**DEAL OR NO\_DEAL**

* To Play Deal or No\_Deal Game with my Banker and win Exciting Amounts

**Team Name: Fearless Group**

**Developed By**

**Deeban N**

**1. INTRODUCTION**

The main motto of my game is to check their Luck in this game and cover the people to play the game without knowing the time of playing.

It is purely luck based game. Because, I have used randomized algorithm for picking and storing the money in the each briefcase. So, selection of briefcase will be more interesting and it will motivate the player to play the game repeatedly and continuously.

In my game I used four functions to play the game effectively. There is a function called instruction, where the player can see the instruction to do in this game. There is another function called pick my case, used to pick up our briefcase containing some money may be a smaller amount or may be a bigger amount. There is a function called case remover, where we can remove the other cases. Then at last there is a banker’s function, one of the important functions in my game, where calculation of banker’s amount will be processed. This banker will be called at the end of each and every level.

On Television: Deal or No Deal has become a popular game show on network television. Stir crazy contestants go on the show for a chance to beat the bank and walk away with a lot of money. But how many contestants are really making the best deals possible? As soon as they get onto the show, contestants let emotion get in the way of the real task at hand: to make the most money possible. Instead of making rational decisions with a cool head, contestants get too wrapped up in the theatricalities of the game itself when they should be really thinking about how they are going to effectively walk away a little heavier in the pockets.



**2. PROJECT DESCRIPTION**

It is purely luck based game. If the player knows the tricks of banker, then they can win the large amount. There will be 26 cases named from 1 to 26. At first player has to select his briefcase that contains some hidden money. The amount will be from Rs. 1 to Rs. 1000000. Then they should remove 4 cases, where we can show the amount in that removed case, at each level. After each level banker will offer some money, either the player can deal with the banker or he/she can continue the game till last. If the player rejected the entire banker’s offer, then he/she can take the money in his briefcase which may be smaller amount or may be bigger amount.

**2.1. Class Diagram**

|  |
| --- |
| **Player** |
| Case\_amount  Case\_checking  Owner\_case\_amount  My\_case |
| Instruction()  Pick\_my\_case()  Case\_removing()  Random() |

|  |
| --- |
| **Bankers** |
| Case\_amount  Decision  Final\_choice  Case\_checking |
| Banker()  Case\_removing()  Exit() |

**2.2. Implementation Details**

It is a python project. In eclipse using pydev python interpreter, my source code is running. I have attached my source code and their sample output as a screenshot at the end of the document.

I have implemented four functions for playing the game. In that, three functions are used for playing the game and one function is used for giving instruction.

Step 1: If the player needs some instruction to play the game then he/she should press the key 1, if they don’t need any instruction then they should press the key 0

Step 2: At first, player has to select his briefcase that contains some money, which will be hidden till the game ends.

Step 3: Then they should remove 4 cases at each level, where we can show the amount in that removed case.

Step 4: After each level Banker will offer some amount, if they deal with the banker’s offer, they will lose the amount in their case. If they continued the game by giving No deal with the banker, still last, they can win the amount in their case even it may be a smaller amount or may be a bigger amount.

|  |
| --- |
| **Player** |
| Case\_amount  Case\_checking  Owner\_case\_amount  My\_case |
| Instruction()  Pick\_my\_case()  Case\_removing()  Random() |

**Random:**

Random method is used for selecting different money from the list given to insert into the different briefcases.

|  |
| --- |
| self.Case = random.choice(self.Case\_amount) |

**Validate input:**

Here, we will get the input as string and we will check the input with our condition given in the program. If the input satisfies the given condition, then the statements executes continuously one by one, else that statement will be repeated until we give the correct input.

**For example code:**

def Instruction(self):

self.Instruction1 = input()

if self.Instruction1 == '1':

self.pick\_my\_case()

elif self.Instruction1 == '0':

self.pick\_my\_case()

else:

print("Press 1 or 0")

self.Instruction()

|  |
| --- |
| **Bankers** |
| Case\_amount  Descision  Final\_choice  Case\_checking |
| Banker()  Case\_removing()  Exit() |

**Banker:**

Here banker method uses simple mathematical calculation to offer amount to the player at each level.

