

Paddle Game Questions

1. Provide a written response for your video that:
 - identifies the programming language;
 - identifies the purpose of your program; and
 - explains what the video illustrates.

Language is javascript, the point of the project was to learn how to program basic interactions between objects, although i already knew how to do that and didn't stick to the directions perfectly, the video shows what happens when balls collide with the top and bottom of the paddle, how the paddle is moved with the mouse, the score, and that more balls respawn when a ball does touch the bottom off the paddle.

2. Describe the **incremental** and **iterative** development process of your program, focusing on two distinct points in that process. Describe the difficulties and/or opportunities you encountered and how they were resolved or incorporated. In your description clearly indicate whether the development described was collaborative or independent. At least one of these points must refer to independent program development.

The incremental process included adding on a couple lines at a time onto the paddle and ball functions and constantly switching out variables for other variables when testing. I didn't look at the instructions at first, so when i incremented in a score counter, it was originally meant to be a debugging tool first and a feature later. This was independent

The iterative development included giving for loops and if statements the correct parameters so that they ran the way i wanted them to be run. I had a lot of trouble getting the if statement that checks if the ball is touching the paddle working at first, but then realized i was overcomplicating the code a lot and was able to fix the issue easy enough. This was collaborative, since i only learned that what i was doing was over complicated when i looked at another person's code.

3. Capture and paste the program code segment that implements an algorithm (marked with an oval) that is fundamental for your program to achieve its intended purpose. Your code segment must include an algorithm that integrates other algorithms and integrates mathematical and/or logical concepts. Describe how each algorithm within your selected algorithm functions independently, as well as in combination with others, to form a new algorithm that helps to achieve the intended purpose of the program. (*Approximately*

200 words)

```
81 //render and update paddle
82 ctx.beginPath();
83 ctx.rect(paddle.x-paddle.w, paddle.y-paddle.h, paddle.w*2, paddle.h*2);
84 ctx.fillStyle = paddle.c;
85 ctx.fill();
86 ctx.strokeStyle = paddle.c;
87 ctx.stroke();
88 paddle.x = mouseX;
89
90 ctx.font = "30px Comic Sans MS";
91 ctx.fillText("Score: " + count, 10, 50);
92
93 if (balls.length === 0){
94     ctx.fillText("Score: " + count + "
95 }
96 }
97
98 function Ball(){
99     //declares ball variables
100     this.x = Math.random()*(innerWidth-40)+20;
101     // this.y = Math.random()*innerHeight + 100;
102     // this.x = 1000;
103     this.y = 100;
104     this.dx = Math.random()*10-5;
105     this.dy = Math.random()*10-5;
106     this.rad = Math.random()*10+10;
107     this.c = 'rgba(' + Math.floor(Math.random()*225) + ',' + Math.floor(Math.random()*225) + ',' + Math.floor(Math.random()*225) + ',' + 1 + ')';
108 }
109
110 function Paddle(){
111     //declares paddle variables
112     this.x = innerWidth/2;
113     this.y = innerHeight/2 + innerHeight/4;
114     this.w = 125;
115     this.h = 37.5;
116     this.c = 'rgba(' + 255 + ',' + 255 + ',' + 255 + ',' + 1 + ')';
117 }
118
```

69:33

The “Paddle” and “Ball” functions are independent of all other algorithms, since they barrow no other variables from other algorithms and make all of their own. However, both are used in the rest off the program to update and render the game objects on the canvas, as seen in the very top algorithm. The very top algorithm uses other algorithms while it is running. For example, the top algorithm uses .fill() and .stroke(), which are algorithms provided by the javascript language itself. We, the users, don't need

to know what is inside fill or stroke, but what we do know is that we use them to give the game objects color and appearance. The top algorithm also uses ctx. quite a lot, meaning all of the function calls following it are other abstractions pulled from the ctx class, which we again, as coders, don't need to see to know what the abstractions do. Finally, the top algorithm pulls variables from the bottom two algorithms so that they can be rendered properly.

4. Capture and paste the program code segment that contains an abstraction you developed (marked with a rectangle in section 3 below). Your abstraction should integrate mathematical and logical concepts. Explain how your abstraction helped manage the complexity of your program.

```

7 requestAnimationFrame(animate);
8 ctx.clearRect(0, 0, canvas.width, canvas.height);
9 //render and update for balls and paddle
10 for (var i = 0; i < balls.length; i++){
11     //render and update a single ball
12     var ball = balls[i];
13     ctx.beginPath();
14     ctx.arc(ball.x, ball.y, ball.rad, 0, Math.PI*2);
15     ctx.fillStyle = ball.c;
16     ctx.fill();
17     ctx.strokeStyle = ball.c;
18     ctx.stroke();
19     //check edges
20     if(ball.x + ball.rad >= innerWidth-ball.dx || ball.x -ball.rad <= 0){
21         ball.dx = -(ball.dx);
22     }
23     if(ball.y + ball.rad >= innerHeight-ball.dy || ball.y - ball.rad <= 0){
24         ball.dy = - (ball.dy/1.05);
25     }
26     ball.x += ball.dx
27     ball.y += ball.dy;
28     ball.dy += 0.5;
29
30     //checks collisions between ball and paddle
31     if((ball.y - ball.rad) - (paddle.y - paddle.h) >= 0
32     && (ball.y + ball.rad) - (paddle.y + paddle.h) <= 0
33     && (ball.x - ball.rad) - (paddle.x - paddle.w) >= 0
34     && (ball.x + ball.rad) - (paddle.x + paddle.w) <= 0){
35         //if moving up
36         if (ball.dy < 0){
37             balls.splice(0, balls.length);
38             lengs = lengs + 5;
39             for(var i = 0; i < lengs; i++){
40                 balls.push(new Ball());
41             }
42             count = 0;
43         }
44         //if moving down
45         if (ball.dy > 0){
46             balls.splice(i, 1);
47             count++;
48         }
49     }
50 }
51 }

```

Using abstraction, i was able to create a large amount of balls and check all of those balls' positions relative to the paddle and walls to see if then needed to be removed or not with only a couple of lines rather than tens of lines. Also with this, i can increase the number of balls I want and how all of those balls will be effected inside the program by only changing one value, rather than many.

