CMSC12 Project

Your project is worth 15% of your final grade. However, there are additional specifications that can give you up to 17% for your project. In the case of excess, the 1-2% can be carried over/added to the lecture component.

In this project, you will be using **Python**. You get to choose one of the two problem domains below. Both options have a feature to **save and display top players data using files**. These are one-player games with an imaginary opponent/player that reacts to the player's actions. You will need to write a code for generating a **random number** to set opponent move/s randomly. One way to generate a random number in python is as follows: random.randint(a,b). The code returns a number N in the inclusive range [a,b], meaning a <= N <= b, where the endpoints are included in the range. The N is a number that can represent opponent action (e.g. 0 = "rock"; or 0 = "punch", depending on the game you chose).

Choose a project topic from the following (two) options:

Option 1. Simple fighting game

Mechanics of the game: This is a simple fighting game where a player makes an offensive move to inflict damage on an opponent/enemy. This is a one-player game where the opponent is played by the computer which, in turn, fights back by randomly determining its next move. Both player's health level starts at 125 and are reduced depending on the fight action's damage level.

The game starts by asking for the user's name and desired character: Boxer, Muay Thai, and Street fighter. The game (proper) then proceeds by asking the user to choose from the following moves: Punch, Kick, and Grapple. The levels of damage are as follows: Punch (min 25 to max 50, Kick (min 20 to max 40), and Grapple (min 15 to max 30). In return, the opponent fights back using a randomly determined fight action. Each of the player's health is reduced depending on the fight action. The game ends when either of the player's health reaches less than or equal to zero (when health falls below 0, display 0). When a player's health reaches zero, the other player is declared the winner and the game ends. See sample run below.

To get full 15%: The user is able to play the game from start to finish. All game

rules are followed and program requirements are met.

To get full 16%: Implement a different theme (define your own character types,

moves) and submission is noticeably creative with how the game

is presented; Ascii text present (+0.25)

To get full 17%: With feature to show top players. Top player is determined by

recording the remaining health level after winning the game. The

higher the health level the better.

Sample program run:

Simple fighting game Mechanics of the game:

SIMPLE FIG Select a [1] B [2] M [3] S Other [8] H [9] R [0] E ce: 1 /	uay Thai treet fighter options igh scores estart
Select a [1] B [2] M [3] S Other [8] H [9] R [0] E ce: 1 /	fighter: oxer uay Thai treet fighter options igh scores estart
[1] B [2] M [3] S Other [8] H [9] R [0] E ce: 1	oxer uay Thai treet fighter options igh scores estart
[2] M [3] S Other [8] H [9] R [0] E ce: 1 /	uay Thai treet fighter options igh scores estart
[3] S Other [8] H [9] R [0] E ce: 1 /	treet fighter options igh scores estart
Other [8] H [9] R [0] E ce: 1 /	options igh scores estart
[8] H [9] R [0] E ce: 1	igh scores estart
[9] R [0] E ce: 1 /	estart
[0] E ce: 1 //	
ce: 1	
PLAYER (HEALTH: ENE HEALTH: [1] F [9] R [0] E	
PLAYER (HEALTH: ENE HEALTH: [1] F [9] R [0] E	
PLAYER (HEALTH: ENE HEALTH: [1] F [9] R [0] E	/
PLAYER (HEALTH: ENE HEALTH: [1] F [9] R [0] E	
PLAYER (HEALTH: ENE HEALTH: [1] F [9] R [0] E	/
PLAYER (HEALTH: ENE HEALTH: [1] F [9] R [0] E	
PLAYER (HEALTH: ENE HEALTH: [1] F [9] R [0] E	
HEALTH: ENE HEALTH: [1] F [9] R [0] E	T NIGHT
HEALTH: ENE HEALTH: [1] F [9] R [0] E	
HEALTH: ENE HEALTH: [1] F [9] R [0] E	Bover)
ENE HEALTH: [1] F [9] R [0] E	
HEALTH: [1] F [9] R [0] E	
[1] F [9] R [0] E	125/125
[1] F [9] R [0] E	
[9] R [0] E	MY
[0] E	MY
	MY 125/125
ce: 1	MY 125/125
· -	MY 125/125 ight estart
	MY 125/125 ight estart
FIGH	MY 125/125 ight estart
	MY 125/125 ight estart
 PLAYER (MY 125/125 ight estart xit

