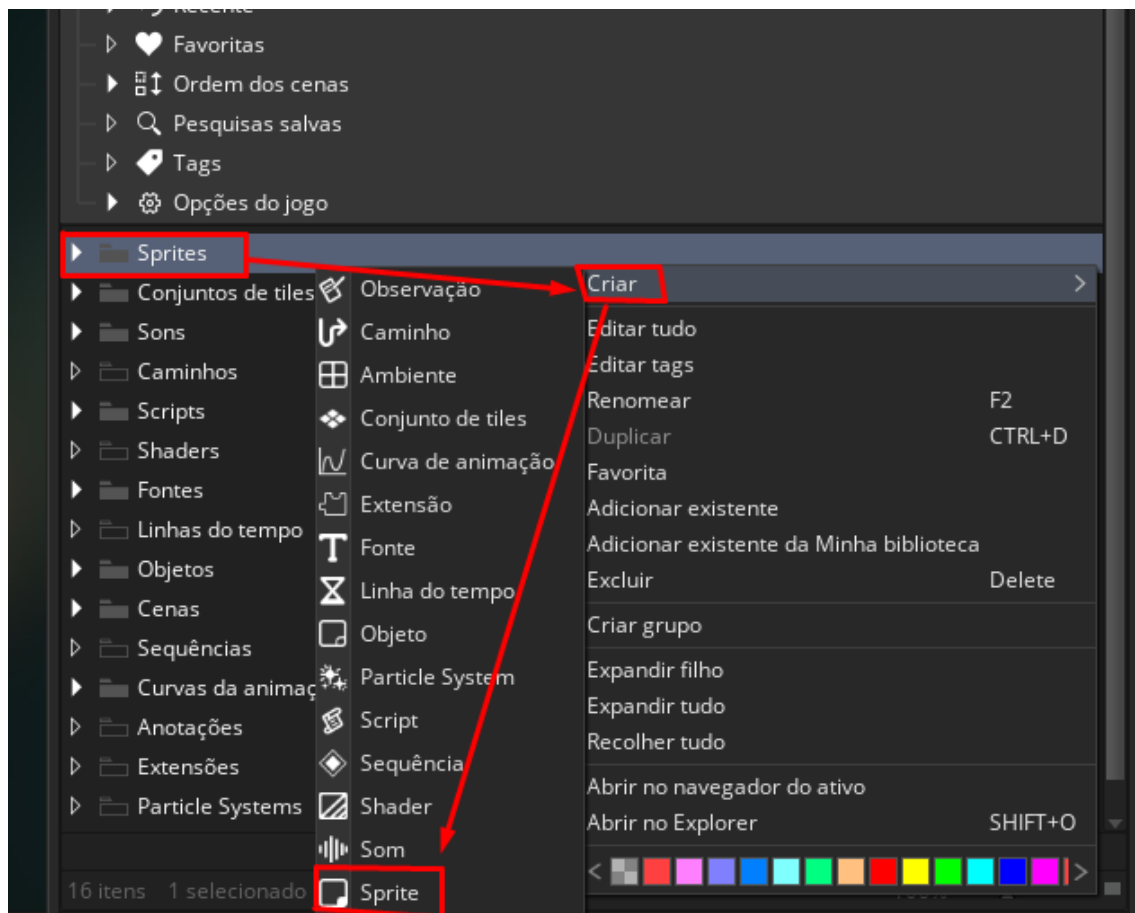


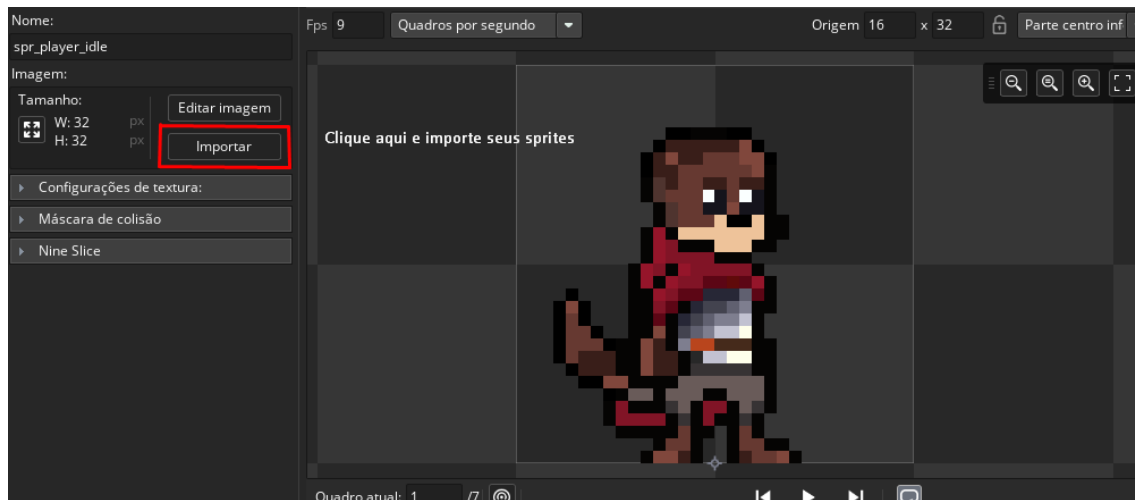
Manual de desenvolvimento do Otter Knight