5. Capture and paste your entire program code in this section.

- Mark with an oval the segment of program code that implements the algorithm and integrates mathematical and /or logical concepts.
- Mark with a rectangle the segment of program code that represents an abstraction you developed.
- Include comments or citations for program code that has been written by someone else.

```
1 canvasDemo.js
2
3 window.onload = init; // wait for the page to load before
4
5 var canvas;
6 var ctx; // This is a better name for a global variable
7 var balls = new Array();
8 var paddle = new Paddle();
9 var count = 0;
10
11 function init(){
12     // get the canvas
13     canvas = document.getElementById('my');
14     // Set the dimensions of the canvas
15     canvas.width = window.innerWidth;
16     canvas.height = window.innerHeight;
17     canvas.style.border = 'solid black 2px';
18     canvas.style.backgroundColor = 'rgba(0, 0, 0, 1.0)';
19     // get the context
20     ctx = canvas.getContext('2d'); // This is the context
21     for(var i = 0; i < 25; i++){
22         balls.push(new Ball());
23     }
24     canvas.addEventListener('mousemove', function(){
25         mouseX = event.offsetX;
26     }, false);
27
28     lengs = balls.length;
29     animate(); // call to your animate function
30 }
31
32 // To do:
33 // 1. Declare and init variables x, y, dx, dy, radius;
34
35 function animate(){
36     //loops animate function
37     requestAnimationFrame(animate);
38     ctx.clearRect(0, 0, canvas.width, canvas.height);
39     //render and update for balls and paddle
40     for (var i = 0; i < balls.length; i++){
41         //render and update a single ball
42         var ball = balls[i];
43         ctx.beginPath();
44         ctx.arc(ball.x, ball.y, ball.rad, 0, Math.PI*2);
45         ctx.fillStyle = ball.c;
46         ctx.fill();
47
48         //check edges
49         if(ball.x + ball.rad >= innerWidth-ball.rad){ ball.x - ball.rad <= 0;
50             ball.dx = -(ball.dx);
51         }
52         if(ball.y + ball.rad >= innerHeight-ball.dy || ball.y - ball.rad <= 0){
53             ball.dy = -(ball.dy/1.05);
54         }
55         //moving up
56         ball.x += ball.dx;
57         ball.y += ball.dy;
58         ball.dy += 0.5;
59
60         //checks collisions between ball and paddle
61         if((ball.y - ball.rad) - (paddle.y - paddle.h) >= 0
62            && (ball.y + ball.rad) - (paddle.y + paddle.h) <= 0
63            && (ball.x + ball.rad) - (paddle.x - paddle.w) >= 0
64            && (ball.x - ball.rad) - (paddle.x + paddle.w) <= 0){
65             //moving up
66             if (ball.dy < 0){
67                 balls.splice(0, balls.length);
68                 lengs = lengs - 5;
69                 for(var i = 0; i < lengs; i++){
70                     balls.push(new Ball());
71                 }
72                 count = 0;
73             }
74             //if moving down
75             if (ball.dy > 0){
76                 balls.splice(i, 1);
77                 count++;
78             }
79         }
80
81         //render and update paddle
82         ctx.beginPath();
83         ctx.rect(paddle.x-paddle.w, paddle.y-paddle.h, paddle.w*2, paddle.h*2);
84         ctx.fillStyle = paddle.c;
85         ctx.fill();
86         ctx.strokeStyle = paddle.c;
87         ctx.stroke();
88         paddle.x = mouseX;
89
90         ctx.font = "30px Comic Sans MS";
91         ctx.fillText("Score: " + count, 10, 50);
92     }
93 }
94
95 //if moving down
96 if (ball.dy > 0){
97     balls.splice(i, 1);
98     count++;
99 }
100
101 //render and update paddle
102 ctx.beginPath();
103 ctx.rect(paddle.x-paddle.w, paddle.y-paddle.h, paddle.w*2, paddle.h*2);
104 ctx.fillStyle = paddle.c;
105 ctx.fill();
106 ctx.strokeStyle = paddle.c;
107 ctx.stroke();
108 paddle.x = mouseX;
109
110 ctx.font = "30px Comic Sans MS";
111 ctx.fillText("Score: " + count, 10, 50);
112 }
113
114 //if moving down
115 if (ball.dy > 0){
116     balls.splice(i, 1);
117     count++;
118 }
119
120 //render and update paddle
121 ctx.beginPath();
122 ctx.rect(paddle.x-paddle.w, paddle.y-paddle.h, paddle.w*2, paddle.h*2);
123 ctx.fillStyle = paddle.c;
124 ctx.fill();
