

```

1
2  var gameChar_x;
3  var gameChar_y;
4  var gameChar_world_x;
5  var floorPos_y;
6  var isLeft;
7  var isRight;
8  var isFalling;
9  var isPlummeting;
10 var canyon;
11 var collectables;
12 var treePos_y;
13 var treePos_X;
14 var cameraPosX;
15 var clouds;
16 var collectableTimer;
17 var game_score;
18 var flagpole;
19 var lives;
20 var platforms;
21
22 /* Setup Function */
23 function setup() {
24   createCanvas(1500, 576);
25   lives = 3; // Initialize lives properly
26   startGame();
27 }
28
29 function startGame() {
30   floorPos_y = (height * 3) / 4;
31   gameChar_x = width / 2;
32   gameChar_y = floorPos_y;
33
34   cameraPosX = 0; // Initialize camera position
35
36   isFalling = false;
37   isLeft = false;
38   isRight = false;
39   isPlummeting = false;
40
41   canyon = { x_pos: 1200, y_pos: floorPos_y, size_x: 100, size_y: 150 };
42
43   collectables = [
44     {
45       x_pos: random(50, width - 50),
46       y_pos: floorPos_y - random(0, 50),
47       size: 40,
48       isFound: false,
49       reappearTime: 5000,
50     },
51     {
52       x_pos: random(50, width - 50),
53       y_pos: floorPos_y - random(0, 50),
54       size: 40,
55       isFound: false,
56       reappearTime: 5000,
57     },
58     {
59       x_pos: random(50, width - 50),
60       y_pos: floorPos_y - random(0, 50),
61       size: 40,
62       isFound: false,
63       reappearTime: 5000,
64     },
65   ];
66
67   collectableTimer = null;
68
69   treePos_X = [
70     random(80, 150),
71     random(250, 450),
72     random(600, 850),
73     random(1000, 1200),

```

```

74     random(1300, 1500),
75 ];
76 treePos_y = floorPos_y;
77
78 clouds = [
79   { x_pos: 200, y_pos: random(50, 200) },
80   { x_pos: 400, y_pos: random(50, 200) },
81   { x_pos: 800, y_pos: random(50, 300) },
82 ];
83
84 platforms = [];
85 platforms.push(createPlatforms(300, floorPos_y - 100, 200));
86 platforms.push(createPlatforms(1000, floorPos_y - 100, 200));
87
88 game_score = 0;
89
90 flagpole = { isReached: false, x_pos: 2000 };
91 }
92
93 function draw() {
94   background(100, 155, 255);
95   noStroke();
96   fill(0, 155, 0);
97   rect(0, floorPos_y, width, height - floorPos_y); // draw ground
98
99   fill(255);
100   textSize(40); // Score display
101   text(game_score, 50, 50);
102   text("Lives: " + lives, 50, height - 50);
103
104   push();
105   translate(-cameraPosX, 0);
106
107   drawSun();
108   drawClouds();
109   drawMountains();
110   drawTree();
111
112   // Create Platforms
113   for (var i = 0; i < platforms.length; i++) {
114     platforms[i].draw();
115   }
116   drawMultipleCollectables();
117   drawFlag();
118   checkFlagpole();
119   drawCanyon(canyon);
120
121   pop();
122
123   drawGameChar();
124
125   // Player Movement and Interaction
126   if (isRight) {
127     if (gameChar_x < width * 0.4) {
128       gameChar_x += 5; // Move the character to the right
129     } else {
130       cameraPosX += 5; // Scroll the world when reaching screen edge
131     }
132   }
133
134   if (isLeft) {
135     if (gameChar_x > width * 0.4) {
136       gameChar_x -= 5; // Move the character to the left
137     } else {
138       cameraPosX -= 5; // Scroll the world when reaching screen edge
139     }
140   }
141
142   // Apply Gravity
143   if (gameChar_y < floorPos_y) {
144     var isContact = false;
145     for (var i = 0; i < platforms.length; i++) {