1. Importe seus áudios, sprites e tilesets:

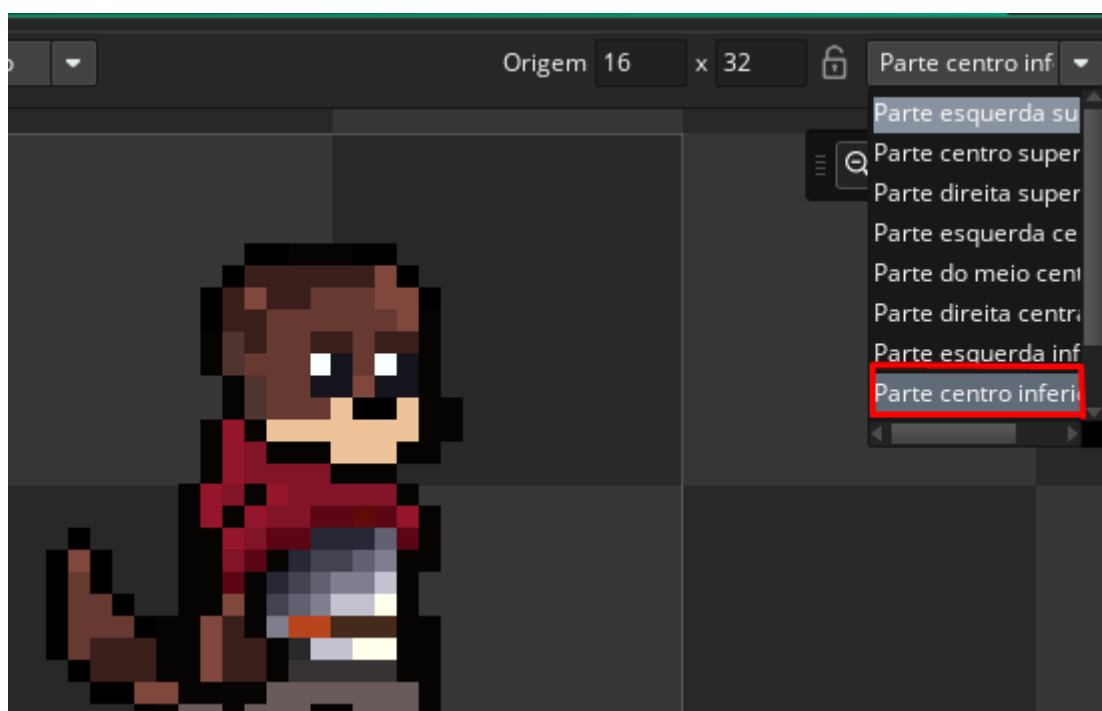
- abra a aba de recursos, clique na pasta “sprites” com o botão direito do mouse



Importe as sprites de todos os personagens (player, inimigos), dos itens e tilesets