125 ctx.strokeStyle = paddle.c;
126 ctx.stroke();
127 paddle.x = mouseX;
128
129 ctx.font = "30px Comic Sans MS";
130 ctx.fillText("Score: " + count, 10, 50);
131 }
132
133 //if moving down
134 if (ball.dy > 0){
135     balls.splice(i, 1);
136     count++;
137 }
138
139 //render and update paddle
140 ctx.beginPath();
141 ctx.rect(paddle.x-paddle.w, paddle.y-paddle.h, paddle.w*2, paddle.h*2);
142 ctx.fillStyle = paddle.c;
143 ctx.fill();
144 ctx.strokeStyle = paddle.c;
145 ctx.stroke();
146 paddle.x = mouseX;
147
148 ctx.font = "30px Comic Sans MS";
149 ctx.fillText("Score: " + count, 10, 50);
150 }
151
152 //if moving down
153 if (ball.dy > 0){
154     balls.splice(i, 1);
155     count++;
156 }
157
158 //render and update paddle
159 ctx.beginPath();
160 ctx.rect(paddle.x-paddle.w, paddle.y-paddle.h, paddle.w*2, paddle.h*2);
161 ctx.fillStyle = paddle.c;
162 ctx.fill();
163 ctx.strokeStyle = paddle.c;
164 ctx.stroke();
165 paddle.x = mouseX;
166
167 ctx.font = "30px Comic Sans MS";
168 ctx.fillText("Score: " + count, 10, 50);
169 }
170
171 //if moving down
172 if (ball.dy > 0){
173     balls.splice(i, 1);
174     count++;
175 }
176
177 //render and update paddle
178 ctx.beginPath();
179 ctx.rect(paddle.x-paddle.w, paddle.y-paddle.h, paddle.w*2, paddle.h*2);
180 ctx.fillStyle = paddle.c;
181 ctx.fill();
182 ctx.strokeStyle = paddle.c;
183 ctx.stroke();
184 paddle.x = mouseX;
185
186 ctx.font = "30px Comic Sans MS";
187 ctx.fillText("Score: " + count, 10, 50);
188 }
189
190 //if moving down
191 if (ball.dy > 0){
192     balls.splice(i, 1);
193     count++;
194 }
195
196 //render and update paddle
197 ctx.beginPath();
198 ctx.rect(paddle.x-paddle.w, paddle.y-paddle.h, paddle.w*2, paddle.h*2);
199 ctx.fillStyle = paddle.c;
200 ctx.fill();
201 ctx.strokeStyle = paddle.c;
202 ctx.stroke();
203 paddle.x = mouseX;
204
205 ctx.font = "30px Comic Sans MS";
206 ctx.fillText("Score: " + count, 10, 50);
207 }
208
209 //if moving down
210 if (ball.dy > 0){
211     balls.splice(i, 1);
212     count++;
213 }
214
215 //render and update paddle
216 ctx.beginPath();
217 ctx.rect(paddle.x-paddle.w, paddle.y-paddle.h, paddle.w*2, paddle.h*2);
218 ctx.fillStyle = paddle.c;
219 ctx.fill();
220 ctx.strokeStyle = paddle.c;
221 ctx.stroke();
222 paddle.x = mouseX;
223
224 ctx.font = "30px Comic Sans MS";
225 ctx.fillText("Score: " + count, 10, 50);
226 }
227
228 //if moving down
229 if (ball.dy > 0){
230     balls.splice(i, 1);
231     count++;
232 }
233
234 //render and update paddle
235 ctx.beginPath();
236 ctx.rect(paddle.x-paddle.w, paddle.y-paddle.h, paddle.w*2, paddle.h*2);
237 ctx.fillStyle = paddle.c;
238 ctx.fill();
239 ctx.strokeStyle = paddle.c;
240 ctx.stroke();
241 paddle.x = mouseX;
242
243 ctx.font = "30px Comic Sans MS";
244 ctx.fillText("Score: " + count, 10, 50);
245 }
246
247 //if moving down
248 if (ball.dy > 0){
249     balls.splice(i, 1);
250     count++;
251 }
252
253 //render and update paddle
254 ctx.beginPath();
255 ctx.rect(paddle.x-paddle.w, paddle.y-paddle.h, paddle.w*2, paddle.h*2);
256 ctx.fillStyle = paddle.c;
257 ctx.fill();
258 ctx.strokeStyle = paddle.c;
259 ctx.stroke();
260 paddle.x = mouseX;
261
262 ctx.font = "30px Comic Sans MS";
263 ctx.fillText("Score: " + count, 10, 50);
264 }
265
266 //if moving down
267 if (ball.dy > 0){
268     balls.splice(i, 1);
269     count++;
270 }
271
272 //render and update paddle
273 ctx.beginPath();
274 ctx.rect(paddle.x-paddle.w, paddle.y-paddle.h, paddle.w*2, paddle.h*2);
275 ctx.fillStyle = paddle.c;
276 ctx.fill();
277 ctx.strokeStyle = paddle.c;
278 ctx.stroke();
279 paddle.x = mouseX;
280
281 ctx.font = "30px Comic Sans MS";
282 ctx.fillText("Score: " + count, 10, 50);
283 }
284
285 //if moving down
286 if (ball.dy > 0){
287     balls.splice(i, 1);
288     count++;
289 }
290
291 //render and update paddle
292 ctx.beginPath();
293 ctx.rect(paddle.x-paddle.w, paddle.y-paddle.h, paddle.w*2, paddle.h*2);
294 ctx.fillStyle = paddle.c;
295 ctx.fill();
296 ctx.strokeStyle = paddle.c;
297 ctx.stroke();
298 paddle.x = mouseX;
299
300 ctx.font = "30px Comic Sans MS";
301 ctx.fillText("Score: " + count, 10, 50);
302 }
303
304 //if moving down
305 if (ball.dy > 0){