146       if (platforms[i].checkContact(gameChar_world_x, gameChar_y)) {

```

```

147         isContact = true;
148         gameChar_y = platforms[i].y;
149         isFalling = false;
150         break;
151     }
152 }
153 if (!isContact) {
154     gameChar_y += 2; // Gravity
155     isFalling = true;
156 }
157 } else {
158     isFalling = false;
159 }
160
161 if (isPlummeting) {
162     gameChar_y += 10; // Plummeting effect
163 }
164
165 gameChar_world_x = gameChar_x + cameraPosX;
166
167 // Check if character falls off the screen
168 if (gameChar_y > height) {
169     if (lives > 0) {
170         lives -= 1;
171         startGame();
172     } else {
173         text("Game Over - Press Space to Restart", width / 2 - 150, height / 2);
174         noLoop();
175     }
176 }
177 }
178
179 // Key Pressed Function
180 function keyPressed() {
181     if (keyCode == 65 || keyCode == 37) {
182         // A or Left Arrow
183         isLeft = true;
184     }
185     if (keyCode == 68 || keyCode == 39) {
186         // D or Right Arrow
187         isRight = true;
188     }
189     if (
190         (keyCode == 87 || keyCode == 38 || key == " ") &&
191         gameChar_y == floorPos_y
192     ) {
193         gameChar_y -= 100;
194         isFalling = true;
195     }
196 }
197
198 // Key Released Function
199 function keyReleased() {
200     if (keyCode == 65 || keyCode == 37) {
201         isLeft = false;
202     }
203     if (keyCode == 68 || keyCode == 39) {
204         isRight = false;
205     }
206 }
207
208 // Drawing and character animations
209 function drawGameChar() {
210     noStroke();
211
212     if (isLeft && isFalling) {
213         //Jumping, turned left
214         var skinColor = color(245, 222, 179); //Wheat
215         var shirtColor = color(135, 206, 250); // LightSkyBlue
216         var trouserColor = color(0, 0, 128); // Navy
217         var shoesColor = color(169, 169, 169); // DarkGray
218         //head
219         fill(skinColor);

```

```

220     rect(gameChar_x - 15, gameChar_y - 80 + 2, 30, 28);
221     fill(128, 0, 0); //maroon
222     rect(gameChar_x - 15, gameChar_y - 80 + 2, 30, 7); // hair
223     //eyes whites
224     fill(255);
225     ellipse(gameChar_x - 7, gameChar_y - 65, 6, 4); // left
226     // ellipse(gameChar_x+7, gameChar_y-65, 6,4); // right
227     //eyes pupil
228     fill(0);
229     ellipse(gameChar_x - 7, gameChar_y - 65, 3, 3); // left
230     // ellipse(gameChar_x+7, gameChar_y-65, 3,3); // right
231     //Legs
232     fill(trouserColor); // Navy
233     rect(gameChar_x, gameChar_y - 28, 13, 40); // right side, left leg straight while
        jumping left
234     rect(gameChar_x - 20, gameChar_y - 20, 20, 13); // left side , right left bent
        while jumping left
235     //body
236     fill(shirtColor); // LightSkyBlue
237     rect(gameChar_x - 15, gameChar_y - 50, 30, 30);
238     //arms
239     fill(skinColor); // PeachPuff
240     // rect(gameChar_x-20, gameChar_y-42, 10, 25); // left
241
242     rect(gameChar_x - 25, gameChar_y - 42, 30, 10); // right
243     //feet
244     fill(shoesColor); // DarkGray
245     rect(gameChar_x, gameChar_y, 13, 13); // right side, right foot/shoe while jumping
246     rect(gameChar_x - 25, gameChar_y - 20, 8, 13); // left side, right foot/shoe bent
        while jumping
247 }
248
249 if (isRight && isFalling) {
250     // Jumping Turned Right