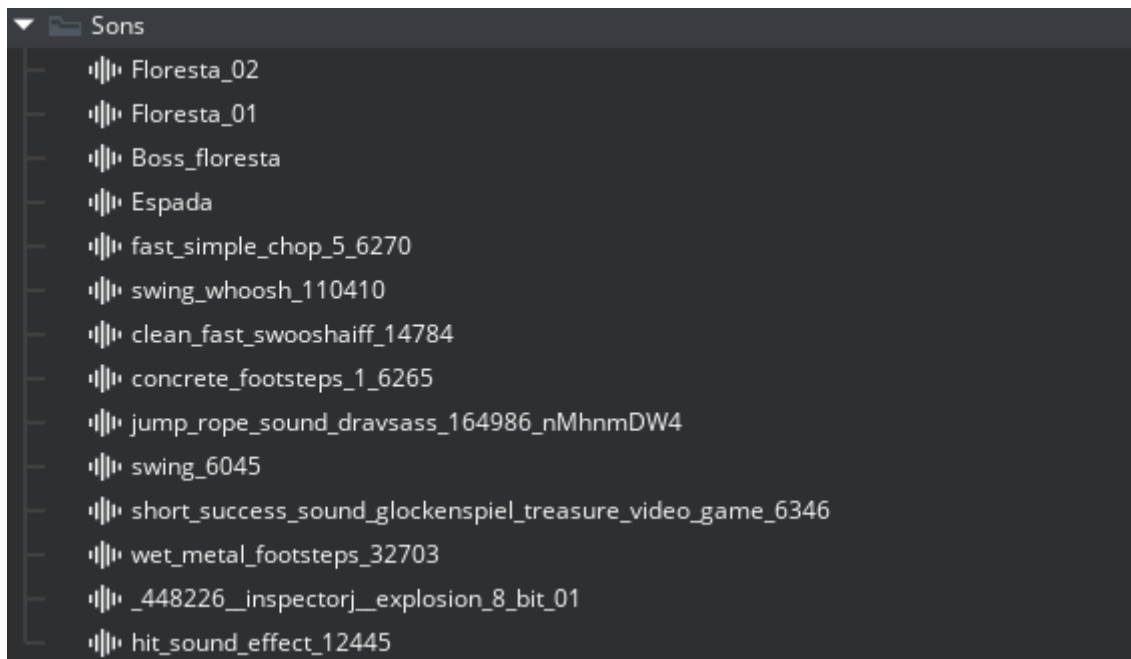


Deixe todos os sprites das entidades com origem na parte do centro inferior

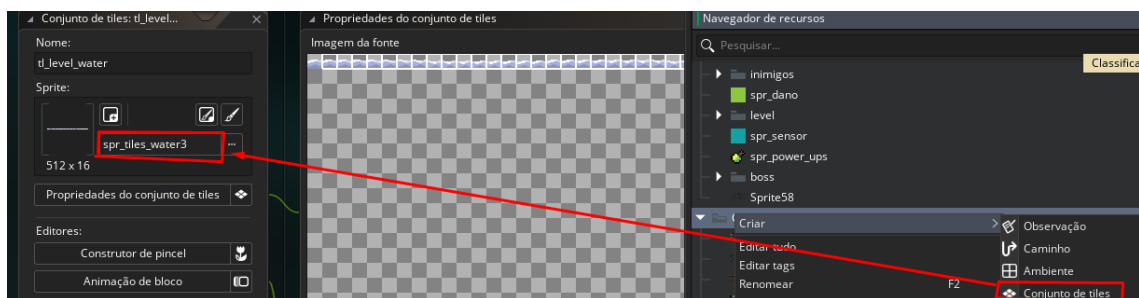


Faça isso com todas as entidades (cada entidade possui várias sprites para movimentos diferentes).

Faça a mesma coisa com os tilesets e os sons em suas respectivas pastas (tilesets ficam na pasta dos sprites)