306     balls.splice(i, 1);
307     count++;
308 }
309
310 //render and update paddle
311 ctx.beginPath();
312 ctx.rect(paddle.x-paddle.w, paddle.y-paddle.h, paddle.w*2, paddle.h*2);
313 ctx.fillStyle = paddle.c;
314 ctx.fill();
315 ctx.strokeStyle = paddle.c;
316 ctx.stroke();
317 paddle.x = mouseX;
318
319 ctx.font = "30px Comic Sans MS";
320 ctx.fillText("Score: " + count, 10, 50);
321 }
322
323 //if moving down
324 if (ball.dy > 0){
325     balls.splice(i, 1);
326     count++;
327 }
328
329 //render and update paddle
330 ctx.beginPath();
331 ctx.rect(paddle.x-paddle.w, paddle.y-paddle.h, paddle.w*2, paddle.h*2);
332 ctx.fillStyle = paddle.c;
333 ctx.fill();
334 ctx.strokeStyle = paddle.c;
335 ctx.stroke();
336 paddle.x = mouseX;
337
338 ctx.font = "30px Comic Sans MS";
339 ctx.fillText("Score: " + count, 10, 50);
340 }
341
342 //if moving down
343 if (ball.dy > 0){
344     balls.splice(i, 1);
345     count++;
346 }
347
348 //render and update paddle
349 ctx.beginPath();
350 ctx.rect(paddle.x-paddle.w, paddle.y-paddle.h, paddle.w*2, paddle.h*2);
351 ctx.fillStyle = paddle.c;
352 ctx.fill();
353 ctx.strokeStyle = paddle.c;
354 ctx.stroke();
355 paddle.x = mouseX;
356
357 ctx.font = "30px Comic Sans MS";
358 ctx.fillText("Score: " + count, 10, 50);
359 }
360
361 //if moving down
362 if (ball.dy > 0){
363     balls.splice(i, 1);
364     count++;
365 }
366
367 //render and update paddle
368 ctx.beginPath();
369 ctx.rect(paddle.x-paddle.w, paddle.y-paddle.h, paddle.w*2, paddle.h*2);
370 ctx.fillStyle = paddle.c;
371 ctx.fill();
372 ctx.strokeStyle = paddle.c;
373 ctx.stroke();
374 paddle.x = mouseX;
375
376 ctx.font = "30px Comic Sans MS";
377 ctx.fillText("Score: " + count, 10, 50);
378 }
379
380 //if moving down
381 if (ball.dy > 0){
382     balls.splice(i, 1);
383     count++;
384 }
385
386 //render and update paddle
387 ctx.beginPath();
388 ctx.rect(paddle.x-paddle.w, paddle.y-paddle.h, paddle.w*2, paddle.h*2);
389 ctx.fillStyle = paddle.c;
390 ctx.fill();
391 ctx.strokeStyle = paddle.c;
392 ctx.stroke();
393 paddle.x = mouseX;
394
395 ctx.font = "30px Comic Sans MS";
396 ctx.fillText("Score: " + count, 10, 50);
397 }
398
399 //if moving down
400 if (ball.dy > 0){
401     balls.splice(i, 1);
402     count++;
403 }
404
405 //render and update paddle
406 ctx.beginPath();
407 ctx.rect(paddle.x-paddle.w, paddle.y-paddle.h, paddle.w*2, paddle.h*2);
408 ctx.fillStyle = paddle.c;
409 ctx.fill();
410 ctx.strokeStyle = paddle.c;
411 ctx.stroke();
412 paddle.x = mouseX;
413
414 ctx.font = "30px Comic Sans MS";
415 ctx.fillText("Score: " + count, 10, 50);
416 }
417
418 //if moving down
419 if (ball.dy > 0){
420     balls.splice(i, 1);
421     count++;
422 }
423
424 //render and update paddle
425 ctx.beginPath();
426 ctx.rect(paddle.x-paddle.w, paddle.y-paddle.h, paddle.w*2, paddle.h*2);
427 ctx.fillStyle = paddle.c;
428 ctx.fill();
429 ctx.strokeStyle = paddle.c;
430 ctx.stroke();
431 paddle.x = mouseX;
432
433 ctx.font = "30px Comic Sans MS";
434 ctx.fillText("Score: " + count, 10, 50);
435 }
436
437 //if moving down
438 if (ball.dy > 0){
439     balls.splice(i, 1);
440     count++;
441 }
442
443 //render and update paddle
444 ctx.beginPath();
445 ctx.rect(paddle.x-paddle.w, paddle.y-paddle.h, paddle.w*2, paddle.h*2);
446 ctx.fillStyle = paddle.c;
447 ctx.fill();
448 ctx.strokeStyle = paddle.c;
449 ctx.stroke();
450 paddle.x = mouseX;
451
452 ctx.font = "30px Comic Sans MS";
453 ctx.fillText("Score: " + count, 10, 50);
454 }
455
456 //if moving down
457 if (ball.dy > 0){
458     balls.splice(i, 1);
459     count++;
460 }
461
462 //render and update paddle
463 ctx.beginPath();
464 ctx.rect(paddle.x-paddle.w, paddle.y-paddle.h, paddle.w*2, paddle.h*2);
465 ctx.fillStyle = paddle.c;
466 ctx.fill();
467 ctx.strokeStyle = paddle.c;
468 ctx.stroke();
469 paddle.x = mouseX;
470
471 ctx.font = "30px Comic Sans MS";
472 ctx.fillText("Score: " + count, 10, 50);
473 }
474
475 //if moving down
476 if (ball.dy > 0){
477     balls.splice(i, 1);
478     count++;
479 }
480
481 //render and update paddle
482 ctx.beginPath();
483 ctx.rect(paddle.x-paddle.w, paddle.y-paddle.h, paddle.w*2, paddle.h*2);
484 ctx.fillStyle = paddle.c;
485 ctx.fill();