Mathematical Calculation: Sum of all the amounts present in the list divide by the number of amounts present in the list

|  |
| --- |
| Banker offer = sum(Case amount) / float( len (Case amount)) |

**3. SYSTEM ANALYSIS**

**3.1 Hardware Requirements**

| **Requirement** | **Minimum** | **Recommended** |
| --- | --- | --- |
| Memory | 512 MB | 1 GB |
| Free disk space | 300 MB | 1 GB |

**3.2 Software Requirements**

* Needs Python interpreter and Eclipse Software with pydev plug-in
* Operating System :
* Microsoft Windows 7 or later and Linux
* Python Version : 3.6 needed and Eclipse : Oxygen version needed

**4. MODULE DESCRIPTION:**

**4.1. Module 1: Instruction**

In this module, Instructions are given to the players, like how to play the game, how to select the briefcases, and then how to remove boxes/ cases

If the player does not need any instruction, then the player can directly involved into the Game by pressing 0

**4.2. Module 2: pick my case**

In this module, the player should select their own Briefcase, which contains hidden amount. The amount is placed in the brief Case at each time differently by using randomized method. There will be 1 – 26 briefcases and the amount ranges from Rs. 1 to Rs.1000000. From these the player will select / pick up a box.

**4.3. Module 3: Removing the case**

After selecting our case, the player should remove the next upcoming cases one by one and it’s amount from the list. The selected box should be Valid, if it is invalid or if the case is already removed repeat the same step to take the valid box/case

**4.4. Module 4: Banker’s calculation**

After finishing a level, Banker function should calculate the banker’sAmount / offer. After calculating the amount, if the player likes to deal with the Banker means then they can take the banker’s amount and leave their own case, or If they need to play the game still end then they can take their selected hidden money in that case

|  |
| --- |
| Banker offer = sum(Case amount) / float( len (Case amount)) |

**5. APPENDIX**

**5.1. Source code**

import random

print("\tWelcome to Deal or No\_Deal game\n Press any key to enter into the game")

Key = input()

#creating a class

class Player():

def \_\_init\_\_(self):

self.Case\_amount = [1, 5, 10,25, 50, 75, 100, 200, 300, 400, 500, 600, 750, 1000, 5000, 10000, 25000, 50000, 75000, 100000, 200000, 300000, 400000, 500000, 750000, 1000000]

self.Case\_checking = []

self.Owner\_case\_amount = []

self.Instruction()

def Instruction(self):

print("If you need Instruction press 1 else Press 0")

self.Instruction1 = input()

if self.Instruction1 == '1':

print("This Game has 6 levels\nThere are 26 Cases and Each case contain some Amounts\nYou need to pick up a Briefcase and keep it with u\nThen remove the cases one by one\nAt the end of Each level there will be a Banker, Who offer you some amount to leave the game\nIf you press Deal then you won Banker's amount else U will continue the game\nAtlast If U Rejected All the Banker's Offer Then you will get ur Case Amount\n\n Let's start the game")

self.pick\_my\_case()

elif self.Instruction1 == '0':

print("\tYou have Entered into the Game\n\n\tStarts to play")

self.pick\_my\_case()

else:

print("Press 1 or 0")

self.Instruction()

def pick\_my\_case(self):

self.My\_case = input("\nChoose your Case from 1 to 26 cases: ")

#Checking the input is correct or not

if int(self.My\_case) > 26 or int(self.My\_case) < 1:

print("Pick a Valid Case")

self.pick\_my\_case()

else:

self.Case = random.choice(self.Case\_amount)

#removing that amount from the cases to avoid repetition of same amount

self.Case\_amount.remove(self.Case)

#adding our amount in our case in separate variable

self.Owner\_case\_amount.append(self.Case)

#adding the selected case in another variable

self.Case\_checking.append(self.My\_case)

#to remove these cases from the list directing to other function

self.Case\_removing(self.My\_case, self.Case\_checking)

def Case\_removing(self,My\_case, Case\_checking):

self.Turn = 4

print("\tHint: There are only 1 - 26 boxes choose one from these boxes")

