

Assignment 1: Joker Twins

Introduction to Computer Science (CS-UH 1001) - Fall 2024

1 Code of Conduct

In the following, we outline the rules and expectations that govern the submission of academic assignments. All students are expected to adhere to the rules of academic honesty and ethical conduct in all academic work and take responsibility for their academic work.

1. **Originality:** All assignments submitted must be the original work of the student. Any document and program code that is submitted must be **entirely** written by the student.
2. **Discussions:** While students are encouraged to discuss their questions with other students, these discussions should be limited to general solution techniques and concepts. Discussions about concrete code or specific results that the student intends to submit are not allowed and are considered an integrity violation.
3. **Solution Sharing:** Discussing an assignment with others should never lead to the possession or temporary access to complete or partial solutions from others. Possessing and accessing solutions, regardless of whether in paper or digital form, and irrespective of the source of the solution, is not allowed and is considered an integrity violation.
4. **Unauthorized Assistance:** Obtaining solutions from sources outside of the course, including but not limited to students of current or previous years or sections, other universities, the Internet, AI-Chatbots, etc. is not allowed and is considered an integrity violation.
5. **Code Sharing:** Sharing submissions with fellow students is not allowed and will be considered an integrity violation. Students must maintain confidentiality and keep their code private.
6. **Publication:** Publishing submissions or assignment descriptions online in any form is not allowed.
7. **Academic Responsibility:** Each student is solely responsible for the work submitted for an assignment. Any doubts regarding the authenticity of the work may lead to an investigation and necessary action by the academic administration. If any suspected violations are detected, they will be reported to the academic administration as per the policies of NYU Abu Dhabi. Disciplinary actions may include but are not limited to failing the assignment, failing the course, or suspension from the university. More details can be found at:

<https://students.nyuad.nyu.edu/academics/registration/academic-policies/academic-integrity/>

Penalty for integrity violations:

Every student found to be involved in an integrity violation will be penalized. In case a first integrity violation is reported, the penalty is a 10% deduction from the final grade, or the full

percentage of the graded assessment, whichever is higher. A second integrity violation will result in an automatic course failure.

2 Introduction

Joker Twins is a modified version of the classical Memory game, a game that has long been a favorite game for all generations. In fact it is so simple that very young children can play it with ease. It requires observation, concentration and a good memory to win. The game is also known as Concentration, Pelmanism, Shinkei-suijaku, Pexeso and Pairs. An example of the classical memory game can be played here:

<https://www.helpfulgames.com/subjects/brain-training/memory.html>

The goal of this assignment is to implement the Joker Twins game in Python3, using the material that has been covered in class. This will give you practice in working with lists, decision structures, iterations and loops. Additionally, it is a good exercise in decomposing a larger problem into smaller and more manageable parts.



The Joker Twins game in this assignment is a two-player game and is played on a two-dimensional board where the memory cards are represented as letters. That is, the "twins" are the upper and lower case of letters, for example "A" and "a". Each card and its twin are randomly placed on the board and hidden from the players. The two players take turns to turn over two cards in each turn. If the two cards have the same letter (upper and lower case), then the player "keeps" the cards, the cards will be removed from the game, and the player continues playing until two cards are turned over that are not twins. In that case, the cards will turn face down again and the game continues with the other player's turn. The winner is the player with the most cards (i.e. twins) when no cards are left in the game.

What sets the Joker Twins game apart from the classical memory game is that the game contains two Joker cards that are added to the other cards. A Joker card, once turned over by a player, forms a twin with the next or previous turned over card. As a result, both cards

comprising the twin plus the Joker will be turned over and 2 points are added to the player's score (1 for the twin cards + 1 for the Joker). Once a Joker card is utilized, it will also be removed from the game. If all twins have been turned over and only one or two Jokers remain on the board, the remaining Jokers will automatically be turned over. Each Joker will count as 1 point for the last player who revealed the final twin.