486 ctx.strokeStyle = paddle.c;
487 ctx.stroke();
488 paddle.x = mouseX;
489
490 ctx.font = "30px Comic Sans MS";
491 ctx.fillText("Score: " + count, 10, 50);
492 }
493
494 //if moving down
495 if (ball.dy > 0){
496     balls.splice(i, 1);
497     count++;
498 }
499
500 //render and update paddle
501 ctx.beginPath();
502 ctx.rect(paddle.x-paddle.w, paddle.y-paddle.h, paddle.w*2, paddle.h*2);
503 ctx.fillStyle = paddle.c;
504 ctx.fill();
505 ctx.strokeStyle = paddle.c;
506 ctx.stroke();
507 paddle.x = mouseX;
508
509 ctx.font = "30px Comic Sans MS";
510 ctx.fillText("Score: " + count, 10, 50);
511 }
512
513 //if moving down
514 if (ball.dy > 0){
515     balls.splice(i, 1);
516     count++;
517 }
518
519 //render and update paddle
520 ctx.beginPath();
521 ctx.rect(paddle.x-paddle.w, paddle.y-paddle.h, paddle.w*2, paddle.h*2);
522 ctx.fillStyle = paddle.c;
523 ctx.fill();
524 ctx.strokeStyle = paddle.c;
525 ctx.stroke();
526 paddle.x = mouseX;
527
528 ctx.font = "30px Comic Sans MS";
529 ctx.fillText("Score: " + count, 10, 50);
530 }
531
532 //if moving down
533 if (ball.dy > 0){
534     balls.splice(i, 1);
535     count++;
536 }
537
538 //render and update paddle
539 ctx.beginPath();
540 ctx.rect(paddle.x-paddle.w, paddle.y-paddle.h, paddle.w*2, paddle.h*2);
541 ctx.fillStyle = paddle.c;
542 ctx.fill();
543 ctx.strokeStyle = paddle.c;
544 ctx.stroke();
545 paddle.x = mouseX;
546
547 ctx.font = "30px Comic Sans MS";
548 ctx.fillText("Score: " + count, 10, 50);
549 }
550
551 //if moving down
552 if (ball.dy > 0){
553     balls.splice(i, 1);
554     count++;
555 }
556
557 //render and update paddle
558 ctx.beginPath();
559 ctx.rect(paddle.x-paddle.w, paddle.y-paddle.h, paddle.w*2, paddle.h*2);
560 ctx.fillStyle = paddle.c;
561 ctx.fill();
562 ctx.strokeStyle = paddle.c;
563 ctx.stroke();
564 paddle.x = mouseX;
565
566 ctx.font = "30px Comic Sans MS";
567 ctx.fillText("Score: " + count, 10, 50);
568 }
569
570 //if moving down
571 if (ball.dy > 0){
572     balls.splice(i, 1);
573     count++;
574 }
575
576 //render and update paddle
577 ctx.beginPath();
578 ctx.rect(paddle.x-paddle.w, paddle.y-paddle.h, paddle.w*2, paddle.h*2);
579 ctx.fillStyle = paddle.c;
580 ctx.fill();
581 ctx.strokeStyle = paddle.c;
582 ctx.stroke();
583 paddle.x = mouseX;
584
585 ctx.font = "30px Comic Sans MS";
586 ctx.fillText("Score: " + count, 10, 50);
587 }
588
589 //if moving down
590 if (ball.dy > 0){
591     balls.splice(i, 1);
592     count++;
593 }
594
595 //render and update paddle
596 ctx.beginPath();
597 ctx.rect(paddle.x-paddle.w, paddle.y-paddle.h, paddle.w*2, paddle.h*2);
598 ctx.fillStyle = paddle.c;
599 ctx.fill();
600 ctx.strokeStyle = paddle.c;
601 ctx.stroke();
602 paddle.x = mouseX;
603
604 ctx.font = "30px Comic Sans MS";
605 ctx.fillText("Score: " + count, 10, 50);
606 }
607
608 //if moving down
609 if (ball.dy > 0){
610     balls.splice(i, 1);
611     count++;
612 }
613
614 //render and update paddle
615 ctx.beginPath();
616 ctx.rect(paddle.x-paddle.w, paddle.y-paddle.h, paddle.w*2, paddle.h*2);
617 ctx.fillStyle = paddle.c;
618 ctx.fill();
619 ctx.strokeStyle = paddle.c;
620 ctx.stroke();
621 paddle.x = mouseX;
622
623 ctx.font = "30px Comic Sans MS";
624 ctx.fillText("Score: " + count, 10, 50);
625 }
626
627 //if moving down
628 if (ball.dy > 0){
629     balls.splice(i, 1);
630     count++;
631 }
632
633 //render and update paddle
634 ctx.beginPath();
635 ctx.rect(paddle.x-paddle.w, paddle.y-paddle.h, paddle.w*2, paddle.h*2);
636 ctx.fillStyle = paddle.c;
637 ctx.fill();
638 ctx.strokeStyle = paddle.c;
639 ctx.stroke();
640 paddle.x = mouseX;
641
642 ctx.font = "30px Comic Sans MS";
643 ctx.fillText("Score: " + count, 10, 50);
644 }
645
646 //if moving down
647 if (ball.dy > 0){
648     balls.splice(i, 1);
649     count++;
650 }
651
652 //render and update paddle
653 ctx.beginPath();
654 ctx.rect(paddle.x-paddle.w, paddle.y-paddle.h, paddle.w*2, paddle.h*2);
655 ctx.fillStyle = paddle.c;
656 ctx.fill();
657 ctx.strokeStyle = paddle.c;
658 ctx.stroke();
659 paddle.x = mouseX;
660
661 ctx.font = "30px Comic Sans MS";
662 ctx.fillText("Score: " + count, 10, 50);
663 }
664
665 //if moving down
666 if (ball.dy > 0){
