose_ticket()
display_ticket()
play()

PlayBoard
----
design_num
win_set
ticket_sym
win_prize

__init__(design_num)
choose_design()
get_prize(win_set)

Algorithm
----
ticket_num
ticket_sym
sym_bank
chance
call_bank
caller_sym

__init__(ticket_num, ticket_sym)
get_win_chance()
get_algo()
algo1()
algo2()
algo3()

BonusBoard (PlayBoard)
----
design_num
win_set
bonus
bonus_prize
----
set_bonus()
get_prize(win_set)

CallerBoard

PlayBoardObj
caller_sym
win_bank
total_win

__init__(PlayBoardObj, caller_sym)
display_caller()
get_win_set()
display_final_prizes()

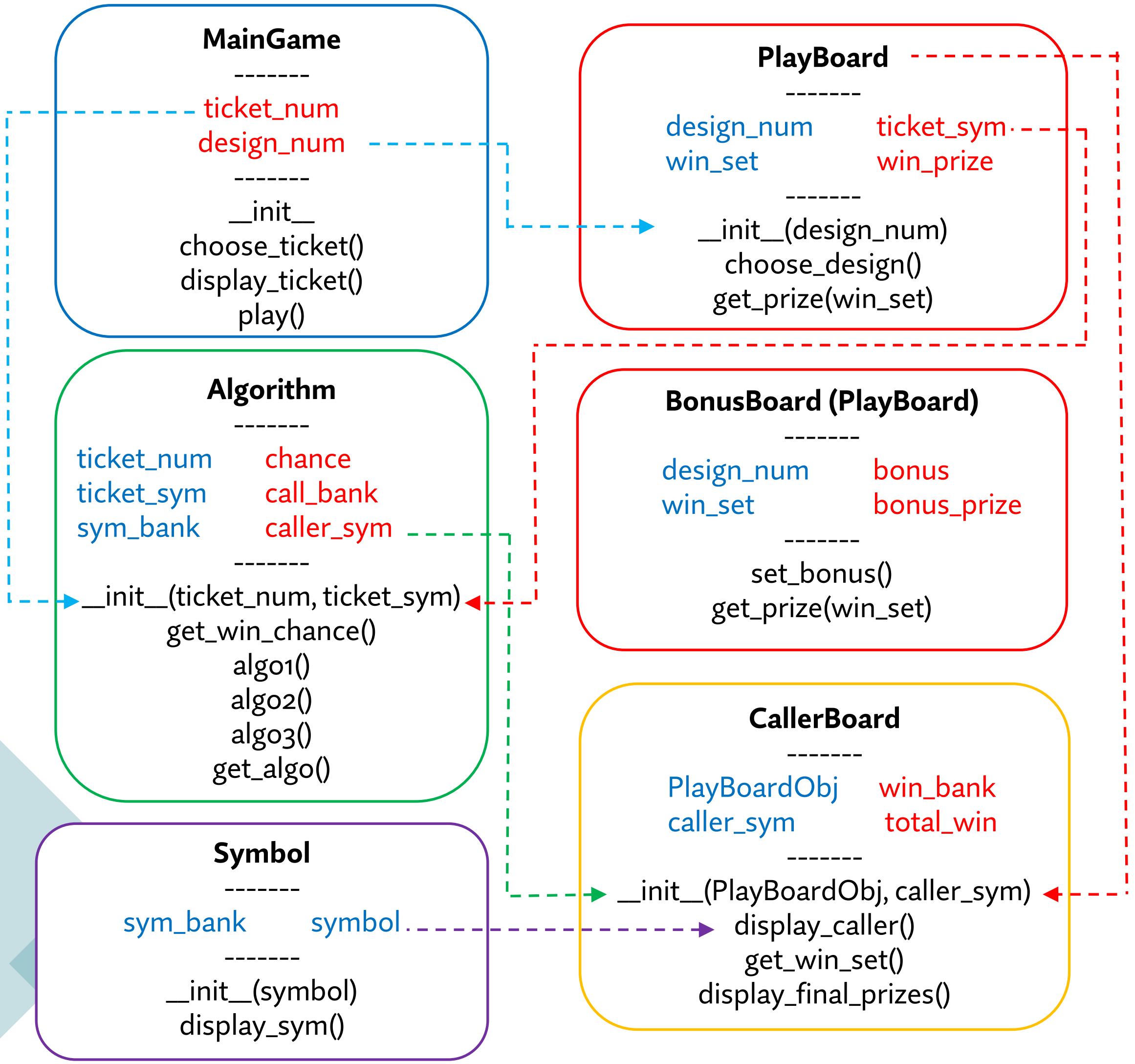
Symbol
----
sym_bank
symbol

__init__(symbol)
display_sym()

- **MainGame:**
  - Prompt player to choose a ticket design & ticket number
    >> choose_ticket()
    # Output: ticket_num, design_num
    >> display_ticket()
    # Output: show the picture of the ticket to player
    >> play()
    # Output: navigate through the game
- **PlayBoard:**
  - Return the ticket symbols
    >> choose_design()
    # Input: design_num # Output: ticket_sym
  - Get winning prize for a matched row, column, or diagonal
    >> get_prize(win_set)
    # Input: win_set, prizes # Output: win_prize
- **BonusBoard**
  - Child class of playboard, set random prizes for bonus game
    >> set_bonus()
    # Input: prizes # Output: bonus
  - Get winning prize for a matched bonus row
    >> get_prize(win_set)
    # Input: win_set, bonus # Output: bonus_prize