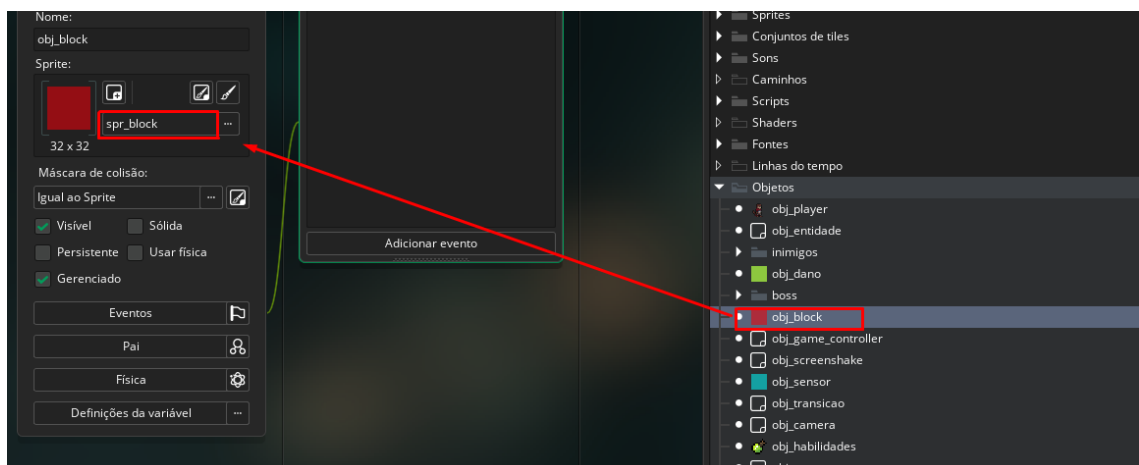


Crie os conjuntos de tiles



2. Crie os objetos

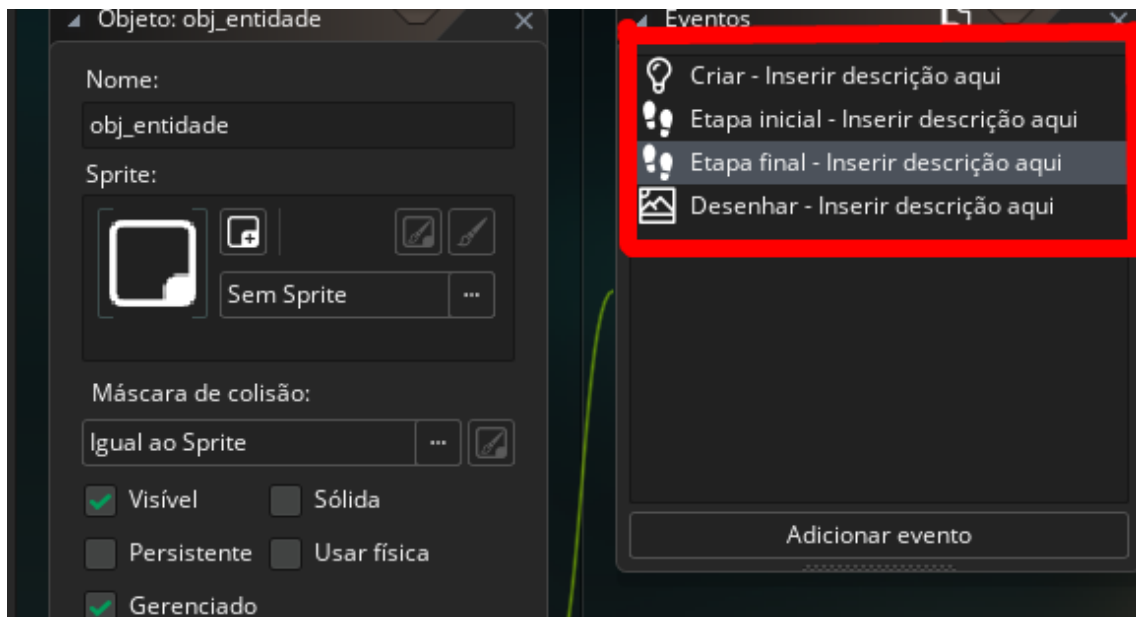
-Crie um sprite simples com um quadrado para ser o chão e as paredes e crie um objeto utilizando esse sprite



Crie uma room e distribua o objeto na room



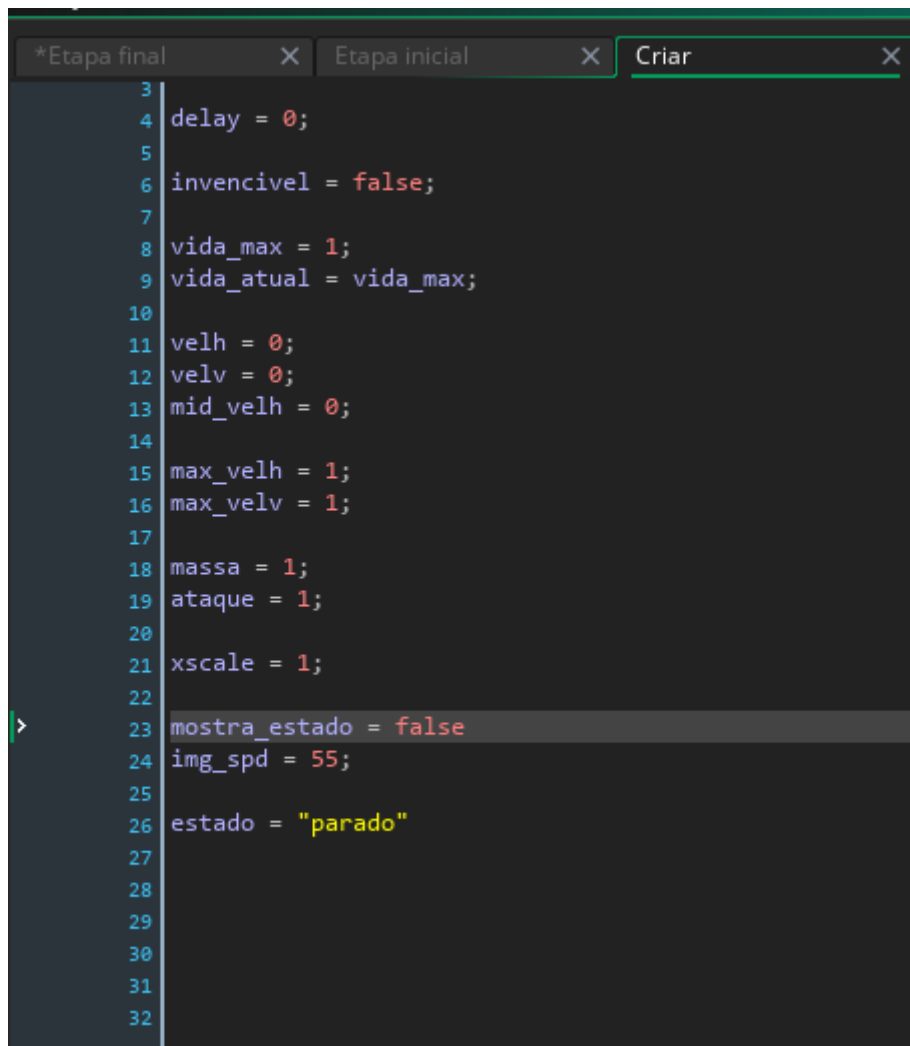
3.1 Crie um objeto para todas as entidades e adicione esses eventos