667     balls.splice(i, 1);
668     count++;
669 }
670
671 //render and update paddle
672 ctx.beginPath();
673 ctx.rect(paddle.x-paddle.w, paddle.y-paddle.h, paddle.w*2, paddle.h*2);
674 ctx.fillStyle = paddle.c;
675 ctx.fill();
676 ctx.strokeStyle = paddle.c;
677 ctx.stroke();
678 paddle.x = mouseX;
679
680 ctx.font = "30px Comic Sans MS";
681 ctx.fillText("Score: " + count, 10, 50);
682 }
683
684 //if moving down
685 if (ball.dy > 0){
686     balls.splice(i, 1);
687     count++;
688 }
689
690 //render and update paddle
691 ctx.beginPath();
692 ctx.rect(paddle.x-paddle.w, paddle.y-paddle.h, paddle.w*2, paddle.h*2);
693 ctx.fillStyle = paddle.c;
694 ctx.fill();
695 ctx.strokeStyle = paddle.c;
696 ctx.stroke();
697 paddle.x = mouseX;
698
699 ctx.font = "30px Comic Sans MS";
700 ctx.fillText("Score: " + count, 10, 50);
701 }
702
703 //if moving down
704 if (ball.dy > 0){
705     balls.splice(i, 1);
706     count++;
707 }
708
709 //render and update paddle
710 ctx.beginPath();
711 ctx.rect(paddle.x-paddle.w, paddle.y-paddle.h, paddle.w*2, paddle.h*2);
712 ctx.fillStyle = paddle.c;
713 ctx.fill();
714 ctx.strokeStyle = paddle.c;
715 ctx.stroke();
716 paddle.x = mouseX;
717
718 ctx.font = "30px Comic Sans MS";
719 ctx.fillText("Score: " + count, 10, 50);
720 }
721
722 //if moving down
723 if (ball.dy > 0){
724     balls.splice(i, 1);
725     count++;
726 }
727
728 //render and update paddle
729 ctx.beginPath();
730 ctx.rect(paddle.x-paddle.w, paddle.y-paddle.h, paddle.w*2, paddle.h*2);
731 ctx.fillStyle = paddle.c;
732 ctx.fill();
733 ctx.strokeStyle = paddle.c;
734 ctx.stroke();
735 paddle.x = mouseX;
736
737 ctx.font = "30px Comic Sans MS";
738 ctx.fillText("Score: " + count, 10, 50);
739 }
740
741 //if moving down
742 if (ball.dy > 0){
743     balls.splice(i, 1);
744     count++;
745 }
746
747 //render and update paddle
748 ctx.beginPath();
749 ctx.rect(paddle.x-paddle.w, paddle.y-paddle.h, paddle.w*2, paddle.h*2);
750 ctx.fillStyle = paddle.c;
751 ctx.fill();
752 ctx.strokeStyle = paddle.c;
753 ctx.stroke();
754 paddle.x = mouseX;
755
756 ctx.font = "30px Comic Sans MS";
757 ctx.fillText("Score: " + count, 10, 50);
758 }
759
760 //if moving down
761 if (ball.dy > 0){
762     balls.splice(i, 1);
763     count++;
764 }
765
766 //render and update paddle
767 ctx.beginPath();
768 ctx.rect(paddle.x-paddle.w, paddle.y-paddle.h, paddle.w*2, paddle.h*2);
769 ctx.fillStyle = paddle.c;
770 ctx.fill();
771 ctx.strokeStyle = paddle.c;
772 ctx.stroke();
773 paddle.x = mouseX;
774
775 ctx.font = "30px Comic Sans MS";
776 ctx.fillText("Score: " + count, 10, 50);
777 }
778
779 //if moving down
780 if (ball.dy > 0){
781     balls.splice(i, 1);
782     count++;
783 }
784
785 //render and update paddle
786 ctx.beginPath();
787 ctx.rect(paddle.x-paddle.w, paddle.y-paddle.h, paddle.w*2, paddle.h*2);
788 ctx.fillStyle = paddle.c;
789 ctx.fill();
790 ctx.strokeStyle = paddle.c;
791 ctx.stroke();
792 paddle.x = mouseX;
793
794 ctx.font = "30px Comic Sans MS";
795 ctx.fillText("Score: " + count, 10, 50);
796 }
797
798 //if moving down
799 if (ball.dy > 0){
800     balls.splice(i, 1);
801     count++;
802 }
803
804 //render and update paddle
805 ctx.beginPath();
806 ctx.rect(paddle.x-paddle.w, paddle.y-paddle.h, paddle.w*2, paddle.h*2);
807 ctx.fillStyle = paddle.c;
808 ctx.fill();
809 ctx.strokeStyle = paddle.c;
810 ctx.stroke();
811 paddle.x = mouseX;
812
813 ctx.font = "30px Comic Sans MS";
814 ctx.fillText("Score: " + count, 10, 50);
815 }
816
817 //if moving down
818 if (ball.dy > 0){
819     balls.splice(i, 1);
820     count++;
821 }
822
823 //render and update paddle
824 ctx.beginPath();
825 ctx.rect(paddle.x-paddle.w, paddle.y-paddle.h, paddle.w*2, paddle.h*2);
826 ctx.fillStyle = paddle.c;
827 ctx.fill();
828 ctx.strokeStyle = paddle.c;
829 ctx.stroke();
830 paddle.x = mouseX;
831
832 ctx.font = "30px Comic Sans MS";
833 ctx.fillText("Score: " + count, 10, 50);
834 }
835
836 //if moving down
837 if (ball.dy > 0){
838     balls.splice(i, 1);
839     count++;
840 }
841
842 //render and update paddle
843 ctx.beginPath();
844 ctx.rect(paddle.x-paddle.w, paddle.y-paddle.h, paddle.w*2, paddle.h*2);
845 ctx.fillStyle = paddle.c;