#loop to remove the cases and its amount from the list

while self.Turn != 0:

self.Remove\_case = input("Choose a case to be removed: ")

if int(self.Remove\_case) > 26 or int(self.Remove\_case) < 1:

print("Select a valid Case")

elif self.Remove\_case in self.Case\_checking:

print("Case already taken")

else:

self.Case = random.choice(self.Case\_amount)

self.Case\_amount.remove(self.Case)

self.Case\_checking.append(self.Remove\_case)

print("Removed case %s ,contains Rs. %.2f"%(self.Remove\_case, self.Case))

self.Turn = self.Turn - 1

Bankers.Banker(self, My\_case, Case\_checking)

class Bankers():

def \_\_init\_\_(self, My\_case, Case\_checking):

self.My\_case = My\_case

self.Case\_checking = Case\_checking

def Banker(self,My\_case, Case\_checking):

#calculate banker's amount

self.Banker\_offer = sum(self.Case\_amount)/float(len(self.Case\_amount))

print("Hint: if Deal then Press Y \n\telse\n\t Press N for No\_Deal")

if len(self.Case\_amount) == 1:

self.Final\_choice = input("Would you like to keep your case\n\tDeal[Y] or NODeal[N]")

if self.Final\_choice == 'y' or self.Final\_choice == 'Y':

print("You Won and Taken your case amount Rs. %.2f"%sum(self.Owner\_case\_amount))

print ("------------Thank You For Playing--------------\n\t\tPress Ctrl + F11 to play again")

exit()

elif self.Final\_choice == 'n' or self.Final\_choice == 'N':

print("You accepted the Banker's offer Rs. %.2f"%self.Banker\_offer)

print ("------------Thank You For Playing--------------\n\t\tPress Ctrl + F11 to play again")

exit()

else:

print("Press Y or N")

self.Banker(self.My\_case, self.Case\_checking)

if len(self.Case\_amount) > 1:

print("The Banker Offer is Rs. %.2f"%self.Banker\_offer,"\nWill You going to take this Amount\n\tDeal[y] or No\_Deal[n]")

self.Descision = input()

if self.Descision == 'y' or self.Descision =='Y':

print("You have accepted the Banker's Offer and won Rs. %.2f"%self.Banker\_offer)

print("You Rejected Your Case and it contains Rs. %.2f"%sum(self.Owner\_case\_amount))

print ("------------Thank You For Playing--------------\n\t\tPress Ctrl + F11 to play again")

exit()

elif self.Descision == 'n' or self.Descision =='N':

Player.Case\_removing(self,My\_case, Case\_checking)

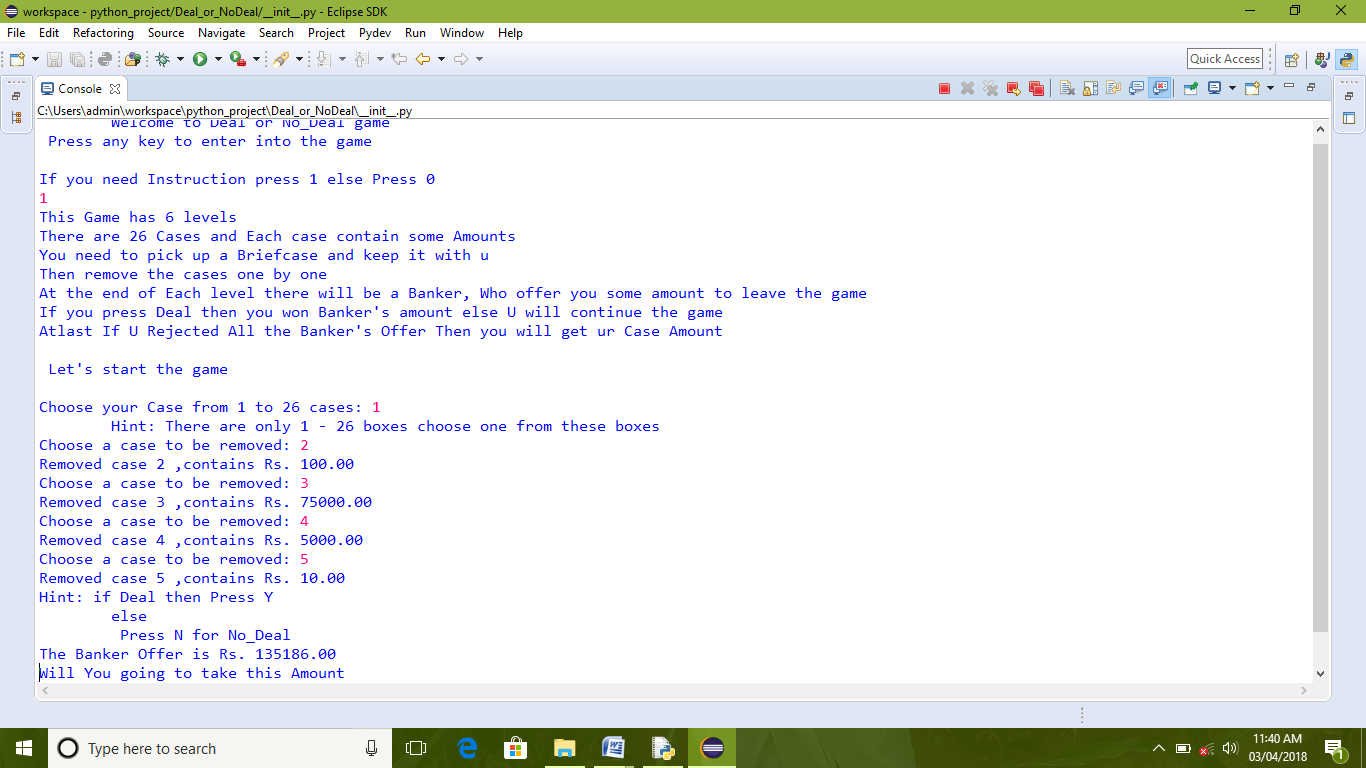
else:

