

บันทึกเกม rock paper scissor



Game.py

```
1  from kivy.app import App
2  from kivy.uix.screenmanager import Screen, ScreenManager
3  from kivy.lang import Builder
4  from kivy.core.window import Window
5  from kivy.uix.button import ButtonBehavior
6  from kivy.uix.image import Image
7  from kivy.animation import Animation as an
8  from random import randint, choice
9  from kivy.clock import Clock
10 from time import sleep
11 from kivy.uix.popup import Popup
12 from kivy.utils import get_color_from_hex
13 from kivy.uix.floatlayout import FloatLayout
14 from kivy.uix.label import Label
15 from kivy.uix.button import Button
16
17
```

```
18 class ImageBtn(ButtonBehavior, Image):
19     def __init__(self, **kwargs):
20         super(ImageBtn, self).__init__(**kwargs)
21         self.allow_stretch= True
22         self.keep_rotio= False
23
24
```

```

25 class Popups(Popup):
26     def __init__(self, title, txt, **kwargs):
27         super(Popups, self).__init__(**kwargs)
28         self.size_hint= [None, None]
29         self.size= [300, 140]
30         self.pos_hint= {'top':.7, 'center_x':.5}
31         self.title= title
32         self.title_size= '15sp'
33         self.separator_height= 5
34         self.separator_color= get_color_from_hex('#FFFFFF')
35         self.the_container= FloatLayout(size_hint=[1,1])
36         self.warning= Label(text=txt.title(), font_size='18sp', bold=True, size_hint=[.9, .3],
37                             pos_hint={'center_y':.6, 'center_x':.5})
38         if title == 'Exit!':
39             self.height= 180
40             self.auto_dismiss= False
41             self.btn1= Button(text='cancel', size_hint=[.4, .2], font_size='15sp',
42                             pos_hint={'top':.22, "x":.05}, background_normal="", background_color=[0,0,0,1])
43             self.btn2= Button(text='yes', size_hint=[.4, .2], font_size='15sp',
44                             pos_hint={'top':.22, "x":.55}, background_normal="", background_color=[0,0,0,1])
45             self.btn1.bind(on_release= lambda x: self.dismiss())
46             self.btn2.bind(on_release= lambda x: app.kill_app())
47             self.the_container.add_widget(self.btn1)
48             self.the_container.add_widget(self.btn2)
49
50
51         self.the_container.add_widget(self.warning)
52         self.add_widget(self.the_container)
53

```



```
54 class ManagerSc(ScreenManager):
55     pass
56 class HomePage(Screen):
57     pass
58 class PlayPage(Screen):
59     pass
60
```

```
62 class MainApp(App):
63     clicked= ''
64     clocks= 0
65     opp= ""
66     you= ""
67
68     def build(self):
69         Window.size=(360, 640)
70         return Builder.load_file('game.kv')
71
72     def reset_all(self):
73         self.clicked= ''
74         self.clocks= 0
75         self.opp= ""
76         self.you= ""
77         self.root.ids.play.ids.opponent.source= 'imgs/empty.png'
78         self.root.ids.play.ids.you.source= 'imgs/empty.png'
79         self.root.ids.play.ids.score.text= 'score'
80         self.root.ids.play.ids.score.szz= '45sp'
81         self.root.ids.play.ids.score.clr= [0,0,0,.3]
82
```

```
83     def process(self, me, him):
84         self.res= ""
85         self.me= me
86         self.him= him
87         for i in range(3):
88             if self.me == 'scissors' and self.him == 'scissors':
89                 self.res= 'tie'
90             if self.me == 'scissors' and self.him == 'paper':
91                 self.res= 'win'
92             if self.me == 'scissors' and self.him == 'rock':
93                 self.res= 'foe'
94             if self.me == 'paper' and self.him == 'scissors':
95                 self.res= 'foe'
96             if self.me == 'paper' and self.him == 'paper':
97                 self.res= 'tie'
98             if self.me == 'paper' and self.him == 'rock':
99                 self.res= 'win'
100            if self.me == 'rock' and self.him == 'scissors':
101                self.res= 'win'
102            if self.me == 'rock' and self.him == 'paper':
