

Name: XXXXXXXXXX
Module: No Module
Course: No Course



Tennis Simulator

default.js

```
1. var matchType = readline("[S]ingles or [D]oubles tennis game? ENTER ONLY S or D: ");
2. var gameStyle = readline("[S]erve or [R]eturn the tennis ball? ENTER ONLY S or R: ");
3.
4. var personalScore = readline("How many points did you win? ENTER ONLY 0, love, 15, 30, 40, deuce, ad-in, ad-out: ");
5. var opponentScore = readline("How many points did your opponent win? ENTER ONLY 0, 15, 30, 40, all, (Nothing for deuce & ads): ");
6.
7. var tennisScore = ["0-0", "0-all", "love-all", "15-0", "0-15", "15-15", "15-all", "30-0", "0-30", "15-30", "30-15", "30-30", "30-all",
8. "40-0", "0-40", "15-40", "40-15", "30-40", "40-30", "40-40", "40-all", "deuce", "ad-in", "ad-out"];
9. var deuceCourt = ["30-0", "0-30", "15-40", "40-15"];
10.
11. var combineScore = personalScore + "-" + opponentScore;
12. var acquiredScore = personalScore;
13.
14. var playerRadius = 20;
15. var playerColor = Color.red;
16.
17. function start()
18. {
19.     tennisCourt();
20.     sameScorePositions();
21.     differentScorePositions();
22.     tieBreakPositions();
23. }
24.
25. function tennisCourt()
26. {
27.     var doublesSideline = new Rectangle(getWidth(), getHeight());
28.     doublesSideline.setPosition(0,0);
29.     doublesSideline.setColor(Color.white);
30.     add(doublesSideline);
31.
32.     var lightBlue = new Color(30, 143, 213);
33.     var court = new Rectangle(getWidth() - 10, getHeight() - 10);
34.     court.setPosition(5, 5);
35.
36.     //rect.setColor("#1E8FD5");
37.     court.setColor(lightBlue);
38.     add(court);
39.
40.     var singlesSidelineOne = new Rectangle(5, 475);
41.     singlesSidelineOne.setPosition(40,0);
42.     singlesSidelineOne.setColor(Color.white);
43.     add(singlesSidelineOne);
44.
45.     var singlesSidelineTwo = new Rectangle(5, 475);
46.     singlesSidelineTwo.setPosition(355,0);
47.     singlesSidelineTwo.setColor(Color.white);
48.     add(singlesSidelineTwo);
49.
50.     var centerMarkOne = new Rectangle(5, 15);
51.     centerMarkOne.setPosition(getWidth() / 2, 0);
52.     centerMarkOne.setColor(Color.white);
53.     add(centerMarkOne);
54.
55.     var centerMarkTwo = new Rectangle(5, 15);
56.     centerMarkTwo.setPosition(getWidth() / 2, 465);
57.     centerMarkTwo.setColor(Color.white);
58.     add(centerMarkTwo);
59.
60.     var netPost = new Rectangle(475, 5);
61.     netPost.setPosition(0, getHeight() / 2);
62.     netPost.setColor(Color.white);
63.     add(netPost);
64.
65.     var serviceLine = new Rectangle(320, 5);
66.     serviceLine.setPosition(40, getHeight() / 4);
67.     serviceLine.setColor(Color.white);
68.     add(serviceLine);
69.
70.     var serviceLineTwo = new Rectangle(320, 5);
71.     serviceLineTwo.setPosition(40, getHeight() * 3/4);
72.     serviceLineTwo.setColor(Color.white);
```

```

73.     add(serviceLineTwo);
74.
75.     var centerServiceLine = new Rectangle(5, 245);
76.     centerServiceLine.setPosition(getWidth() / 2, getHeight() / 4);
77.     centerServiceLine.setColor(Color.white);
78.     add(centerServiceLine);
79. }
80.
81. function sameScorePositions()
82. {
83.     if(lookup(tennisScore, combineScore))
84.     {
85.         var scoreText = new Text(personalScore + "-" + opponentScore, "30pt Arial");
86.         scoreText.setPosition(getWidth()/2 - scoreText.getWidth()/2 + 2.5, getHeight()/2 + scoreText.getHeight()/2 - 7);
87.         scoreText.setColor(Color.black);
88.         add(scoreText);
89.
90.         if(matchType == "S" && gameStyle == "S" && personalScore == opponentScore)
91.         {
92.             //Personal Serving Position: (225, 460)
93.             renderPlayer(225, 460);
94.
95.             //Opponent Returning Position: (60, 20)
96.             renderPlayer(60, 20);
97.         }
98.
99.         if(matchType == "S" && gameStyle == "R" && personalScore == opponentScore)
100.        {
101.            //Opponent Serving Position: (180, 20)
102.            renderPlayer(180, 20);
103.
104.            //Personal Returning Position: (340, 460)
105.            renderPlayer(340, 460);
106.        }
107.
108.        if(matchType == "D" && gameStyle == "S" && personalScore == opponentScore)
109.        {
110.            //Personal Serving Position: (340, 460)
111.            renderPlayer(340, 460);
112.
113.            //Opponent Returning Position: (60, 20)
114.            renderPlayer(60, 20);
115.
116.            //Personal Net Position: (getWidth() / 3.4, getHeight() / 1.6)
117.            renderPlayer(getWidth() / 3.4, getHeight() / 1.6);
118.
119.            //Opponent Net Position: (getWidth() / 1.45, getHeight() / 2.6)
120.            renderPlayer(getWidth() / 1.45, getHeight() / 2.6);
121.        }
122.
123.        if(matchType == "D" && gameStyle == "R" && personalScore == opponentScore)
124.        {
125.            //Opponent Serving Position: (60, 20)
126.            renderPlayer(60, 20);
127.
128.            //Personal Returning Position: (340, 460)
129.            renderPlayer(340, 460);
130.
131.            //Opponent Net Position: (getWidth() / 1.45, getHeight() / 2.6)
132.            renderPlayer(getWidth() / 1.45, getHeight() / 2.6);
133.
134.            //Personal Net Position: (getWidth() / 3.4, getHeight() / 1.6)
135.            renderPlayer(getWidth() / 3.4, getHeight() / 1.6);
136.        }
137.    }
138. }
139.
140. function differentScorePositions()
141. {
142.     if(personalScore != opponentScore && opponentScore != "all" && personalScore != "ad")
143.     {
144.         if(lookup(deuceCourt, combineScore))
145.         {
146.             if(matchType == "S" && gameStyle == "S")
147.             {
148.                 //Personal Serving Position: (225, 460)
149.                 renderPlayer(225, 460);
150.
151.                 //Opponent Returning Position: (60, 20)
152.                 renderPlayer(60, 20);
153.             }
154.
155.             if(matchType == "S" && gameStyle == "R")
156.             {
157.                 //Opponent Serving Position: (180, 20)
158.                 renderPlayer(180, 20);
159.
160.                 //Personal Returning Position: (340, 460)
161.                 renderPlayer(340, 460);
162.             }
163.
164.             if(matchType == "D" && gameStyle == "S")