HEALTH: 108/125 Action: Punch Damage dealt to enemy: 20
HEALTH: 120/125 Enemy fought back! Enemy used: Grapple (Damage dealt:17)
Other options
Choice: 1
,,
PLAYER (Boxer)
HEALTH: 88/125 Action: Kick Damage dealt to enemy: 22
ENEMY
HEALTH: 98/125 Enemy fought back! Enemy used: Punch (Damage dealt:20)
Other options
Choice: 1
,,
PLAYER (Boxer)
HEALTH: 66/125 Action: Grapple Damage dealt to enemy: 17
ENEMY

	Enemy	used:	Kick	(Damage dealt:20)	
1					
			Othe	er options	
			[9]	Restart	
			[0]	Exit	
'					ı

<Fast forward>

Case 1: You win

,,
HEALTH: 20/125 Damage dealt to enemy: 17
ENEMY
You WON!
Other options

Congratulations Mylah! You are one of the top players. Score saved.



Choice: 9

Case 2: Opponent wins

,	,FIGHT	NIGHT
1		
ı	PLAYER (Bo	er)

HEALTH: 0/125 Action: Grapple	
Damage dealt to enemy: 21	
ENEMY	
HEALTH: 43/125 Enemy fought back! Enemy used: Kick (Damage dealt:18)
ENEMY HAS WON	
Other options [9] Restart [0] Exit	

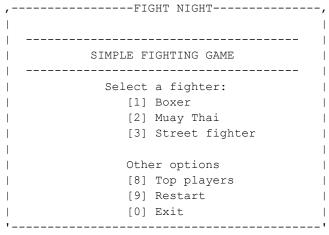
Back to main menu

,,
SIMPLE FIGHTING GAME
Select a fighter:
[1] Boxer
[2] Muay Thai
[3] Street fighter
Other options
[8] Top players
[9] Restart
[0] Exit
''

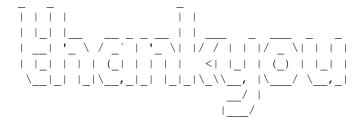
Choice: 8

-----,

Choice: 9



Thank you for playing, Mylah!



Option 2. Rock, Paper, and Scissors

Mechanics of the game: This is a one-player game where the player chooses from the following moves: Rock, Paper, and Scissors. The computer, in turn, responds by using a randomly determined move. Both player's scores start with 0, and 1 point is added for every win. The first to get 5 points wins.

The game starts by asking for the user's name. See sample run below.

To get full 15%: The user is able to play the game from start to finish. All game

rules are followed and program requirements are met.

To get full 16%: Implement a different theme (define your own character types,

moves) and submission is noticeably creative with how the game

is presented; Ascii text present (+0.25)

To get full 17%: With feature to show top players. Top players are determined by

recording the number of rounds/games it took to win the game.

The lower the number, the better

Sample program run:

Simple fighting game Mechanics of the game:

Your name: Mylah
,ROCK PAPER SCISSORS
PLAYER 1
SCORE: 0 [1] Rock [2] Paper [3] Scissors
PLAYER 2
SCORE: 0
Other options [9] Restart [0] Exit
Choice: 1
,ROCK PAPER SCISSORS
PLAYER 1
SCORE: 1 [1] Rock [2] Paper [3] Scissors Last move: Rock
PLAYER 2
SCORE: 0 Last move: Scissors
WIN