103                self.res= 'foe'
104            if self.me == 'rock' and self.him == 'rock':
105                self.res= 'tie'
106        return self.res
107
```

```
108 def animate_go_btn(self, target, what):
109     if what == 1:
110         anime= an(sz=[.7, .06], clr=[1,0,0,.8], ps={'top':.08, 'x':.15},
111                 d=0.1, c= [1,0,0,.8])
112         anime.start(target)
113     elif what == 2:
114         animea= an(sz=[.9, .08], clr=[1,1,1,1], ps= {'top':.1, 'x':.05},
115                 d=0.5, c= [0,0,0,1])
116         animea.start(target)
117
118 def play(self):
119     self.bomb= Clock.schedule_interval(self.change_img, 0.2)
120
121 def change_img(self, *args):
122     self.clocks+= 1
123     if self.clocks <= 20:
124         opponent= self.root.ids.play.ids.opponent
125         self.one= {1:'rocke.png', 2:'pepper.png', 3:'sis.png'}[randint(1,3)]
126         opponent.source= self.one
127     else:
128         self.bomb.cancel()
129         if self.one == 'rocke.png':
130             self.opp= 'rock'
131         elif self.one == 'pepper.png':
132             self.opp= 'paper'
133         elif self.one == 'sis.png':
134             self.opp= 'scissors'
135
136     self.get_res()
```



```
138     def get_res(self):
139         if self.you != '' and self.opp != '':
140             self.the_res= self.process(self.you, self.opp)
141             self.show_res(self.the_res)
142
143         else: pass
144
145     def show_res(self, res, *args):
146         self.board= self.root.ids.play.ids.score
147         if res == 'win':
148             self.board.text= 'you won'.upper()
149             self.board.clr= [0,1,0,1]
150             self.board.szz= '60sp'
151
152         elif res == 'foe':
153             self.board.text= 'you lost'.upper()
154             self.board.clr= [1,0,0,1]
155             self.board.szz= '60sp'
156
157         elif res == 'tie':
158             self.board.text= 'a tie'.upper()
159             self.board.clr= [0,0,0,1]
160             self.board.szz= '60sp'
161         self.you= ''
162         self.opp= ''
163         print(res)
164
```

```
165     def show_popup(self, why_popup):
166         if why_popup == 'BtnUnclicked':
167             pop= Popups(title='little warning!',
168                         txt='you should select one\nof the three options first...')
169         elif why_popup == 'exit':
170             pop= Popups(title='Exit!',
171                         txt='are you sure you want to exit?')
172
173         pop.open()
174
175     def kill_app(self):
176         self.stop()
177
178
179 if __name__ == '__main__':
180     app= MainApp()
181     app.run()
```

Game.ky

```
1  #:kivy 1.11.1
2  #: import utils kivy.utils
3
4  ManagerSc:
5      id: screenmanager
6      HomePage:
7          id: home
8      PlayPage:
9          id: play
10
11  <Btn@Button>:
12      background_norml: ''
13      background_color: utils.get_color_from_hex('#FFB13B')
14      background_down: ''
15
16
```

```
17 <HomePage>:
18     name: 'home'
19     FloatLayout:
20         canvas.before:
21             Color:
22                 rgba: utils.get_color_from_hex('#730000')
23             Rectangle:
24                 size: self.size
25                 pos: self.pos
26             Color:
27                 rgba: utils.get_color_from_hex('#350505')
28             Triangle:
29                 points: [0,root.size[1]*3/4, root.size[0],root.size[1]*3/4, root.size[0]/2,0]
30             Color:
31                 rgba: utils.get_color_from_hex('#350505')
32             Rectangle:
33                 size: root.size[0], root.size[1]/4
34                 pos: 0, root.size[1]*3/4
35             Color:
36                 rgba: utils.get_color_from_hex('#730000')
37             Rectangle:
38                 size: root.size[0]*3/4, root.size[1]/7
39                 pos: root.size[0]/8, (root.size[1]*3/4)+(root.size[1]/14)
```



```
40     Label:
41         # this is a comment
42         size_hint: None, None