digite esses códigos:

```
*Etapa final X Etapa inicial X Criar X Desenhar X
4 if (abs(velh) <= max_velh)
5 {
6     velh += mid_velh;
7 }
8 else
9 {
10     velh = 0;
11 }
12 //colisão e movimentos
13 var _velh = sign(velh);
14 var _velv = sign(velv);
15
16 repeat(abs(velh))
17 {
18     if (place_meeting(x + _velh, y, obj_block))
19     {
20         velh = 0;
21         break;
22     }
23     x += _velh;
24 }
25 repeat(abs(velv))
26 {
27     if (place_meeting(x, y + _velv, obj_block))
28     {
29         velv = 0;
30         break;
31     }
32     y += _velv;
33 }
```

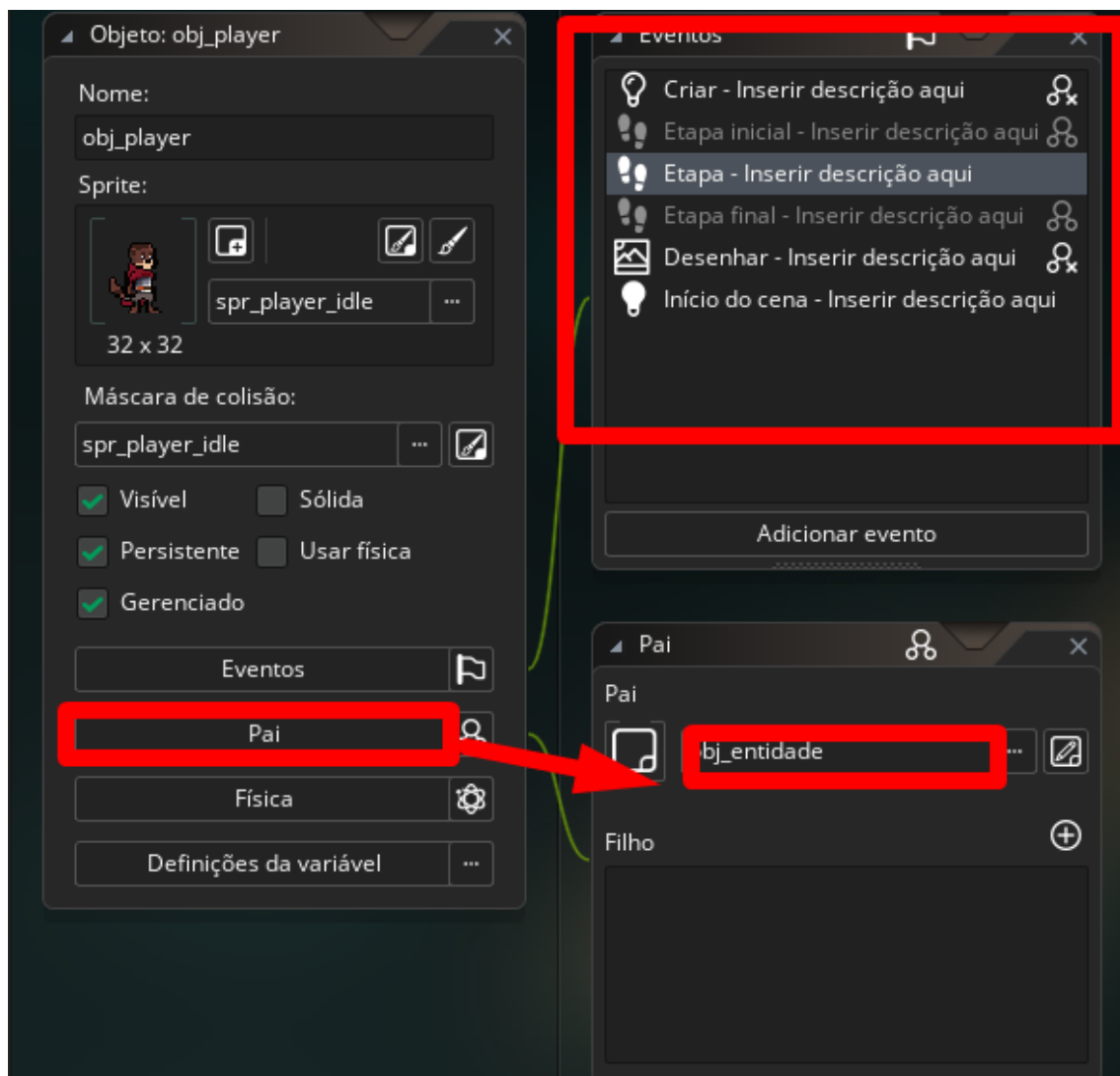
```
*Etapa final X Etapa inicial X Criar X Dese
1 /// @description Inserir descrição aqui
2 // Você pode escrever seu código neste editor
3 if (velh != 0) xscale = sign(velh);
4 image_xscale = xscale;
5
6 if (position_meeting(mouse_x, mouse_y, id))
7 {
8     if (mouse_check_button_released(mb_left))
9     {
10         mostra_estado = !mostra_estado;
11     }
12
13     image_speed = (img_spd/room_speed)
14
15
16
17
18
```