print("Press Y or N")

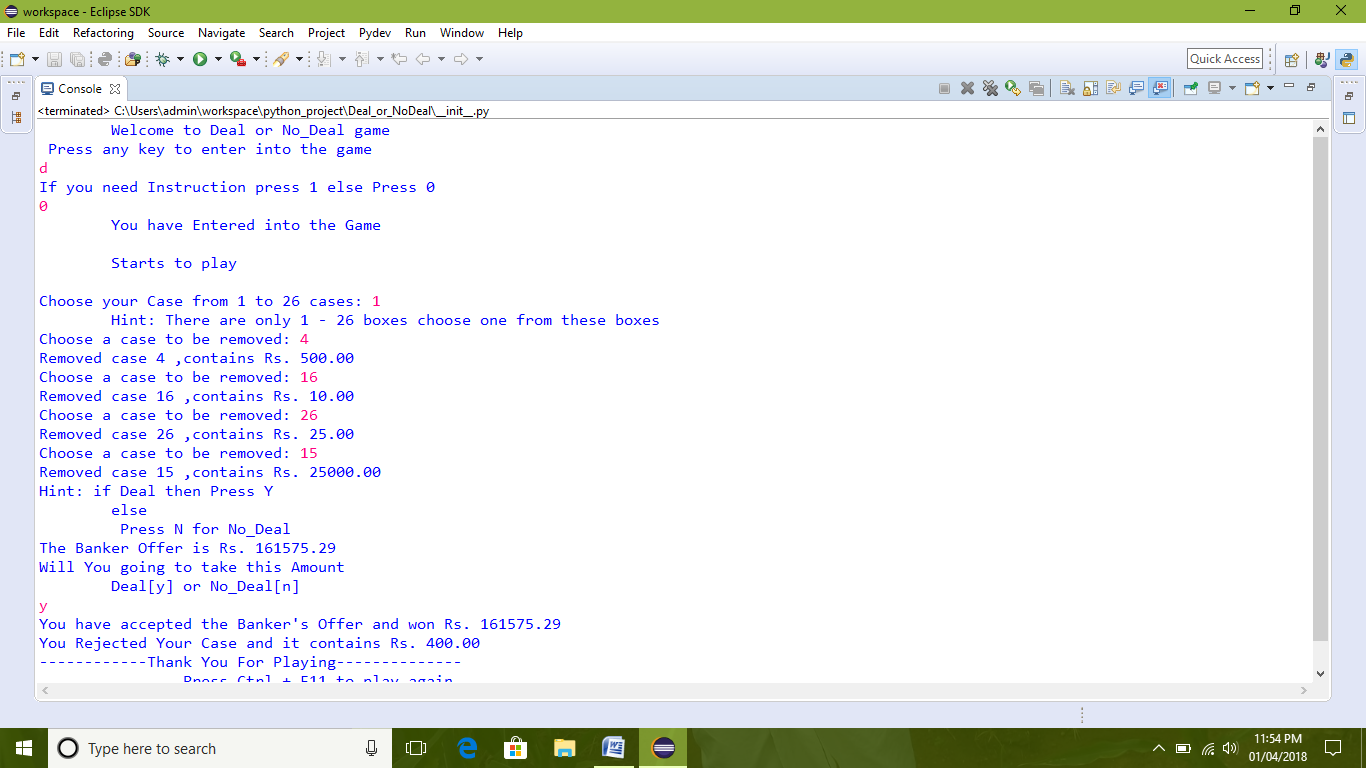
self.Banker(self.My\_case, self.Case\_checking)