Figure 1 shows example output of the Joker Twins game with two players.

```

Player 2 score: 0
Player 1 score: 0
  A  B  C  D  E  F
+---+---+---+---+---+
0| # | # | # | # | # | # |
+---+---+---+---+---+
1| # | # | # | # | # | # |
+---+---+---+---+---+
2| # | # | # | # | # | # |
+---+---+---+---+---+
3| # | # | # | # | # | # |
+---+---+---+---+---+
4| # | # | # | # | # | # |
+---+---+---+---+---+
5| # | # | # | # | # | # |
+---+---+---+---+---+
Player 2: Enter a coordinate (e.g. B3):

```

a)

```

Player 2 score: 1
Player 1 score: 3
  A  B  C  D  E  F
+---+---+---+---+---+
0|  | # | # | # | # | # |
+---+---+---+---+---+
1| # | # | # | # | # | # |
+---+---+---+---+---+
2| # |  | # | # |  | # |
+---+---+---+---+---+
3| # | # |  | # | # |  |
+---+---+---+---+---+
4| # |  | # | # | # | # |
+---+---+---+---+---+
5| # | # | # | # |  | # |
+---+---+---+---+---+
Player 1: Enter a coordinate (e.g. B3):

```

b)

```

Player 2 score: 2
Player 1 score: 5
  A  B  C  D  E  F
+---+---+---+---+---+
0|  | # | d | # | # |  |
+---+---+---+---+---+
1| # | # |  |  | # | # |
+---+---+---+---+---+
2| # |  | # |  |  | # |
+---+---+---+---+---+
3| # | # |  | # | # |  |
+---+---+---+---+---+
4| # |  | D | # | # | # |
+---+---+---+---+---+
5|  | # | # | # |  |  |
+---+---+---+---+---+

```

c)

```

Player 2 score: 2
Player 1 score: 6
  A  B  C  D  E  F
+---+---+---+---+---+
0|  | # |  | # | # |  |
+---+---+---+---+---+
1| # | # |  |  | # | # |
+---+---+---+---+---+
2| # |  | # |  |  | # |
+---+---+---+---+---+
3| # | # |  | # | # |  |
+---+---+---+---+---+
4| # |  |  | # | # | # |
+---+---+---+---+---+
5|  | # | # | # |  |  |
+---+---+---+---+---+
Player 1: Enter a coordinate (e.g. B3):

```

d)

Figure 1: Example output of the game. a) Initial game (player 2 starts). b) Player 2 turned over 1 twin, player 1 turned over 2 twins, including 1 Joker. c) Player 1 just entered a second coordinate for their current turn (C0 or C4) and the game pauses to show the two turned over cards to both players. Note that no output below the board is shown while the game pauses. d) The score of player 1 is incremented and the program is waiting for the next input of player 1.

3 Implementation

The implementation of this game consists mainly of two stages:

Initialization:

The game starts with a board where columns are identified by letters and rows by numbers (see Figure 1). The board is represented as a two-dimensional list, with each element representing a card. Initially, all cards are shown face down and are hidden from the player (denoted by a "#"). The cards are randomly placed on a second, hidden board that holds the actual cards (letters) and the Jokers. This hidden board is used internally to save the locations of the cards, check for matches, etc. The dimensions of both boards must match with each dimension set to 6x6 and declared as a global constant variable. (You need one variable for the number of rows and one for the number of columns, as shown below).

The letters that represent a card should be dynamically generated by the program according to the board dimensions. That is, the program should generate letters from A-Q and a-q, plus two Joker. The program must always generate twins of a letter and they always start with an "A" and "a" and increment according to the English alphabet. Again, do not "hardcode" a list of all possible letter twins. These dynamically generated letters, including two Jokers which are represented by a "*", will be randomly placed on the board.