The image shows a code editor window with three tabs: '*Etapa final', 'Etapa inicial', and 'Criar'. The 'Criar' tab is active and contains the following Python code:

```
3
4 delay = 0;
5
6 invencivel = false;
7
8 vida_max = 1;
9 vida_atual = vida_max;
10
11 velh = 0;
12 velv = 0;
13 mid_velh = 0;
14
15 max_velh = 1;
16 max_velv = 1;
17
18 massa = 1;
19 ataque = 1;
20
21 xscale = 1;
22
23 mostra_estado = false
24 img_spd = 55;
25
26 estado = "parado"
27
28
29
30
31
32
```

-Crie outro objeto, para o player



Digite os códigos:

```
Desenhar X *Criar X Etapa X Início do cena
1  /// @description Inserir descrição aqui
2  // Você pode escrever seu código neste editor
3  randomize();
4
5  var cam = instance_create_layer(x, y, layer, obj_camera);
6  cam.alvo = id;
7  // Inherit the parent event
8  event_inherited();
9
10
11  vida_max = 10;
12  vida_atual = vida_max;
13
14  max_velh = 3;
15  max_velv = 6;
16  dash_vel = 7;
17
18  mostra_estado = true
19
20  combo = 0;
21  dano = noone;
22  possoatk = true;
23  ataque_multi = 1;
24  ataque_buff = room_speed;
25
26  invencivel = false;
27  invencivel_timer = room_speed * 2;
28  tempo_invencivel = invencivel_timer;
29
30  dash_delay = room_speed * 2;
31  dash_timer = 0;
32  dash_aereo_timer = 0
33  dash_aereo = true;
34
```

```
35  //controle de habilidades (wall jump e dash)
36  global.power_ups = [false, false]
37
38  aplica_gravidade = function()
39  {
40      var chao = place_meeting(x, y + 1, obj_block);
41      //gravidade
42      if (!chao)
43      {
44          if (velv < max_velv * 2)
45          {
46              velv += GRAVIDADE * massa * global.vel_mult;
47          }
48      }
49  }
```



```

1  /// @description Inserir descrição aqui
2  // Você pode escrever seu código neste editor
3
4  //checando se o objeto transição existe
5  if (instance_exists(obj_transicao)) exit;
6
7
8  //controlando a invencibilidade
9  if (invencivel && tempo_invencivel > 0)
10 {
11     tempo_invencivel--;
12     image_alpha = max(sin(get_timer()/100000), 0.2);
13 }
14 else
15 {
16     invencivel = false;
17     image_alpha = 1;
18 }
19
20 //movimentação
21 var right, left, jump, attack, dash;
22 var chao = place_meeting(x, y + 1, obj_block);
23
24
25
26
27 right = keyboard_check(ord("D")) || (gamepad_axis_value(0, gp_axislh) > 0);
28 left = keyboard_check(ord("A")) || (gamepad_axis_value(0, gp_axislh) < 0);
29 jump = keyboard_check_pressed(ord("W")) || (gamepad_button_check_pressed(0, gp_face1));
30 attack = keyboard_check_pressed(ord("L")) || (gamepad_button_check_pressed(0, gp_face3));
31 dash = keyboard_check_pressed(ord("J")) || (gamepad_button_check_pressed(0, gp_shoulderrb));
32

