Semester Project Report

Computer Programming – I (MCT – 242)

**Shut the Box Game**



Submitted By: 2018-MC-73 Shahzad Niaz

Submitted To: Mr. M. Ahsan Naeem

**DEPARTMENT OF MECHATRONICS & CONTROL ENGINEERING**

**University of Engineering and Technology, Lahore.**

December 17, 2019

Table of Contents

[**1.** **Abstract** 3](#_Toc27344861)

[**2.** **Introduction** 3](#_Toc27344862)

[**2.1.** **Main Idea** 3](#_Toc27344863)

[**2.2.** **Features** 3](#_Toc27344864)

[**3.** **Methodology** 4](#_Toc27344866)

[**3.1.** **Algorithm of Program** 4](#_Toc27344867)

[**4.** **Conclusions and Future Enhancements** 5](#_Toc27344868)

[**5.** **Recommendations** 6](#_Toc27344869)

[**6.** **Appendix** 6](#_Toc27344870)

# **Abstract**

This project is about a game called shut the box. This program is just a basic model of the game allowing the user to play the game using dices. This game has 2 modes; two player mode and a four-player mode. In four player mode the player having the lowest score wins. As for the two-player mode one of the players will have to shut the box in order to win.

# **Introduction**

## **Main Idea**

The main idea of this project is to provide the user with the computerized version of the game using python. The game is mostly dependent on the probability of the rolled dices and the choice of user, as he will choose how many tiles and which ones he wants to low. Moreover, logically this project is quite conceptual as use of logic is main part of it.

## **Features**

This program is very user friendly and easy to use. When user starts the program, it will show the previous lowest score and asks the user which mode he wants to play (2 or 4 player). User can’t enter modes other than these two. In 4 player mode 4 players will play turn by turn and the player with the lowest score will win. Whereas in 2 player mode, player will win when he shuts the box, a condition if either one of the players lowers all the tiles or ups all the tiles.

The advanced aspects can involve the exception handling used in the program as the user Is only allowed to enter numbers only and in some cases number between ranges. The user has the option to choose whether he wants to play with one dice or two when he lowers the tiles 7,8 and 9. If any of the players shuts the box then he will win the game.

An example can be like if user selected 4 player mode. First player 1 will play and rest will follow. If player 1 rolls 10, he will see the numbers present to low in list and he will choose the combination he can use to make the sum of selected tiles equal to 10. For example, he can choose 9,1 or 8,2 or 5,4,1 or 6,3,1 or 4,3,2,1.

# **Methodology**

This program has user define functions and different condition used for the production of the game. It has mainly two parts; functions and main code explained step by step in the algorithm.

## **Algorithm of Program**

Check function:

This function is used to low or up the tiles selected by the user.

toLow1 and toLow2 functions:

These functions are used in two player mode. These functions have the same function but 1 lows the tiles selected which goes into a specific list. Function 2 ups the tiles selected by player 2 in the list having lowered tiles.

Shut box function:

It is used to check the shut the box condition. It will be valid when list has all tiles lowered or its sum is zero.

One dice and two dice functions:

These functions are used to give out sum of dice or dices in every turn. Two dice gives sum of two random dice values whereas one dice gives one random dice value.

Ifsumexists function:

This function is used to check if the sum of the rolled dice or dices is present in the tiles or list or any combination of it.

Getnumeric function:

This function is used to get only number from user nothing else can be input.

Now in main user will enter mode he wants to play.

In 4 player mode, Loop will run 4 times for 4 players.

Player having the lowest score will win.

Lowest score is stored in a file with time and date.

In 2 player mode first player 1 will low the tiles he wants to low. They will be store in a list. When turn is over player 1 will try to up the tiles lowered by player 1. This will continue until 1 of them gets all the tiles up or all the tiles low.

If user doesn’t select any of two modes he is asked if he wants to play again.

If user enters “r” he will be again asked to enter mode, if not game will end.

If user inputs anything except alphabet, he will be given three tries to enter an alphabet, after that game will end.

# **Conclusions and Future Enhancements**

I faced difficulty in the formation of ifsumexists function. It was most difficult part to check if the sum exists in the remaining tiles in the list. For that I used nested for loop with multiple if conditions. First of all it takes first element of the list given to it and checks if it is equal to zero or not. After that it again takes firt element of the list and check if it is equal to the sum, if sum is not present then it checks whether the two elements from the same list are equal or not , if they are equal it takes next element from the same list while the first element taken remains same. It then sums both and checks if it is equal to sum or not. On basis of that it returns true or false.