Since the game is a two-player game, the program chooses a random player for the first turn. The following code can be used to define the board dimensions and to determine a random player to start. Please note that the `randint()` function in line 9 should also be used to randomly place letters on the board:

```
1 import random          # imports the random module
2
3 NUM_ROWS = 6           # constant variable for number of rows
4 NUM_COLS = 6           # constant variable for number of columns
5 NUM_PLAYERS = 2        # constant variable for number of players
6
7 # the following command returns a random number between 0 and 1
8 # it can be used as an index for the player's score
9 turn = random.randint(0, NUM_PLAYERS-1)
10
11 # your code comes here
12
13 # alternating turns, i.e. turn alternates between 0 and 1
14 turn = (turn + 1) % NUM_PLAYERS
15
16 # your code comes here
```

After creating the board and placing the twins and Jokers, the board should be printed on the screen (as depicted in Figure 1a) where all cards are shown face down and are represented as a "#". In addition, both players' scores should be printed above the board in the order of their turns, and an input prompt for the randomly chosen player should be printed below the board.

Game phase:

The game starts after the first player inputs the first location/coordinate of the card that should be turned over. The coordinate the player enters consists of the column (a letter) followed by the row (a digit). For example, valid coordinates are B4, C1, A5, D0, etc. Notice that the column letter is case-sensitive. If the player enters an invalid coordinate or an invalid input, the program informs the player about this and asks the player for another input until the input is valid. Examples of an invalid input are AB, A 6, 1A, 32, A;5, a6, A-1, 5C, 2a, empty string, whitespaces, special characters, etc., coordinates out of the board dimensions, coordinates that are currently shown as a turned over cards or coordinates that do not contain a card anymore (i.e. turned over in a previous turn).

After a valid user input, the program (temporarily) displays the card (i.e. displays the actual card instead of the "#") and asks the player for the coordinate to turn over the second card. If both turned over cards form a twin (i.e. an uppercase letter and its lowercase letter twin), both cards are removed from the board (represented as " ") and the player continues its turn to find more twins. Remember, the Joker can be used to form twins and will be removed from the board as well, together with the other card of the twin that was not turned over yet. For example, if the player turns over a Joker and then an "e" (or vice versa), the card with an "E" will automatically be turned over and also removed from the board. Please note that Joker twins are also considered twins that add 2 points to the player's score when found. In case a twin is formed using a Joker and a letter, 2 points will be added to the player's score (1 point for the twin plus 1 point for the Joker). Also, the player's score should be updated and displayed after the game paused for two seconds (use code below).

A player's turn ends once two cards are turned over that do not form a twin. In that case, the program will pause for two seconds so that both players have the chance to memorize the turned over cards. Then, the board should be updated where both cards are turned face down again (displayed as a "#") and the program should clear the screen (use code below) before the board is displayed again. The game continues with the other player following the same steps as described above.

The following functions can be used to pause the program for two seconds and to clear the screen:

```
1 import os                # imports the os module
2 import time              # imports the time module
3
4 # your code comes here
5
```

```

6 time.sleep(2)          # pause the program for 2 seconds
7 os.system("clear")     # clears the screen
8
9 # your code comes here

```

After each turn, the program should check if further turns are possible and if not, the game ends and the program terminates. The program should inform the user that the game is over, which player won and what their scores are.

4 Grading

Description	Score (/15)
Using global variables (constants) for board dimensions and number of players	0.5
Correctly creating the board	0.5
Printing of the board, board labels, user scores and user input, as shown in Figure 1	2
Dynamically generating all possible twins	2
Randomly placing twins and Jokers on the board	2
Checking for invalid user input	2
Checking for invalid coordinates (repetitive input of same cell, out of board, etc))	2
Correctly turning over cards (reveal and hide)	2
Detecting the end of the game and terminating the game after a win	1
Using comments and readability of code (style, variable naming, etc.)	1

5 Submission

Submission details/policy are mentioned below:

Submission Deadline: The deadline of this assignment is after 10 days of its release via Brightspace. No extensions will be given.

Submission Format and System: You can directly submit your Python .py file on Brightspace. Submissions via email are not accepted. Note that your solution must work using Python 3 and Linux/Unix terminal, otherwise your submission will not be graded.

Late Submissions: Late submissions will be penalized by 20% per 24 hours.