```

```

33 if (ataque_buff > 0) ataque_buff -= 20;
34
35 if (dash_timer > 0) dash_timer--;
36
37
38 velh = (right - left) * max_velh * global.vel_mult;
39
40
41
42
43 //estados
44
45 switch(estados)
46 {
47     #region parado
48     case "parado":
49     {
50
51         if (chao)
52         {
53             dash_aereo = true;
54         }
55
56         mid_velh = 0;
57
58         sprite_index = spr_player_idle;
59
60         //condição troca de estado
61         //movendo
62
63         if (velh != 0)
64         {
65             estado = "movendo"
66         }
67     }
68 }

```

```
Desenhar x *Criar x Etapa x Início do cena x
67 else if (jump || !chao)
68 {
69     estado = "pulando";
70     velv = (-max_velv * jump);
71     image_index = 0;
72 }
73 else if (attack)
74 {
75     estado = "ataque";
76     velh = 0;
77     image_index = 0;
78 }
79 else if (dash && dash_timer <= 0)
80 {
81     estado = "dash";
82     image_index = 0;
83 }
84 if (right && left && jump)
85 {
86     velv = 0;
87     sprite_index = spr_player_idle;
88     estado = "parado";
89 }
90 break;
91 }
92 #endregion
93
94 #region movendo
95 case "movendo":
96 {
97     //comportamento estado de
98     sprite_index = spr_player_run;
99
100
```

```
Desenhar x *Criar x Etapa x Início do cena x
101 //condição troca de estado
102 //parado
103 if(abs(velh) < .1)
104 {
105     estado = "parado";
106     velh = 0;
107 }
108 else if (jump || !chao)
109 {
110     estado = "pulando";
111     velv = (-max_velv * jump);
112     image_index = 0;
113 }
114 else if (attack)
115 {
116     estado = "ataque";
117     velh = 0;
118     image_index = 0;
119 }
120 else if (dash && dash_timer <= 0)
121 {
122     estado = "dash";
123     image_index = 0;
124 }
125 break;
126 }
127 #endregion
128
129 #region pulando
130 case "pulando":
131 {
132     if (velv > 0)
133     {
134
```

```
Desenhar x *Criar x Etapa x Início do cena x
134 {
135     sprite_index = spr_player_fall;
136     if (image_index >= image_number-1)
137     {
138         image_index = image_number - 1;
139     }
140 }
141 else
142 {
143     sprite_index = spr_player_jump;
144     if (image_index >= image_number-1)
145     {
146         image_index = image_number - 1;
147     }
148 }
149
150 //condição troca de estado
151 if (attack)
152 {
153     estado = "ataque aereo";
154 }
155 if (chao)
156 {
157     estado = "parado"
158     velh = 0;
159 }
160 //trocando de sprite se toquei na parede
161 //trocando de sprite se toquei na parede
162 var wall = place_meeting(x + sign(velh), y, obj_block);
163 //se estou na parede e tenho essa habilidade
164
165 if (wall && global.power_ups[0])
166 {
167     //fazendo eu pular ao apertar a tecla jump
```

```
Desenhar x *Criar x Etapa x Início do cena x
167 //fazendo eu pular ao apertar a tecla jump
168 if (jump)
169 {
170     //indo pra cima
171     velv = -max_velv;
172     //indo pra direção certa
173     mid_velh = (max_velh * 2) * sign(velh) * -1;
174 }
175
176
177 sprite_index = spr_player_wall;
178 //só altera o velv se estiver caindo
179 if (velv > 1)
180 {
181     velv = 1;
182 }
183 else
184 {
185     aplica_gravidade();
186 }
187 }
188 else
189 {
190     aplica_gravidade();
191
192     //diminuindo o valor do mid_velh
193     mid_velh = lerp(mid_velh, 0, 0.4);
194 }
195
196 //dash aereo
197 if (dash && dash_aereo == true && global.power_ups[1])
198 {
199     estado = "dash aereo"
200 }
201
```

```
Desenhar X *Criar X Etapa X Início do cena X
202 break;
203 }
204 #endregion
205
206 #region dash aereo
207
208 case "dash aereo":
209 {
210     dash_aereo = false;
211
212     if (sprite_index != spr_player_dash)
213     {
214         audio_play_sound(swing_whoosh_110410, 5, false);
215         sprite_index = spr_player_dash;
216         dash_aereo_timer = room_speed / 4;
217     }
218
219     dash_aereo_timer--;
220
221     if (dash_aereo_timer <= 0)
222     {
223         estado = "parado"
224     }
225
226     velh = 0;
227     mid_velh = dash_vel * image_xscale;
228     velv = 0;
229
230     break;
231 }
232 #endregion
233
234 #region ataque
```

```
Espaço de tra... X Room2 X Room1 X Room3 X spr_boss_jump X rm_menu X rm_init X Script3 X obj_menu: Ev...
Desenhar X *Criar X Etapa X Início do cena X