251     var skinColor = color(245, 222, 179); //Wheat
252     var shirtColor = color(135, 206, 250); // LightSkyBlue
253     var trouserColor = color(0, 0, 128); // Navy
254     var shoesColor = color(169, 169, 169); // DarkGray
255     //head
256     fill(skinColor);
257     rect(gameChar_x - 15, gameChar_y - 80 + 2, 30, 28);
258     fill(128, 0, 0); //maroon
259     rect(gameChar_x - 15, gameChar_y - 80 + 2, 30, 7); // hair
260     //eyes whites
261     fill(255);
262     // ellipse(gameChar_x-7, gameChar_y-65, 6,4); // left
263     ellipse(gameChar_x + 7, gameChar_y - 65, 6, 4); // right
264     //eyes pupil
265     fill(0);
266     // ellipse(gameChar_x-7, gameChar_y-65, 3,3); // left
267     ellipse(gameChar_x + 7, gameChar_y - 65, 3, 3); // right
268     //Legs
269     fill(trouserColor); // Navy
270     rect(gameChar_x - 13, gameChar_y - 28, 13, 40); // left side , right leg straight
        while jumping right
271     rect(gameChar_x, gameChar_y - 20, 22, 13); // right side , left leg up while
        jumping right
272     //body
273     fill(shirtColor); // LightSkyBlue
274     rect(gameChar_x - 15, gameChar_y - 50, 30, 30);
275     //arms
276     fill(skinColor); // PeachPuff
277     // rect(gameChar_x-20, gameChar_y-42, 10, 25); // left
278     // rect(gameChar_x-5, gameChar_y-42, 10, 25); // right
279     rect(gameChar_x - 5, gameChar_y - 42, 30, 10); // Right arm pointing right
280
281     //feet
282     fill(shoesColor); // DarkGray
283     rect(gameChar_x - 13, gameChar_y, 13, 12); // left side, right foot/shoe while
        jumping
284     rect(gameChar_x + 17, gameChar_y - 20, 8, 13); // right side, left left foot/shoe
        while jumping
285

```

```

286 //End Facing Right
287 }
288
289 if (isLeft) {
290 //Walking, turned left
291 var skinColor = color(245, 222, 179); //Wheat
292 var shirtColor = color(135, 206, 250); // LightSkyBlue
293 var trouserColor = color(0, 0, 128); // Navy
294 var shoesColor = color(169, 169, 169); // DarkGray
295 //head
296 fill(skinColor);
297 rect(gameChar_x - 15, gameChar_y - 80 + 2, 30, 28);
298 fill(128, 0, 0); //maroon
299 rect(gameChar_x - 15, gameChar_y - 80 + 2, 30, 7); // hair
300 //eyes whites
301 fill(255);
302 ellipse(gameChar_x - 7, gameChar_y - 65, 6, 4); // left
303 // ellipse(gameChar_x+7, gameChar_y-65, 6,4); // right
304 //eyes pupil
305 fill(0);
306 ellipse(gameChar_x - 7, gameChar_y - 65, 3, 3); // left
307 // ellipse(gameChar_x+7, gameChar_y-65, 3,3); // right
308 //Legs
309 fill(trouserColor); // Navy
310 rect(gameChar_x - 13, gameChar_y - 28, 26, 40);
311 //body
312 fill(shirtColor); // LightSkyBlue
313 rect(gameChar_x - 15, gameChar_y - 50, 30, 30);
314 //arms
315 fill(skinColor); // PeachPuff
316 // rect(gameChar_x-20, gameChar_y-42, 10, 25); // left side - right arm
317 rect(gameChar_x - 25, gameChar_y - 42, 30, 10); // right side - left arm pointing
left
318
319 // rect(gameChar_x-5, gameChar_y-42, 10, 25); // right side - left arm pointing
down
320 //feet
321 fill(shoesColor); // DarkGray
322 rect(gameChar_x - 13, gameChar_y, 26, 12);
323 }
324 if (isRight) {
325 // Walking Turned Right
326 var skinColor = color(245, 222, 179); //Wheat
327 var shirtColor = color(135, 206, 250); // LightSkyBlue
328 var trouserColor = color(0, 0, 128); // Navy
329 var shoesColor = color(169, 169, 169); // DarkGray
330
331 //head
332 fill(skinColor);
