

Sandbox | CMU CS Academy

```

1 #####
2 app.background = 'silver'
3 # defines whether the map has been drawn or not
4 app.isMapDrawn = False
5 # sets the speed of the onStep function
6 app.stepsPerSecond = 25
7 # establishes a current index
8 app.index = 0
9 app.colorIndex = 0
10 # true/false for if crazy map is on and if its top or bottom
11 app.isCrazyOn = False
12 app.topColors = False
13 app.bottomColors = False
14 # splashscreen, instructions
15 instructions = Group(
16     Label('Laser Tag!', 200, 50, size=35, bold=True, italic=True,
17         fill=gradient('green', 'red', 'green', start='top')),
18     Label('Press \'SPACE\' to start a game of laser tag.',
19         200, 350, size=18, bold=True, fill='red'),
20     Label('Game Rules and Controls:', 200, 110, size=25, bold=True),
21     Label('Player 1 is controlled with the arrow keys.', 200, 140, size=15, bold=True),
22     Label('Player 2 is controlled with the W, A, S, and D keys.',
23         200, 160, size=15, bold=True),
24     Label('Player 1 shoots their laser with the \'p\' key', 200, 180, size=15, bold=True),
25     Label('Player 2 shoots their laser with the \'q\' key', 200, 200, size=15, bold=True),
26     Label('NOTE!', 200, 230, size=40, fill='red', bold=True),
27     Label('Hitting the red DANGER ZONE will cause you to lose!',
28         200, 265, size=15, bold=True),
29     Label('Press \'C\' on your keyboard to make the top map crazy!!!',
30         200, 290, size=13, bold=True),
31     Label('Press \'V\' on your keyboard to make the bottom map crazy!!!',
32         200, 305, size=13, bold=True),
33     Label('(C) 2022 Grant Webb, CS 1', 200, 385, size=15)
34 )
35 # user 1 and user 2 drawing
36 user1 = Group(
37     Polygon(30, 380, 30, 360, 40, 355, 50, 360, 50, 380, fill='red', border='black'),
38     Label('1', 40, 367, size=13)
39 )
40 user2 = Group(
41     Polygon(360, 30, 360, 50, 370, 55, 380, 50, 380, 30, fill='red', border='black'),
42     Label('2', 370, 39, size=13)
43 )
44 user1.visible = False
45 user2.visible = False
46 # both lasers, and their dy's
47 laser1 = Rect(38, 325, 4, 25, fill=gradient('green', 'red'), visible=False)
48 laser2 = Rect(369, 60, 4, 25, fill=gradient('green', 'red'), visible=False)
49 laser1.dy = -20
50 laser2.dy = 20
51 # labels which indicate the health of each user
52 user2Health = Label(100, 35, 25, size=20, visible=False)

```

```

53 user1Health = Label(100, 270, 375, size=20, visible=False)
54 # circles and stars group for crazyColors distraction
55 circles = Group()
56 stars = Group()
57 def crazyColors(x, y, r, p):
58     # colors of shapes
59     shapeColors = ['red', 'orange', 'yellow', 'green', 'blue', 'purple']
60     # checks if isCrazyOn is True. if it is, begins loops for circles using shapeColors
61     list
62     if(app.isCrazyOn == True):
63         for color in shapeColors:
64             circles.add(Circle(x, y, r, fill = shapeColors[randrange(0, 6)],
65                             border = 'white', borderWidth = 3))
66         for color in shapeColors:
67             stars.add(Star(x, y, r, p, fill = shapeColors[randrange(0, 6)],
68                             border = 'white', borderWidth = 3))
69
70     app.index += 1
71
72     # increases radius of circles and stars. if too big, removes
73     for circle in circles.children:
74         circle.radius += 10
75         if(circle.radius > 50):
76             circles.remove(circle)
77
78     for star in stars.children:
79         star.radius += 10
80         if(star.radius > 50):
81             stars.remove(star)
82     # possible background colors
83     backgroundColors = ['silver', 'darkOrange', 'gold', 'darkGreen',
84                         'skyBlue', 'mediumPurple']
85     # cycles through background colors
86     if(app.index > len(backgroundColors) - 1):
87         app.index = 0
88     app.background = backgroundColors[app.index]
89
90 def onKeyPress(key):
91     # begins the game from splashscreen
92     if((key == 'space') and (app.isMapDrawn == False)):
93         instructions.visible = False
94         drawMap()
95         app.isMapDrawn = True
96
97     # shoots laser
98     if(key == 'p'):
99         laser1.centerX = user1.centerX
100        laser1.centerY = user1.centerY
101        laser1.visible = True
102        shootLaser1()
103    elif(key == 'q'):
104        laser2.centerX = user2.centerX
105        laser2.centerY = user2.centerY
106        laser2.visible = True
107        shootLaser2()
108    # turns crazy mode on and off for top part of map