Future enhancement can be:

* Increase in number of tiles
* It can be made in GUI
* More than one rounds can be played and player having less score will be eliminated
* More players can play

# **Recommendations**

Overall lab was a great but just at the end the time was short compared to the complexity of the manuals.

# **Appendix**

1. #Shehzad Niaz , 2018-mc-73 , sec B
2. import random,os,time
3. def check(s,r): #lows the plate selected in 4 player mode
4. while(s>0):
5. i=r-1
6. if(s==list1[i]):
7. list1[i]='-'
8. s=s-r
9. return s
10. else:
11. s=s-r
12. list1[i]='-'
13. return s
14. def toLow1(s,r):    #lows the plate selected in 2 player mode
15. while(s>0):
16. i=r-1
17. if(s==list1[i]):
18. list1[i]='-'
19. list3[i]=s
20. s=s-r
21. return s
22. else:
23. s=s-r
24. list1[i]='-'
25. list3[i]=r
26. return s
27. def toLow2(s,r):    #ups the plate selected in 2 player mode
28. while(s>0):
29. i=r-1
30. if(s==list4[i]):
31. list4[i]='-'
32. list2[i]=s
33. s=s-r
34. return s
35. else:
36. s=s-r
37. list4[i]='-'
38. list2[i]=r
39. return s
40. def shut\_box(list1):    #checks shut the box condition
41. s2=0
42. for i in list1:
43. if(i!="-"):
44. s2+=i
45. if(s2==0):
46. return True
47. def two\_dice():     #gives sum of two dices
48. d1=random.randint(1,6)
49. d2=random.randint(1,6)
50. s=d1+d2
51. return s
52. def one\_dice():     #gives sum of one dice
53. d1=random.randint(1,6)
54. s=d1
55. return s
56. def ifSumExists(Sum,s,list1):   #checcks if sum exists in the list
57. for i in list1:
58. if(i!='-'):
59. for x in list1:
60. if(x!='-'):
61. if(s==x):
62. Sum=True
63. return Sum
64. if(x!=i):
65. sum2=i+x
66. if(s==sum2):
67. Sum=True
68. return Sum
69. for y in list1:
70. if(y!=i):
71. if(x!=y):
72. if(y!='-'):
73. sum3=sum2+y
74. if(s==sum3):
75. Sum=True
76. return Sum
77. for z in list1:
78. if(z!=y):
79. if(x!=z):
80. if(z!=i):
81. if(z!='-'):
82. sum4=sum3+z
83. if(s==sum4):
84. Sum=True
85. return Sum
86. def getnumeric(lol):    #for exception handling
87. while(True):
88. response=input(lol)
89. try:
90. return int(response)
91. except ValueError:
92. print("please enter a number")
93. #asking for mode 4 players or 2
94. print("WELCOME TO SHUT THE BOX")
95. time.sleep(1)
96. os.system("cls")
97. print("Select mode:")
98. repeat="r"
99. time.sleep(1)
100. os.system("cls")
101. score={}
102. win=0
103. count=0
104. s2=s1=0
105. while(repeat=="r" or repeat=="R"):
106. file=open("Lol.txt","r")        #reads file
107. for i in file:
108. print(i)
109. mode=getnumeric("How many players are playing? (2 or 4): ") #asks for mode
110. time.sleep(1)
111. os.system("cls")
112. if(mode==4):  #for 4 players
113. for j in range(1,5):    #for turns
114. dice=True
115. choice=2
116. list1=[1,2,3,4,5,6,7,8,9]
117. print("It is player "+str(j)+" turn")
118. while True:
119. Sum=False
120. check1=True
121. list2=[]
122. for i in list1:
123. list2.append(i)
125. for i in range(6,9,1):
126. if( list1[i] != '-'):
127. check1=False
128. break
130. if(dice):
131. if(check1):
132. choice=getnumeric("How many dices you want to use: ")
133. dice=False
134. if(choice==2):
135. s=two\_dice()
136. else:
137. s=one\_dice()
138. print("You rolled :",s)
139. Sum=ifSumExists(Sum,s,list1)
141. if(Sum):
142. low=getnumeric("How many plates you want to lower: ")
143. while(low>4):
144. low=getnumeric("How many plates you want to lower: ")
145. for i in range(low):
146. while True:
147. r=getnumeric("Which plate you want to lower: ")
148. index=r-1
149. if(not r>9):
150. if(r==list1[index]):
151. s=check(s,r)
152. break
153. elif(r>s):
154. print("plate value is not present sum")
155. else:
156. print("plate is not present to low")
157. if(s==0):
158. print(list1)
159. if(s!=0):
160. print(list2)
161. print("Your turn is over.")