```

```

165.     {
166.         //Personal Serving Position: (340, 460)
167.         renderPlayer(340, 460);
168.
169.         //Opponent Returning Position: (60, 20)
170.         renderPlayer(60, 20);
171.
172.         //Personal Net Position: (getWidth() / 3.4, getHeight() / 1.6)
173.         renderPlayer(getWidth() / 3.4, getHeight() / 1.6);
174.
175.         //Opponent Net Position: (getWidth() / 1.45, getHeight() / 2.6)
176.         renderPlayer(getWidth() / 1.45, getHeight() / 2.6);
177.     }
178.
179.     if(matchType == "D" && gameStyle == "R")
180.     {
181.         //Opponent Serving Position: (60, 20)
182.         renderPlayer(60, 20);
183.
184.         //Personal Returning Position: (340, 460)
185.         renderPlayer(340, 460);
186.
187.         //Opponent Net Position: (getWidth() / 1.45, getHeight() / 2.6)
188.         renderPlayer(getWidth() / 1.45, getHeight() / 2.6);
189.
190.         //Personal Net Position: (getWidth() / 3.4, getHeight() / 1.6)
191.         renderPlayer(getWidth() / 3.4, getHeight() / 1.6);
192.     }
193. }
194. else
195. {
196.     if(lookup(tennisScore, combineScore))
197.     {
198.         if(matchType == "S" && gameStyle == "S")
199.         {
200.             //Personal Serving Position: (180, 460)
201.             renderPlayer(180, 460);
202.
203.             //Opponent Returning Position: (340, 20)
204.             renderPlayer(340, 20);
205.         }
206.         if(matchType == "S" && gameStyle == "R")
207.         {
208.             //Opponent Serving Position: (225, 20)
209.             renderPlayer(225, 20);
210.
211.             //Personal Returning Position: (60, 460)
212.             renderPlayer(60, 460);
213.         }
214.
215.         if(matchType == "D" && gameStyle == "S")
216.         {
217.             //Personal Serving Position: (60, 460)
218.             renderPlayer(60, 460);
219.
220.             //Opponent Returning Position: (340, 20)
221.             renderPlayer(340, 20);
222.
223.             //Personal Net Position: (getWidth() / 1.45, getHeight() / 1.6)
224.             renderPlayer(getWidth() / 1.45, getHeight() / 1.6);
225.
226.             //Opponent Net Position: (getWidth() / 3.4, getHeight() / 2.6)
227.             renderPlayer(getWidth() / 3.4, getHeight() / 2.6);
228.         }
229.
230.         if(matchType == "D" && gameStyle == "R")
231.         {
232.             //Opponent Serving Position: (60, 460)
233.             renderPlayer(60, 460);
234.
235.             //Personal Returning Position: (340, 20)
236.             renderPlayer(340, 20);
237.
238.             //Opponent Net Position: (getWidth() / 3.4, getHeight() / 2.6)
239.             renderPlayer(getWidth() / 3.4, getHeight() / 2.6);
240.
241.             //Personal Net Position: (getWidth() / 1.45, getHeight() / 1.6)
242.             renderPlayer(getWidth() / 1.45, getHeight() / 1.6);
243.         }
244.     }
245. }
246. }
247.
248. if(matchType == "S" && opponentScore == "all")
249. {
250.     if(gameStyle == "S")
251.     {
252.         //Personal Serving Position: (225, 460)
253.         renderPlayer(225, 460);
254.
255.         //Opponent Returning Position: (60, 20)
256.         renderPlayer(60, 20);