```

```

109     if((key == 'c') and (app.isCrazyOn == False)):
110         app.isCrazyOn = True
111         app.topColors = True
112         print(app.isCrazyOn)
113     elif((key == 'c') and (app.isCrazyOn == True)):
114         app.isCrazyOn = False
115         app.topColors = False
116         print(app.isCrazyOn)
117     # turns crazy mode on and off for bottom part of map
118     if((key == 'v') and (app.isCrazyOn == False)):
119         app.isCrazyOn = True
120         app.bottomColors = True
121         print(app.isCrazyOn)
122     elif((key == 'v') and (app.isCrazyOn == True)):
123         app.isCrazyOn = False
124         app.bottomColors = False
125
126 def onKeyHold(keys):
127     # defines the keys that move user 1 and 2
128     if('up' in keys):
129         user1.centerY -= 5
130         user1.rotateAngle = 0
131     if('left' in keys):
132         user1.centerX -= 5
133         user1.rotateAngle = -90
134     if('right' in keys):
135         user1.centerX += 5
136         user1.rotateAngle = 90
137     if('down' in keys):
138         user1.centerY += 5
139         user1.rotateAngle = 180
140     if('w' in keys):
141         user2.centerY -= 5
142         user2.rotateAngle = 180
143     if('a' in keys):
144         user2.centerX -= 5
145         user2.rotateAngle = 90
146     if('s' in keys):
147         user2.centerY += 5
148         user2.rotateAngle = 0
149     if('d' in keys):
150         user2.centerX += 5
151         user2.rotateAngle = -90
152     # creates boundaries for user 1 and 2
153     if(user1.left < 10):
154         user1.left = 10
155     if(user1.right > 390):
156         user1.right = 390
157     if(user1.bottom > 390):
158         user1.bottom = 390
159     if(user2.left < 10):
160         user2.left = 10
161     if(user2.right > 390):
162         user2.right = 390
163     if(user2.top < 10):
164         user2.top = 10
165     # checks if a user hits the danger zone and if they do, ends game accordingly

```

```

166     if(user1.hitsShape(dangerZone) == True):
167         endGame('User 2', 'User 1 died to zone.')
168
169     if(user2.hitsShape(dangerZone) == True):
170         endGame('User 1', 'User 2 died to zone.')
171
172 def onStep():
173     # moves the laser once it is visible
174     if(laser1.visible == True):
175         shootLaser1()
176     if(laser2.visible == True):
177         shootLaser2()
178     # damages the health of a user if they are hit by a laser
179     if(laser1.hitsShape(user2) == True):
180         laser1.visible = False
181         laser1.centerY = user1.centerY
182         damageDealt('User 2')
183         if(user2Health.value <= 0):
184             endGame('User 1', ' User 2 died to laser.')
185     elif(laser2.hitsShape(user1) == True):
186         laser2.visible = False
187         laser2.centerY = user2.centerY
188         damageDealt('User 1')
189         if(user1Health.value <= 0):
190             endGame('User 2', ' User 1 died to laser.')
191     # calls crazyColors continuously so it checks app.isCrazyOn
192     if(app.topColors == True):
193         crazyColors(randrange(0, 401), randrange(0, 201),
194                     randrange(10, 26), randrange(3, 8))
195     elif(app.bottomColors == True):
196         crazyColors(randrange(0, 401), randrange(200, 401),
197                     randrange(10, 26), randrange(3, 8))
198
199
200 def drawMap():
201     # draws map, players, and danger zone
202     Rect(0, 0, 400, 400, fill=None,
203         border=gradient('green', 'red', 'green'), borderWidth = 10)
204     user1.visible = True
205     user2.visible = True
206     dangerZone.visible = True
207     for i in range(8):
208         Line(25 + 45 * i, 175, 60 + 45 * i, 225, opacity = 25)
209     user1Health.visible = True
210     user2Health.visible = True
211     Label('Health Points!', 100, 25, size=15)
212     Label('Health Points!', 340, 375, size=15)
213 # draws danger zone and hides it so it can be drawn with the map and called later
214 dangerZone = Group(
215     Rect(0, 175, 400, 50, fill='crimson', opacity=50, border='black')
216 )
217 # draws failure message and hides it so it can be drawn with the map and called later
218 failureMessage = Group(
219     Rect(200, 200, 300, 70, fill='lavender', align='center')
220 )
221 dangerZone.visible = False
222 failureMessage.visible = False

```

```
223 def endGame(player, reason):
224     # function used to end the game, which is called with the user who won,
225     # and the reason the other user lost.
226     failureMessage.visible = True
227     Label(player + ' wins!', 200, 200, align='center', size=25)
228     Label('Reason:' + reason, 200, 225, align='center', size=15)
229     app.stop()
230 def shootLaser1():
231     # moves the laser up by the .dy when called
232     laser1.centerY += laser1.dy
233 def shootLaser2():
234     # moves the other laser up by the .dy when called
235     laser2.centerY += laser2.dy
236 def damageDealt(player):
237     # chooses to remove health points from the user that was called upon
238     if(player == 'User 1'):
239         user1Health.value -= randrange(12, 26)
240     elif(player == 'User 2'):
241         user2Health.value -= randrange(12, 26)
```