333 rect(gameChar_x - 15, gameChar_y - 80 + 2, 30, 28);
334 fill(128, 0, 0); //maroon
335 rect(gameChar_x - 15, gameChar_y - 80 + 2, 30, 7); // hair
336 //eyes whites
337 fill(255);
338 // ellipse(gameChar_x-7, gameChar_y-65, 6,4); // left
339 ellipse(gameChar_x + 7, gameChar_y - 65, 6, 4); // right
340 //eyes pupil
341 fill(0);
342 // ellipse(gameChar_x-7, gameChar_y-65, 3,3); // left
343 ellipse(gameChar_x + 7, gameChar_y - 65, 3, 3); // right
344
345 //Legs
346 fill(trouserColor); // Navy
347 rect(gameChar_x - 13, gameChar_y - 28, 26, 40);
348 //body
349 fill(shirtColor); // LightSkyBlue
350 rect(gameChar_x - 15, gameChar_y - 50, 30, 30);
351 //arms
352 fill(skinColor); // PeachPuff
353
354 rect(gameChar_x - 5, gameChar_y - 42, 30, 10); // Right arm pointing right
355
356 //feet

```

```

357     fill(shoesColor); // DarkGray
358     rect(gameChar_x - 13, gameChar_y, 26, 12);
359
360     //End Facing Right
361 }
362
363 if (isFalling || isPlummeting) {
364     // Character Facing Front - jumping
365     if (!isLeft && !isRight) {
366         //Standing, facing frontwards
367         var skinColor = color(245, 222, 179); //Wheat
368         var shirtColor = color(135, 206, 250); // LightSkyBlue
369         var trouserColor = color(0, 0, 128); // Navy
370         var shoesColor = color(169, 169, 169); // DarkGray
371         //head
372         fill(skinColor);
373         rect(gameChar_x - 15, gameChar_y - 80 + 2, 30, 28);
374         fill(128, 0, 0); //maroon
375         rect(gameChar_x - 15, gameChar_y - 80 + 2, 30, 7); // hair
376         //eyes whites
377         fill(255);
378         ellipse(gameChar_x - 7, gameChar_y - 65, 6, 4);
379         ellipse(gameChar_x + 7, gameChar_y - 65, 6, 4);
380         //eyes pupil
381         fill(0);
382         ellipse(gameChar_x - 7, gameChar_y - 65, 3, 3);
383         ellipse(gameChar_x + 7, gameChar_y - 65, 3, 3);
384         //arms
385         fill(skinColor); // PeachPuff
386         // rect(gameChar_x-20, gameChar_y-42, 10, 25); // left hands down
387         rect(gameChar_x - 30, gameChar_y - 50, 15, 13); // left arm out jumping
388         // rect(gameChar_x+10, gameChar_y-42, 10, 25); // right
389         rect(gameChar_x + 15, gameChar_y - 50, 15, 13); // Right arm pointing right
390         //body
391         fill(shirtColor); // LightSkyBlue
392         rect(gameChar_x - 15, gameChar_y - 50, 30, 30); // body while jumping / shirt
393         rect(gameChar_x + 15, gameChar_y - 50, 5, 13); // Left arm sleeve jumping
394         rect(gameChar_x - 20, gameChar_y - 50, 5, 13); // Right arm sleeve jumping
395
396         //Legs
397         fill(trouserColor); // Navy
398         rect(gameChar_x - 15, gameChar_y - 28, 30, 40);
399
400         //feet
401         fill(shoesColor); // DarkGray
402         rect(gameChar_x - 15, gameChar_y, 30, 12);
403     }
404 } else {
405     // Character Facing Front
406     if (!isLeft && !isRight) {
407         //Standing, facing frontwards
408         var skinColor = color(245, 222, 179); //Wheat
409         var shirtColor = color(135, 206, 250); // LightSkyBlue
410         var trouserColor = color(0, 0, 128); // Navy
411         var shoesColor = color(169, 169, 169); // DarkGray
412         //head
413         fill(skinColor);
414         rect(gameChar_x - 15, gameChar_y - 80 + 2, 30, 28);
415         fill(128, 0, 0); //maroon