43         pos_hint: {'center_x':.5, 'top':.98}
44         size: self.texture_size
45         font_size: "95sp"
46         bold: True
47         color: [0,0,0,1]
48         text: 'R[color=#350505].[/color]P[color=#350505].[/color]S'
49         markup: True
50     Label:
51         text: '[b]R[/b]ock, [b]P[/b]aper, [b]S[/b]cissors'
52         size_hint: None, None
53         size: self.texture_size
54         pos_hint: {'top':.82, 'center_x':.5+.01}
55         font_size: '25sp'
56         markup: True
57         color: utils.get_color_from_hex('#730000')
```

```
58     Btn:
59         text: 'start the game'.title()
60         font_size: '35sp'
61         bold: True
62         color: [0,0,0,1]
63         size_hint: .8, .1
64         pos_hint: {'top':.46, 'x':.1}
65         on_press:
66             self.background_color= utils.get_color_from_hex('#730000')
67         on_release:
68             self.background_color= utils.get_color_from_hex('#FFB13B')
69             app.root.current='play'
70             root.manager.transition.duration= 1
71             root.manager.transition.direction= 'up'
72
73     Btn:
74         text: 'exit the game'.title()
75         font_size: '35sp'
76         bold: True
77         color: [0,0,0,1]
78         size_hint: .8, .1
79         pos_hint: {'top':.16, 'x':.1}
80         on_press:
81             self.background_color= utils.get_color_from_hex('#730000')
82         on_release:
83             self.background_color= utils.get_color_from_hex('#FFB13B')
84             app.show_popup('exit')
85
```

```
86 <PlayPage>:
87     id: play
88     opponent: opponent
89     name: 'play'
90     FloatLayout:
91         opponent: opponent
92         canvas.before:
93             Color:
94                 rgba: utils.get_color_from_hex('#730000')
95             Rectangle:
96                 size: self.size
97                 pos: self.pos
98             Color:
99                 rgba: utils.get_color_from_hex('#FFFFFF')
100             Rectangle:
101                 size: root.size[0], root.size[1]/5
102                 pos: 0, root.size[1]*5.8/8
103             Color:
104                 rgba: utils.get_color_from_hex("#000000")
105             Line:
106                 points: [(0,root.size[1]*5.8/8), (root.size[0],root.size[1]*5.8/8)]
107                 width: 5
108             Color:
109                 rgba: utils.get_color_from_hex("#000000")
110             Line:
111                 points: [(0,(root.size[1]*(5.8/8))+root.size[1]/5), (root.size[0],(root.size[1]*(5.8/8))+root.size[1]/5)]
112                 width: 5
113             Color:
114                 rgba: utils.get_color_from_hex("#000000")
115             Line:
116                 points: [(root.size[0]/2,(root.size[1]*(5.8/8))), (root.size[0]/2,(root.size[1]*(2.7/8)))]
117                 width: 3
118             Color:
119                 rgba: utils.get_color_from_hex("#000000")
120             Line:
121                 points: [(0,(root.size[1]*(2.7/8))), (root.size[0],(root.size[1]*(2.7/8)))]
122                 width: 3
```

```
123     Label:
124         id: score
125         clr: [0,0,0,.3]
126         szz: '45sp'
127         text: "the score"
128         size_hint: None, None
129         size: self.texture_size
130         color: self.clr
131         font_size: self.szz
132         pos_hint: {'center_x':.5, "top":.88}
133         bold: True
134
135     Button:
136         text: 'Go Back'
137         color: [0,0,0,1]
138         size_hint: .15, .05
139         pos_hint: {'x':.03, 'top':.99}
140         background_normal: ''
141         background_down: ''
142         background_color: utils.get_color_from_hex('#730000')
143         bold: True
144         underline: True
145         on_press:
146             self.color= utils.get_color_from_hex('#FFB13B')
147             score.clr= [0,0,0,.3]
148             score.szz= '45sp'
149             score.text= 'the score'
150         on_release:
151             self.color= utils.get_color_from_hex('#000000')