Objects

Note - color coding:
class: Bold
Input attributes: blue
Output attributes: red
Methods: black()



MainGame
----
ticket_num
design_num

__init__
choose_ticket()
display_ticket()
play()

PlayBoard
----
design_num
win_set
ticket_sym
win_prize

__init__(design_num)
choose_design()
get_prize(win_set)

Algorithm
----
ticket_num
ticket_sym
sym_bank
chance
call_bank
caller_sym

__init__(ticket_num, ticket_sym)
get_win_chance()
algo1()
algo2()
algo3()
get_algo()

BonusBoard (PlayBoard)
----
design_num
win_set
bonus
bonus_prize
----
set_bonus()
get_prize(win_set)

CallerBoard

PlayBoardObj
caller_sym
win_bank
total_win

__init__(PlayBoardObj, caller_sym)
display_caller()
get_win_set()
display_final_prizes()

Symbol
----
sym_bank
symbol

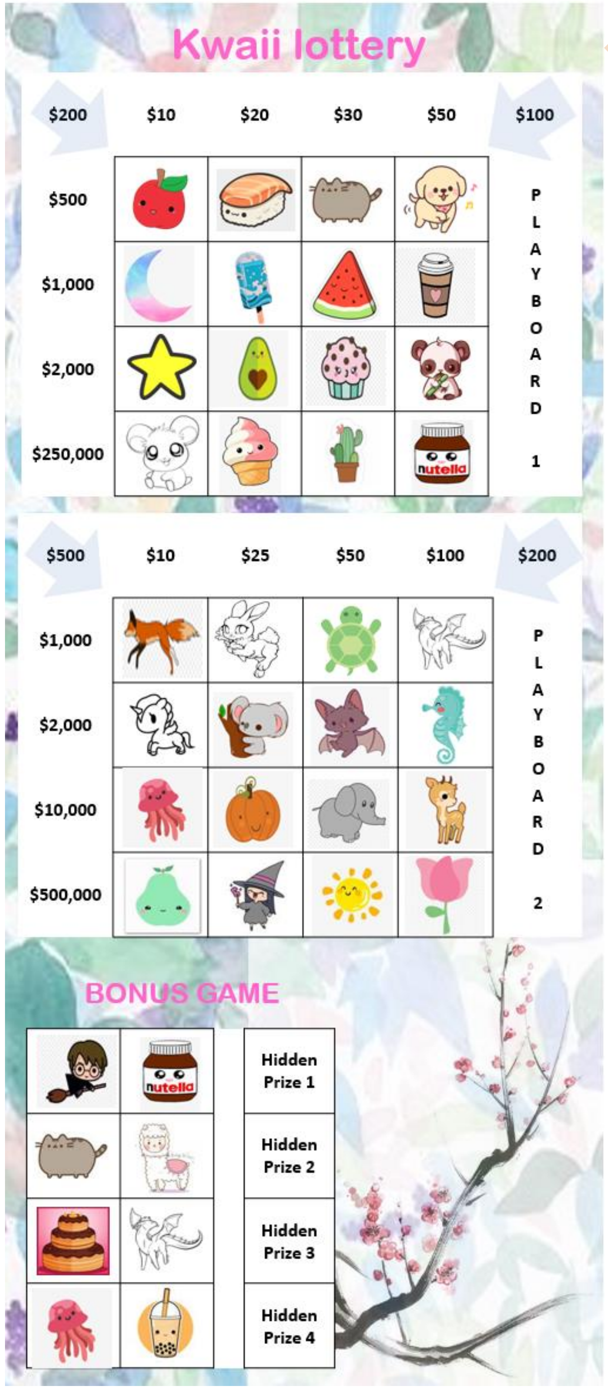
__init__(symbol)
display_sym()

- **Algorithm:**
  - Get the winning chance from the ticket number
    >> get_win_chance()
    # Input: ticket_num   # Output : chance
  - Choose an algorithm to run base on the winning chance. Return the 28 caller symbols
    >> algo1()   # No win
    >> algo2()   # Win up to $2,000
    >> algo3()   # Win up to $500,000

    # Input: ticket_sym, sym_bank_bank   # Output : call_bank
    >> get_algo()
    # Input: chance, call_bank   # Output : caller_sym
- **CallerBoard:**
  - Display caller symbols & picture of symbols to player
    >> display_caller()
    # Input: caller_sym, PlayBoardObj   # Output: symbol picture
  - Get winning set from the caller symbols, display bonus hidden prizes & total winning prizes
    >> get_win_set()
    # Input: caller_sym   # Output : win_bank
    >> display_final_prizes()
    # Input: win_bank   # Output: display bonus_prize, total_win
- **Symbol**
  - Store the bank of all symbols & path to their pictures
  - Show the picture of a symbol to the player
    >> display_sym()
    # Input: symbol   # Output: symbol picture

Running & Testing

- HOW TO PLAY?
  - Run in the command line: python kwaii_lottery.py
  - Read the Help Menu and How To Play instruction
  - You can enter any options that the game is asking. If an invalid answer is entered, it will print out a helpful message and repeat the question, until you enter a valid answer.
  - A picture of your game ticket will be shown in a separate window, use Photos drawing tool to keep track of your game progress.
  - A screen shot example of how I play and enter different answers is shown in the game_demo.ipynb file, it could also be used for testing the output of each class separately.



Enjoy the game!