846 ctx.fill();
847 ctx.strokeStyle = paddle.c;
848 ctx.stroke();
849 paddle.x = mouseX;
850
851 ctx.font = "30px Comic Sans MS";
852 ctx.fillText("Score: " + count, 10, 50);
853 }
854
855 //if moving down
856 if (ball.dy > 0){
857     balls.splice(i, 1);
858     count++;
859 }
860
861 //render and update paddle
862 ctx.beginPath();
863 ctx.rect(paddle.x-paddle.w, paddle.y-paddle.h, paddle.w*2, paddle.h*2);
864 ctx.fillStyle = paddle.c;
865 ctx.fill();
866 ctx.strokeStyle = paddle.c;
867 ctx.stroke();
868 paddle.x = mouseX;
869
870 ctx.font = "30px Comic Sans MS";
871 ctx.fillText("Score: " + count, 10, 50);
872 }
873
874 //if moving down
875 if (ball.dy > 0){
876     balls.splice(i, 1);
877     count++;
878 }
879
880 //render and update paddle
881 ctx.beginPath();
882 ctx.rect(paddle.x-paddle.w, paddle.y-paddle.h, paddle.w*2, paddle.h*2);
883 ctx.fillStyle = paddle.c;
884 ctx.fill();
885 ctx.strokeStyle = paddle.c;
886 ctx.stroke();
887 paddle.x = mouseX;
888
889 ctx.font = "30px Comic Sans MS";
890 ctx.fillText("Score: " + count, 10, 50);
891 }
892
893 //if moving down
894 if (ball.dy > 0){
895     balls.splice(i, 1);
896     count++;
897 }
898
899 //render and update paddle
900 ctx.beginPath();
901 ctx.rect(paddle.x-paddle.w, paddle.y-paddle.h, paddle.w*2, paddle.h*2);
902 ctx.fillStyle = paddle.c;
903 ctx.fill();
904 ctx.strokeStyle = paddle.c;
905 ctx.stroke();
906 paddle.x = mouseX;
907
908 ctx.font = "30px Comic Sans MS";
909 ctx.fillText("Score: " + count, 10, 50);
910 }
911
912 //if moving down
913 if (ball.dy > 0){
914     balls.splice(i, 1);
915     count++;
916 }
917
918 //render and update paddle
919 ctx.beginPath();
920 ctx.rect(paddle.x-paddle.w, paddle.y-paddle.h, paddle.w*2, paddle.h*2);
921 ctx.fillStyle = paddle.c;
922 ctx.fill();
923 ctx.strokeStyle = paddle.c;
924 ctx.stroke();
925 paddle.x = mouseX;
926
927 ctx.font = "30px Comic Sans MS";
928 ctx.fillText("Score: " + count, 10, 50);
929 }
930
931 //if moving down
932 if (ball.dy > 0){
933     balls.splice(i, 1);
934     count++;
935 }
936
937 //render and update paddle
938 ctx.beginPath();
939 ctx.rect(paddle.x-paddle.w, paddle.y-paddle.h, paddle.w*2, paddle.h*2);
940 ctx.fillStyle = paddle.c;
941 ctx.fill();
942 ctx.strokeStyle = paddle.c;
943 ctx.stroke();
944 paddle.x = mouseX;
945
946 ctx.font = "30px Comic Sans MS";
947 ctx.fillText("Score: " + count, 10, 50);
948 }
949
950 //if moving down
951 if (ball.dy > 0){
952     balls.splice(i, 1);
953     count++;
954 }
955
956 //render and update paddle
957 ctx.beginPath();
958 ctx.rect(paddle.x-paddle.w, paddle.y-paddle.h, paddle.w*2, paddle.h*2);
959 ctx.fillStyle = paddle.c;
960 ctx.fill();
961 ctx.strokeStyle = paddle.c;
962 ctx.stroke();
963 paddle.x = mouseX;
964
965 ctx.font = "30px Comic Sans MS";
966 ctx.fillText("Score: " + count, 10, 50);
967 }
968
969 //if moving down
970 if (ball.dy > 0){
971     balls.splice(i, 1);
972     count++;
973 }
974
975 //render and update paddle
976 ctx.beginPath();
977 ctx.rect(paddle.x-paddle.w, paddle.y-paddle.h, paddle.w*2, paddle.h*2);
978 ctx.fillStyle = paddle.c;
979 ctx.fill();
980 ctx.strokeStyle = paddle.c;
981 ctx.stroke();
982 paddle.x = mouseX;
983
984 ctx.font = "30px Comic Sans MS";
985 ctx.fillText("Score: " + count, 10, 50);
986 }
987
988 //if moving down
989 if (ball.dy > 0){
990     balls.splice(i, 1);
991     count++;
992 }
993
994 //render and update paddle
995 ctx.beginPath();
996 ctx.rect(paddle.x-paddle.w, paddle.y-paddle.h, paddle.w*2, paddle.h*2);
997 ctx.fillStyle = paddle.c;
998 ctx.fill();
999 ctx.strokeStyle = paddle.c;
1000 ctx.stroke();
1001 paddle.x = mouseX;
1002
1003 ctx.font = "30px Comic Sans MS";
1004 ctx.fillText("Score: " + count, 10, 50);
1005 }
1006
1007 //if moving down
1008 if (ball.dy > 0){
1009     balls.splice(i, 1);
1010     count++;
1011 }
1012
1013 //render and update paddle
1014 ctx.beginPath();
1015 ctx.rect(paddle.x-paddle.w, paddle.y-paddle.h, paddle.w*2, paddle.h*2);
1016 ctx.fillStyle = paddle.c;
1017 ctx.fill();
1018 ctx.strokeStyle = paddle.c;
1019 ctx.stroke();
1020 paddle.x = mouseX;
1021
1022 ctx.font = "30px Comic Sans MS";
1023 ctx.fillText("Score: " + count, 10, 50);
1024 }
1025