152             app.root.current='home'
153             root.manager.transition.duration= 1
154             root.manager.transition.direction= 'down'
155             app.reset_all()
```



```
157 Button:
158     clr: [1,1,1,1]
159     sz: [.9, .08]
160     ps: {'top':.1, 'x':.05}
161     c: [0,0,0,1]
162     text: 'GO'
163     font_size: '30sp'
164     bold: True
165     color: self.c
166     background_normal: ''
167     background_down: ''
168     background_color: self.clr
169     size_hint: self.sz
170     pos_hint: self.ps
171     on_press:
172         app.animate_go_btn(self, 1)
173     on_release:
174         app.animate_go_btn(self, 2)
175         app.play() if app.clicked == 'yes' else app.show_popup('BtnUnclicked')
176         app.clocks=0
177
```

```
178     BoxLayout:
179         orientation: 'horizontal'
180         size_hint: 1, .38
181         pos_hint: {'top':.72}
182         padding: 10
183         spacing: 20
184         BoxLayout:
185             orientation: 'vertical'
186             Label:
187                 text: 'you'
188                 font_size: '30sp'
189                 size_hint_y: None
190                 height: self.texture_size[1]
191                 bold: True
192             Image:
193                 id: you
194                 source: "ໄມ້ມື້໑.png"
195                 allow_stretch: True
196                 keep_ratio: False
197         BoxLayout:
198             orientation: 'vertical'
199             Label:
200                 text: 'opponent'
201                 font_size: '30sp'
202                 size_hint_y: None
203                 height: self.texture_size[1]
204                 bold: True
205             Image:
206                 id: opponent
207                 source: "ໄມ້ມື້໑.png"
208                 allow_stretch: True
209                 keep_ratio: False
```

```
210     BoxLayout:
211         orientation: 'vertical'
212         size_hint: 1, .19
213         pos_hint: {'top': .34}
214         spacing: 5
215         padding: 10
216         Label:
217             text: 'select your option:'
218             font_size: '20sp'
219             size_hint: None, None
220             size: self.texture_size
221             color: [1,1,1,1]
222             bold: True
223         BoxLayout:
224             orientation: 'horizontal'
225             spacing: 5
226             ImageBtn:
227                 source: 'rocke.png'
228                 on_press:
229                     ro.color= [1,0,0,1]
230                 on_release:
231                     ro.color= [1,1,1,1]
232                     you.source= 'rocke.png'
233                     app.clicked= 'yes'
234                     app.you= 'rock'
```

```
235 ImageBtn:
236     source: 'sis.png'
237     on_press:
238         so.color= [1,0,0,1]
239     on_release:
240         so.color= [1,1,1,1]
241         you.source= 'sis.png'
242         app.clicked= 'yes'
243         app.you= 'scissors'
244 ImageBtn:
245     source: 'pepper.png'
246     on_press:
247         po.color= [1,0,0,1]
248     on_release:
249         po.color= [1,1,1,1]
250         you.source= 'pepper.png'
251         app.clicked= 'yes'
252         app.you= 'paper'
```



```
253     BoxLayout:
254         orientation: 'horizontal'
255         spacing: 5
256         padding: 10
257         size_hint: 1, .1
258         pos_hint: {"top":.2}
259         Label:
260             id: ro
261             text: 'Rock'
262             bold: True
263             font_size: '18sp'
264             size_hint_y: None
265             height: self.texture_size[1]
266         Label:
267             id: so
268             text: 'Scissors'
269             bold: True
270             font_size: '18sp'
271             size_hint_y: None
272             height: self.texture_size[1]
273         Label:
274             id: po
275             text: 'Paper'
276             bold: True
277             font_size: '18sp'
278             size_hint_y: None
279             height: self.texture_size[1]
```