Player()

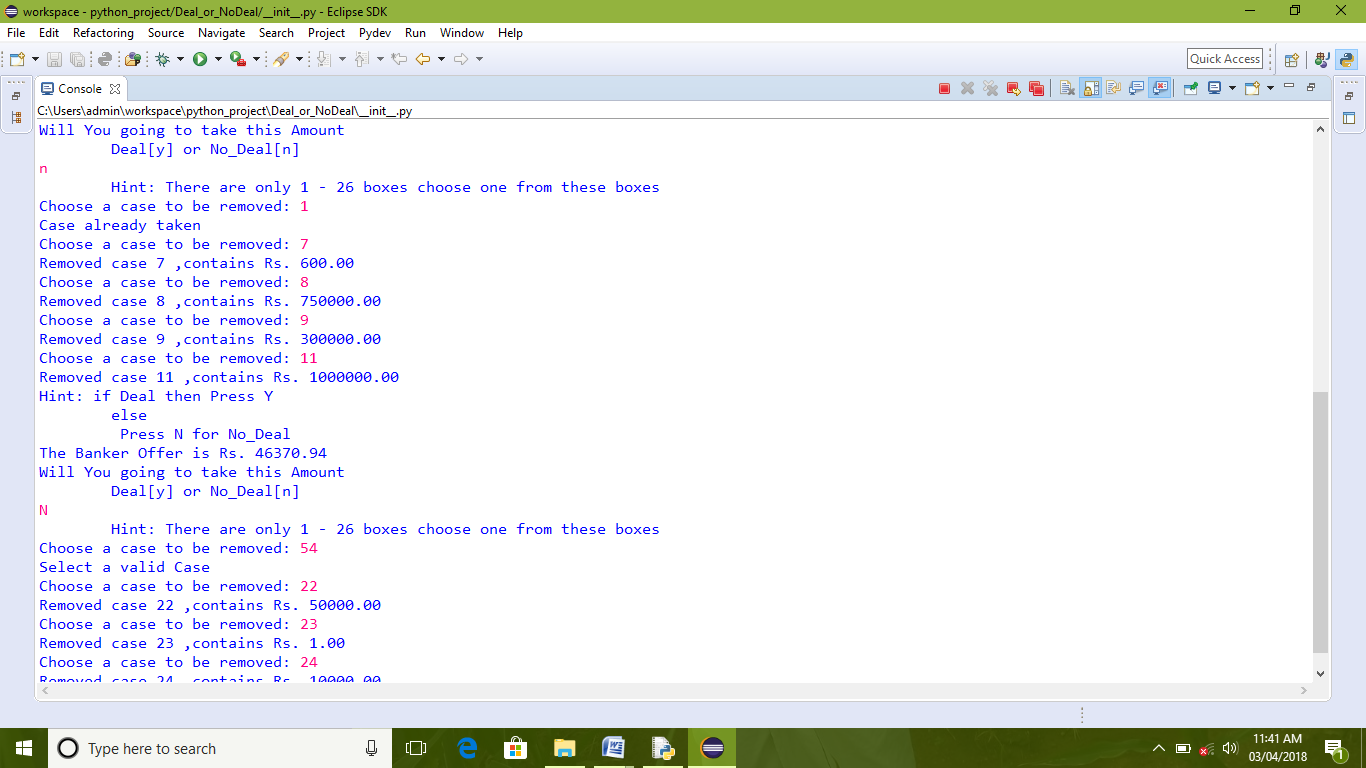
**5.2. SCREEN SHOT**



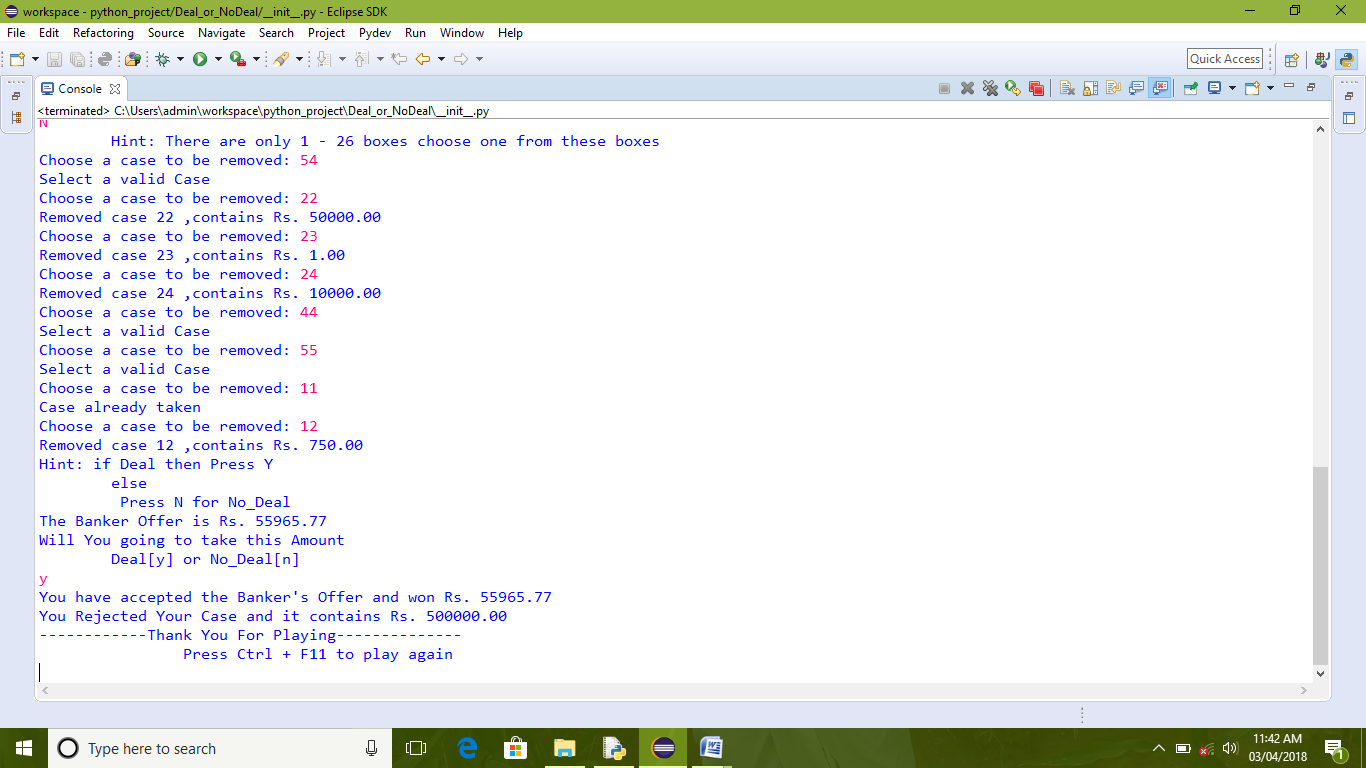
**Screenshot 1: Instruction Page**



**Screenshot 2: Choosing our briefcase**



**Screenshot 3: No dealing with banker**



**Screenshot 4: Dealing with the banker**

**6. CONCLUSION**

From this Deal or No Deal game, we will came to know about our luck, and we are going to make this game so interesting, so everyone will be addicted in my game. This game is time consuming. If anyone seems to be bored go and give this game then he/she will not come out from my game. If anyone finds the trick of Banker then they can solve this game easily and they can win the game with higher amount too. I am concluding by sayings that play my game and enjoy with your luck today.