416         rect(gameChar_x - 15, gameChar_y - 80 + 2, 30, 7); // hair
417         //eyes whites
418         fill(255);
419         ellipse(gameChar_x - 7, gameChar_y - 65, 6, 4);
420         ellipse(gameChar_x + 7, gameChar_y - 65, 6, 4);
421         //eyes pupil
422         fill(0);
423         ellipse(gameChar_x - 7, gameChar_y - 65, 3, 3);
424         ellipse(gameChar_x + 7, gameChar_y - 65, 3, 3);
425         //body
426         fill(shirtColor); // LightSkyBlue
427         rect(gameChar_x - 20, gameChar_y - 50, 40, 30);
428         //Legs
429         fill(trouserColor); // Navy

```

```

430     rect(gameChar_x - 15, gameChar_y - 28, 30, 40);
431     //arms
432     fill(skinColor); // PeachPuff
433     rect(gameChar_x - 20, gameChar_y - 42, 10, 25); // left hands down
434     // rect(gameChar_x-30, gameChar_y-50, 15, 13); // left arm out jumping
435     rect(gameChar_x + 10, gameChar_y - 42, 10, 25); // right
436     // rect(gameChar_x+15, gameChar_y-50, 15, 13); // Right arm pointing right
437
438     //feet
439     fill(shoesColor); // DarkGray
440     rect(gameChar_x - 15, gameChar_y, 30, 12);
441 }
442 }
443 }
444 function drawMountains() {
445     let mountainHeight = 15;
446     let baseWidth = 50;
447     let startX = width / 4;
448     let startY = floorPos_y;
449
450     for (var i = 0; i < 3; i++) {
451         let y = startY;
452         let x1 = startX + cameraPosX / 10 + (mountainHeight - i) * baseWidth;
453         let x2 =
454             startX + cameraPosX / 6 + ((mountainHeight + i - 10) * baseWidth) / 4;
455
456         fill(240);
457         triangle(x1, y, x2, y, (x1 + x2) / 2, y - 350);
458         fill(100);
459         triangle(x1 + 20, y, x2, y, (x1 + x2) / 2, y - 300);
460     }
461 }
462
463 function drawTree() {
464     for (var i = 0; i < treePos_X.length; i++) {
465         fill(120, 100, 19); // Brown color for trunk
466         rect(treePos_X[i] - 16, treePos_y - 100, 30, 100);
467
468         fill(34, 139, 34); // Green color for leaves
469         ellipse(treePos_X[i], treePos_y - 70, 150, 60);
470         ellipse(treePos_X[i], treePos_y - 110, 120, 70);
471         ellipse(treePos_X[i], treePos_y - 140, 80, 50);
472     }
473 }
474
475 function drawClouds() {
476     for (var i = 0; i < clouds.length; i++) {
477         fill(255);
478         ellipse(clouds[i].x_pos + cameraPosX / 20, clouds[i].y_pos, 55, 55);
479         ellipse(clouds[i].x_pos + 25 + cameraPosX / 20, clouds[i].y_pos, 55, 75);
480         ellipse(clouds[i].x_pos + 45 + cameraPosX / 20, clouds[i].y_pos, 55, 55);
481     }
482 }
483
484 function drawSun() {
485     fill("gold");
486     ellipse(150, floorPos_y - 300, 75, 75); // Draw the sun
487 }
488
489 function drawCanyon(t_canyon) {
490     fill(0);
491     rect(
492         t_canyon.x_pos,
493         t_canyon.y_pos,
494         t_canyon.size_x,
495         t_canyon.size_y
496     );
497     checkPlummeting();
498 }
499
500 function checkPlummeting() {
501     // Adjust the plummeting logic to ensure it triggers as soon as the character
    reaches the canyon's edge

```

```

502     if (
503         gameChar_world_x > canyon.x_pos &&
504         gameChar_world_x < canyon.x_pos + canyon.size_x &&
505         gameChar_y >= floorPos_y - 1 // Ensures the character is on the ground
506     ) {
507         isPlummeting = true;
508     }
509
510     if (isPlummeting) {