1026 //if moving down
1027 if (ball.dy > 0){
1028     balls.splice(i, 1);
1029     count++;
1030 }
1031
1032 //render and update paddle
1033 ctx.beginPath();
1034 ctx.rect(paddle.x-paddle.w, paddle.y-paddle.h, paddle.w*2, paddle.h*2);
1035 ctx.fillStyle = paddle.c;
1036 ctx.fill();
1037 ctx.strokeStyle = paddle.c;
1038 ctx.stroke();
1039 paddle.x = mouseX;
1040
1041 ctx.font = "30px Comic Sans MS";
1042 ctx.fillText("Score: " + count, 10, 50);
1043 }
1044
1045 //if moving down
1046 if (ball.dy > 0){
1047     balls.splice(i, 1);
1048     count++;
1049 }
1050
1051 //render and update paddle
1052 ctx.beginPath();
1053 ctx.rect(paddle.x-paddle.w, paddle.y-paddle.h, paddle.w*2, paddle.h*2);
1054 ctx.fillStyle = paddle.c;
1055 ctx.fill();
1056 ctx.strokeStyle = paddle.c;
1057 ctx.stroke();
1058 paddle.x = mouseX;
1059
1060 ctx.font = "30px Comic Sans MS";
1061 ctx.fillText("Score: " + count, 10, 50);
1062 }
1063
1064 //if moving down
1065 if (ball.dy > 0){
1066     balls.splice(i, 1);
1067     count++;
1068 }
1069
1070 //render and update paddle
1071 ctx.beginPath();
1072 ctx.rect(paddle.x-paddle.w, paddle.y-paddle.h, paddle.w*2, paddle.h*2);
1073 ctx.fillStyle = paddle.c;
1074 ctx.fill();
1075 ctx.strokeStyle = paddle.c;
1076 ctx.stroke();
1077 paddle.x = mouseX;
1078
1079 ctx.font = "30px Comic Sans MS";
1080 ctx.fillText("Score: " + count, 10, 50);
1081 }
1082
1083 //if moving down
1084 if (ball.dy > 0){
1085     balls.splice(i, 1);
1086     count++;
1087 }
1088
1089 //render and update paddle
1090 ctx.beginPath();
1091 ctx.rect(paddle.x-paddle.w, paddle.y-paddle.h, paddle.w*2, paddle.h*2);
1092 ctx.fillStyle = paddle.c;
1093 ctx.fill();
1094 ctx.strokeStyle = paddle.c;
1095 ctx.stroke();
1096 paddle.x = mouseX;
1097
1098 ctx.font = "30px Comic Sans MS";
1099 ctx.fillText("Score: " + count, 10, 50);
1100 }
1101
1102 //if moving down
1103 if (ball.dy > 0){
1104     balls.splice(i, 1);
1105     count++;
1106 }
1107
1108 //render and update paddle
1109 ctx.beginPath();
1110 ctx.rect(paddle.x-paddle.w, paddle.y-paddle.h, paddle.w*2, paddle.h*2);
1111 ctx.fillStyle = paddle.c;
1112 ctx.fill();
1113 ctx.strokeStyle = paddle.c;
1114 ctx.stroke();
1115 paddle.x = mouseX;
1116
1117 ctx.font = "30px Comic Sans MS";
1118 ctx.fillText("Score: " + count, 10, 50);
1119 }
1120
1121 //if moving down
1122 if (ball.dy > 0){
1123     balls.splice(i, 1);
1124     count++;
1125 }
1126
1127 //render and update paddle
1128 ctx.beginPath();
1129 ctx.rect(paddle.x-paddle.w, paddle.y-paddle.h, paddle.w*2, paddle.h*2);
1130 ctx.fillStyle = paddle.c;
1131 ctx.fill();
1132 ctx.strokeStyle = paddle.c;
1133 ctx.stroke();
1134 paddle.x = mouseX;
1135
1136 ctx.font = "30px Comic Sans MS";
1137 ctx.fillText("Score: " + count, 10, 50);
1138 }
1139
1140 //if moving down
1141 if (ball.dy > 0){
1142     balls.splice(i, 1);
1143     count++;
1144 }
1145
1146 //render and update paddle
1147 ctx.beginPath();
1148 ctx.rect(paddle.x-paddle.w, paddle.y-paddle.h, paddle.w*2, paddle.h*2);
1149 ctx.fillStyle = paddle.c;
1150 ctx.fill();
1151 ctx.strokeStyle = paddle.c;
1152 ctx.stroke();
1153 paddle.x = mouseX;
1154
1155 ctx.font = "30px Comic Sans MS";
1156 ctx.fillText("Score: " + count, 10, 50);
1157 }
1158
1159 //if moving down
1160 if (ball.dy > 0){
1161     balls.splice(i, 1);
1162     count++;
1163 }
1164
1165 //render and update paddle
1166 ctx.beginPath();
1167 ctx.rect(paddle.x-paddle.w, paddle.y-paddle.h, paddle.w*2, paddle.h*2);
1168 ctx.fillStyle = paddle.c;
1169 ctx.fill();
1170 ctx.strokeStyle = paddle.c;
1171 ctx.stroke();
1172 paddle.x = mouseX;
1173
1174 ctx.font = "30px Comic Sans MS";
1175 ctx.fillText("Score: " + count, 10, 50);
1176 }
1177
1178 //if moving down
1179 if (ball.dy > 0){
1180     balls.splice(i, 1);
1181     count++;
1182 }
1183
1184 //render and update paddle
1185 ctx.beginPath();
1186 ctx.rect(paddle.x-paddle.w, paddle.y-paddle.h, paddle.w*2, paddle.h*2);
1187 ctx.fillStyle = paddle.c;
1188 ctx.fill();
1189 ctx.strokeStyle = paddle.c;
1190 ctx.stroke();
1191 paddle.x = mouseX;
1192
1193 ctx.font = "30px Comic Sans MS";
1194 ctx.fillText("Score: " + count, 10, 50);
1195 }
1196
1197 //if moving down
1198 if (ball.dy > 0){
1199
```