Other options [9] Restart

	'
ice: 2	
ROCK PAPER SCISSORS	,
	l
PLAYER 1	
SCORE: 2	 -
[1] Rock	
[2] Paper [3] Scissors	
Last move: Paper	'
PLAYER 2	
SCORE: 0	
Last move: Rock	
WIN	
	I
Other options	
[8] Restart	
<pre>[8] Restart [9] Restart</pre>	
[8] Restart	
<pre>[8] Restart [9] Restart</pre>	
[8] Restart [9] Restart [0] Exitice: 3	
[8] Restart [9] Restart [0] Exit	
[8] Restart [9] Restart [0] Exit ice: 3	
[8] Restart [9] Restart [0] Exit	
[8] Restart [9] Restart [0] Exit ice: 3	
[8] Restart [9] Restart [0] Exit ice: 3 ROCK PAPER SCISSORS PLAYER 1 SCORE: 2 [1] Rock	
[8] Restart [9] Restart [0] Exit ice: 3 ROCK PAPER SCISSORS PLAYER 1 SCORE: 2 [1] Rock [2] Paper	
[8] Restart [9] Restart [0] Exit ice: 3 ROCK PAPER SCISSORS PLAYER 1 SCORE: 2 [1] Rock [2] Paper [3] Scissors	
[8] Restart [9] Restart [0] Exit ice: 3 ROCK PAPER SCISSORS PLAYER 1 SCORE: 2 [1] Rock [2] Paper	
[8] Restart [9] Restart [0] Exit ice: 3 ROCK PAPER SCISSORS PLAYER 1 SCORE: 2 [1] Rock [2] Paper [3] Scissors	
[8] Restart [9] Restart [0] Exit ice: 3 ROCK PAPER SCISSORS PLAYER 1 SCORE: 2 [1] Rock [2] Paper [3] Scissors Last move: Scissors PLAYER 2	
[8] Restart [9] Restart [0] Exit ice: 3 ROCK PAPER SCISSORS PLAYER 1 SCORE: 2 [1] Rock [2] Paper [3] Scissors Last move: Scissors	
[8] Restart [9] Restart [0] Exit ice: 3 ROCK PAPER SCISSORS PLAYER 1 SCORE: 2 [1] Rock [2] Paper [3] Scissors Last move: Scissors PLAYER 2 SCORE: 1	
[8] Restart [9] Restart [0] Exit ice: 3 ROCK PAPER SCISSORS PLAYER 1 SCORE: 2 [1] Rock [2] Paper [3] Scissors Last move: Scissors PLAYER 2 SCORE: 1 Last move: Rock	
[8] Restart [9] Restart [0] Exit ice: 3 ROCK PAPER SCISSORS PLAYER 1 SCORE: 2 [1] Rock [2] Paper [3] Scissors Last move: Scissors PLAYER 2 SCORE: 1	
[8] Restart [9] Restart [0] Exit ice: 3 ROCK PAPER SCISSORS PLAYER 1 SCORE: 2 [1] Rock [2] Paper [3] Scissors Last move: Scissors PLAYER 2 SCORE: 1 Last move: Rock	
[8] Restart [9] Restart [0] Exit ice: 3 ROCK PAPER SCISSORS PLAYER 1 SCORE: 2 [1] Rock [2] Paper [3] Scissors Last move: Scissors PLAYER 2 SCORE: 1 Last move: Rock	

1	[9]	Restart
	[0]	Exit
' _		

((Fast forward)) Case 1. You win

,,
I DI AVED 1
PLAYER 1
SCORE: 5
[1] Rock
[2] Paper
[3] Scissors
Last move: Paper
·
DI AVED O
PLAYER 2
SCORE: 1
Last move: Rock
Last move. Nock
WIN
Other options
[8] Restart
[9] Restart
[0] Exit
· · · · · · · · · · · · · · · · · ·

5 points reached, we have a winner

Game 1 WIN

Game 2 WIN

Game 3 WIN

Game 4 LOSE

Game 5 No score (same move)

Game 6 WIN

Game 7 WIN

Congratulations Mylah! You are one of the top players. Player name saved. $\,$



Case 2. You lose

,ROCK PAPER SCISSORS,
PLAYER 1
SCORE: 3 [1] Rock [2] Paper [3] Scissors Last move: Scissors
PLAYER 2
SCORE: 5 Last move: Rock
LOSE
Other options Other options [8] Top players [9] Restart [0] Exit

5 points reached, we have a winner

```
Game 1 WIN
Game 2 WIN
Game 3 WIN
Game 4 LOSE
Game 5 No score (same move)
Game 6 LOSE
Game 7 LOSE
Game 8 LOSE
Game 9 LOSE
```

Thank you for playing, Mylah!

__/

Choice: 8

Choice: 0

Thank you for playing, Mylah!