511         gameChar_y += 10; // Character falls faster when plummeting
512     }
513 }
514
515 function drawCollectable(t_collectable) {
516     if (!t_collectable.isFound) {
517         noFill();
518         strokeWeight(6);
519         stroke(220, 185, 0);
520         ellipse(
521             t_collectable.x_pos,
522             t_collectable.y_pos - 20,
523             t_collectable.size,
524             t_collectable.size
525         );
526         fill(255, 0, 255);
527         stroke(255);
528         strokeWeight(1);
529         quad(
530             t_collectable.x_pos - 5,
531             t_collectable.y_pos - t_collectable.size,
532             t_collectable.x_pos - 10,
533             t_collectable.y_pos - (t_collectable.size + 15),
534             t_collectable.x_pos + 10,
535             t_collectable.y_pos - (t_collectable.size + 15),
536             t_collectable.x_pos + 5,
537             t_collectable.y_pos - t_collectable.size
538         );
539     }
540 }
541
542 function checkCollectable(t_collectable) {
543     if (
544         !t_collectable.isFound &&
545         dist(
546             gameChar_world_x,
547             gameChar_y,
548             t_collectable.x_pos,
549             t_collectable.y_pos
550         ) < 50
551     ) {
552         t_collectable.isFound = true;
553         game_score += 1;
554         t_collectable.collectableTimer = millis();
555     }
556
557     if (
558         t_collectable.isFound &&
559         millis() - t_collectable.collectableTimer >= t_collectable.reappearTime
560     ) {
561         t_collectable.isFound = false; // Reappear collectable
562         t_collectable.collectableTimer = null; // Reset timer
563     }
564 }
565
566 function drawMultipleCollectables() {
567     for (var i = 0; i < collectables.length; i++) {
568         drawCollectable(collectables[i]);
569         checkCollectable(collectables[i]);
570     }
571 }
572
573 function drawFlag() {
574     stroke(255, 255, 0);

```



```

575     strokeWeight(3);
576     line(
577         flagpole.x_pos,
578         floorPos_y - 150,
579         flagpole.x_pos,
580         floorPos_y
581     );
582     if (flagpole.isReached) {
583         noStroke();
584         fill(255, 0, 0);
585         rect(flagpole.x_pos, floorPos_y - 150, 40, 30);
586     } else {
587         noStroke();
588         fill(255, 0, 0);
589         rect(flagpole.x_pos, floorPos_y - 30, 40, 30);
590     }
591 }
592
593 function checkFlagpole() {
594     if (gameChar_world_x + 30 >= flagpole.x_pos) {
595         flagpole.isReached = true;
596     }
597 }
598
599 function createPlatforms(x, y, length) {
600     var p = {
601         x: x,
602         y: y,
603         length: length,
604         draw: function () {
605             // Adjust the platform's drawing position by adding scrollPos
606             fill(255, 0, 255);
607             rect(this.x, this.y, this.length, 20);
608         },
609         checkContact: function (gc_x, gc_y) {
610             // Use gameChar_world_x to check if the character is aligned with the platform
611             if (gc_x > this.x && gc_x < this.x + this.length) {
612                 var d = this.y - gc_y;
613                 if (d >= 0 && d < 5) {
614                     return true;
615                 }
616             }
617             return false;
618         },
619     };
620     return p;
621 }
622

```