```

```

257.     }
258.     else
259.     {
260.         //Opponent Serving Position: (180, 20)
261.         renderPlayer(180, 20);
262.
263.         //Personal Returning Position: (340, 460)
264.         renderPlayer(340, 460);
265.     }
266. }
267.
268. if(matchType == "D" && opponentScore == "all")
269. {
270.     if(gameStyle == "S")
271.     {
272.         //Personal Serving Position: (340, 460)
273.         renderPlayer(340, 460);
274.
275.         //Opponent Returning Position: (60, 20)
276.         renderPlayer(60, 20);
277.
278.         //Personal Net Position: (getWidth() / 3.4, getHeight() / 1.6)
279.         renderPlayer(getWidth() / 3.4, getHeight() / 1.6);
280.
281.         //Opponent Net Position: (getWidth() / 1.45, getHeight() / 2.6)
282.         renderPlayer(getWidth() / 1.45, getHeight() / 2.6);
283.     }
284.     else
285.     {
286.         //Opponent Serving Position: (60, 20)
287.         renderPlayer(60, 20);
288.
289.         //Personal Returning Position: (340, 460)
290.         renderPlayer(340, 460);
291.
292.         //Opponent Net Position: (getWidth() / 1.45, getHeight() / 2.6)
293.         renderPlayer(getWidth() / 1.45, getHeight() / 2.6);
294.
295.         //Personal Net Position: (getWidth() / 3.4, getHeight() / 1.6)
296.         renderPlayer(getWidth() / 3.4, getHeight() / 1.6);
297.     }
298. }
299. }
300.
301. function tieBreakPositions()
302. {
303.     if(lookup(tennisScore, acquiredScore))
304.     {
305.         var scoreText = new Text(personalScore, "30pt Arial");
306.         scoreText.setPosition(getWidth()/2 - scoreText.getWidth()/2 + 2.5, getHeight()/2 + scoreText.getHeight()/2 - 7);
307.         scoreText.setColor(Color.black);
308.         add(scoreText);
309.
310.         if(matchType == "S" && personalScore == "deuce")
311.         {
312.             if(gameStyle == "S")
313.             {
314.                 //Personal Serving Position: (225, 460)
315.                 renderPlayer(225, 460);
316.
317.                 //Opponent Returning Position: (60, 20)
318.                 renderPlayer(60, 20);
319.             }
320.             else
321.             {
322.                 //Opponent Serving Position: (180, 20)
323.                 renderPlayer(180, 20);
324.
325.                 //Personal Returning Position: (340, 460)
326.                 renderPlayer(340, 460);
327.             }
328.         }
329.         if(matchType == "D" && personalScore == "deuce")
330.         {
331.             if(gameStyle == "S")
332.             {
333.                 //Personal Serving Position: (340, 460)
334.                 renderPlayer(340, 460);
335.
336.                 //Opponent Returning Position: (60, 20)
337.                 renderPlayer(60, 20);
338.
339.                 //Personal Net Position: (getWidth() / 3.4, getHeight() / 1.6)
340.                 renderPlayer(getWidth() / 3.4, getHeight() / 1.6);
341.
342.                 //Opponent Net Position: (getWidth() / 1.45, getHeight() / 2.6)
343.                 renderPlayer(getWidth() / 1.45, getHeight() / 2.6);
344.             }
345.             else
346.             {
347.                 //Opponent Serving Position: (60, 20)
348.                 renderPlayer(60, 20);