164. for i in list2:
165. if(i!="-"):
166. s1+=i
167. s2=s1
168. print("Score of player "+str(j)+" is :",s1)
169. score["Player"+str(j)]=s1
170. s1=0
171. break
172. else:
173. print("no sum")
174. print("Your turn is over.")
175. for i in list2:
176. if(i!="-"):
177. s1+=i
178. s2=s1
179. print("Score of player "+str(j)+" is :",s1)
180. score["Player"+str(j)]=s1
181. s1=0
182. break
183. time.sleep(3)
184. os.system("cls")
185. if(shut\_box(list1)):
186. print("You have shut the box!!")
187. print("Player "+str(j)+" is winner!")
188. break
189. print(score.values())   #stores the score of players
190. win=min(score.values())
191. print("Winner is",[k for k,v in score.items() if v==win]," with score",win)
192. file=open("Lol.txt","w")
193. file.write("The lowest score is")
194. file.write(" ")
195. file.write(str(win))
196. file.write(" ")
197. file.write(time.ctime())
198. file.close()
199. elif(mode==2):#for 2 players
200. for i in range(2):
201. list3=['-','-','-','-','-','-','-','-','-'] #list for player 2
202. list1=[1,2,3,4,5,6,7,8,9]   #list for player 1
203. while True:
205. while True:
206. Sum=False
207. list2=[]
208. list4=[]
209. for i in list1:
210. list2.append(i)
211. for i in list3:
212. list4.append(i)
213. print("It is player 1 turn")
214. s=two\_dice()
215. print("You rolled :",s)
216. Sum=ifSumExists(Sum,s,list1)
218. if(Sum):
219. low=getnumeric("How many plates you want to lower: ")
220. while(low>4):
221. low=getnumeric("How many plates you want to lower: ")
222. for i in range(low):
223. while True:
224. r=getnumeric("Which plate you want to lower: ")
225. index=r-1
226. if(not r>9):
227. if(r==list1[index]):
228. s=toLow1(s,r)
229. break
230. elif(r>s):
231. print("plate value is not present sum")
232. else:
233. print("plate is not present to low")
234. if(s==0):
235. print(list1)
236. if(s!=0):
237. print(list2)
238. print("Your turn is over.")
239. print(list4)
240. break
241. if(shut\_box(list1)):
242. print("You have shut the box!!")
243. print("Player 1 is winner!")
244. break
245. else:
246. print("no sum")
247. print("Your turn is over.")
248. print(list4)
249. break
250. if(shut\_box(list1)):
251. break
252. while(True):
253. Sum=False
254. list1=[]
255. for i in list2:
256. list1.append(i)
257. list3=[]
258. for i in list4:
259. list3.append(i)
261. print("player 2 turn")
262. #print(list3)
263. s1=two\_dice()
264. print("You rolled :",s1)
265. Sum=ifSumExists(Sum,s1,list4)
266. if(Sum):
267. low=getnumeric("How many plates you want to up: ")
268. while(low>4):
269. low=getnumeric("How many plates you want to up: ")
270. for i in range(low):
271. while True:
272. r=getnumeric("Which plate you want to up: ")
273. index=r-1
274. if(not r>9):
275. if(r==list4[index]):
276. s1=toLow2(s1,r)
277. break
278. elif(r>s1):
279. print("plate value is not present sum")
280. else:
281. print("plate is not present to low")
282. if(s1==0):
283. print(list4)
284. if(s1!=0):
285. print(list3)
286. print("Your turn is over.")
287. print(list1)
288. break
289. if(shut\_box(list4)):
290. print("You have shut the box!!")
291. print("Player 2 is winner!")
292. break
294. else:
295. print("no sum")
296. print("Your turn is over.")
297. print(list1)
298. break
300. if(shut\_box(list4)):
301. break
302. else:
303. print("Only 2 or 4 players can play!")
304. repeat=input("Enter R if you play again!")  #asking if he wants to play again
305. while(repeat.isalpha()==False and count<2): #if user doesnot enters alphabet
306. print("Enter alphabets only")
307. repeat=input("Enter R if you play again: ")
308. count+=1
309. print("You dont want to play more")
310. print("Goodbye")