```

```

349.
350.         //Personal Returning Position: (340, 460)
351.         renderPlayer(340, 460);
352.
353.         //Opponent Net Position: (getWidth() / 1.45, getHeight() / 2.6)
354.         renderPlayer(getWidth() / 1.45, getHeight() / 2.6);
355.
356.         //Personal Net Position: (getWidth() / 3.4, getHeight() / 1.6)
357.         renderPlayer(getWidth() / 3.4, getHeight() / 1.6);
358.     }
359. }
360. }
361.
362. if(matchType == "S" && personalScore != "deuce")
363. {
364.     if(personalScore == "ad-in" || "ad-out" && opponentScore == "")
365.     {
366.         if(gameStyle == "S")
367.         {
368.             //Personal Serving Position: (180, 460)
369.             renderPlayer(180, 460);
370.
371.             //Opponent Returning Position: (340, 20)
372.             renderPlayer(340, 20);
373.         }
374.         else
375.         {
376.             //Opponent Serving Position: (225, 20)
377.             renderPlayer(225, 20);
378.
379.             //Personal Returning Position: (60, 460)
380.             renderPlayer(60, 460);
381.         }
382.     }
383. }
384.
385. if(matchType == "D" && personalScore != "deuce")
386. {
387.     if(personalScore == "ad-in" || "ad-out" && opponentScore == "")
388.     {
389.         //Personal Serving Position: (60, 460)
390.         renderPlayer(60, 460);
391.
392.         //Opponent Returning Position: (340, 20)
393.         renderPlayer(340, 20);
394.
395.         //Personal Net Position: (getWidth() / 1.45, getHeight() / 1.6)
396.         renderPlayer(getWidth() / 1.45, getHeight() / 1.6);
397.
398.         //Opponent Net Position: (getWidth() / 3.4, getHeight() / 2.6)
399.         renderPlayer(getWidth() / 3.4, getHeight() / 2.6);
400.     }
401. }
402. }
403.
404. function lookUp(arr, gameScore)
405. {
406.     for(var i = 0; i < arr.length; i++)
407.     {
408.         if(gameScore == arr[i])
409.         {
410.             {
411.                 return true;
412.             }
413.         }
414.     }
415.     return false;
416. }
417.
418. function renderPlayer(x1, y1)
419. {
420.     var player = new Circle(playerRadius);
421.     player.setPosition(x1, y1);
422.     player.setColor(playerColor);
423.     add(player);
424. }

```