236 case "ataque":
237 {
238     velh = 0;
239
240     if (combo == 0)
241     {
242         sprite_index = spr_atk_;
243     }
244     else if (combo == 1)
245     {
246         sprite_index = spr_player_atk_2;
247     }
248     else if (combo == 2)
249     {
250         sprite_index = spr_player_atk_3;
251         mid_velh = image_xscale * dash_vel;
252         dano = instance_create_layer(x + sprite_width/2, y - sprite_height/2, layer, obj_dano);
253         dano.dano = ataque;
254         dano.pai = id;
255     }
256
257     if (image_index >= 2 && dano == noone && possoatk)
258     {
259         audio_play_sound(Espada, 10, false)
260         dano = instance_create_layer(x + sprite_width/2, y - sprite_height/2, layer, obj_dano);
261         dano.dano = ataque * ataque_multi;
262         dano.pai = id;
263         possoatk = false;
264     }
265
266     //buff
267     if (attack && combo < 2)
268     {
269
```

```
Desenhar x *Criar x Etapa x Inicio do cena x
269 {
270     ataque_buff = room_speed;
271 }
272
273
274 if (ataque_buff && combo < 2 && image_index >= image_number -1)
275 {
276     combo ++;
277     image_index = 0;
278     possoatk = true;
279     ataque_multi += .5;
280     if (dano)
281     {
282         instance_destroy(dano, false);
283         dano = noone;
284     }
285     //zerar buff
286     ataque_buff = 0;
287 }
288
289 if (image_index > image_number-1)
290 {
291     estado = "parado";
292     velh = 0;
293     combo = 0;
294     possoatk = true;
295     ataque_multi = 1;
296     if (dano)
297     {
298         instance_destroy(dano, false);
299         dano = noone;
300     }
301     if (dash && dash_timer <= 0)
302     {
303         estado = "dash"
```

```
Desenhar x *Criar x Etapa x Inicio do cena x
304     image_index = 0;
305     combo = 0;
306     if (dano)
307     {
308         instance_destroy(dano, false);
309         dano = noone;
310     }
311 }
312 }
313 break
314 }
315 #endregion
316 #region dash
317 case "dash":
318 {
319     if (sprite_index != spr_player_dash)
320     {
321         audio_play_sound(swing_whoosh_110410, 7, false, 5);
322         sprite_index = spr_player_dash;
323         image_index = 0;
324     }
325     mid_velh = image_xscale * dash_vel;
326     velh = 0;
327     if (image_index >= image_number - 1 || !chao )
328     {
329         estado = "parado"
330         mid_velh = 0;
331         dash_timer = dash_delay;
332     }
333 }
```

```

Espaço de tra... X Room2 X Room1 X Room3 X spr_boss_jump X rm_menu X rm_init X
Desenhar X *Criar X Etapa X Início do cena X
338     break;
339 }
340 #endregion
341
342 case "hit":
343 {
344     if (sprite_index != spr_player_hit)
345     {
346         sprite_index = spr_player_hit;
347         image_index = 0;
348         audio_play_sound(hit_sound effect 12445, 8, false, 0.5)
349         screenshake(5);
350
351         //invencibilidade
352         invencivel = true;
353         tempo_invencivel = invencivel_timer;
354     }
355     // Ficando parado ao levar dano
356     velh = 0;
357     velv = 0;
358     // Verificando se a animação de ser atingido terminou
359     if (image_index >= image_number - 1)
360     {
361         // Verificando se a vida está menor que zero
362         if (vida_atual < 1)
363         {
364             estado = "dead";
365         }
366         else
367         {
368             estado = "parado";
369         }
370     }
371     break;

```

```

Desenhar X *Criar X Etapa X Início do cena X
373
374 case "dead":
375 {
376     if (instance_exists(obj_game_controller))
377     {
378         with(obj_game_controller)
379         {
380             game_over = true;
381         }
382     }
383
384     velh = 0;
385     if (sprite_index != spr_player_death)
386     {
387         image_index = 0;
388         sprite_index = spr_player_death;
389     }
390
391     if (image_index >= image_number -1)
392     {
393         image_index = image_number -1
394     }
395
396     break;
397 }
398
399 #region ataqueaereo
400 case "ataque aereo":
401 {
402     aplica_gravidade();
403     //checando se troquei de sprite
404     if (sprite_index != spr_player_airatk)
405     {
406         sprite_index = spr_player_airatk;
407         image_index = 0;

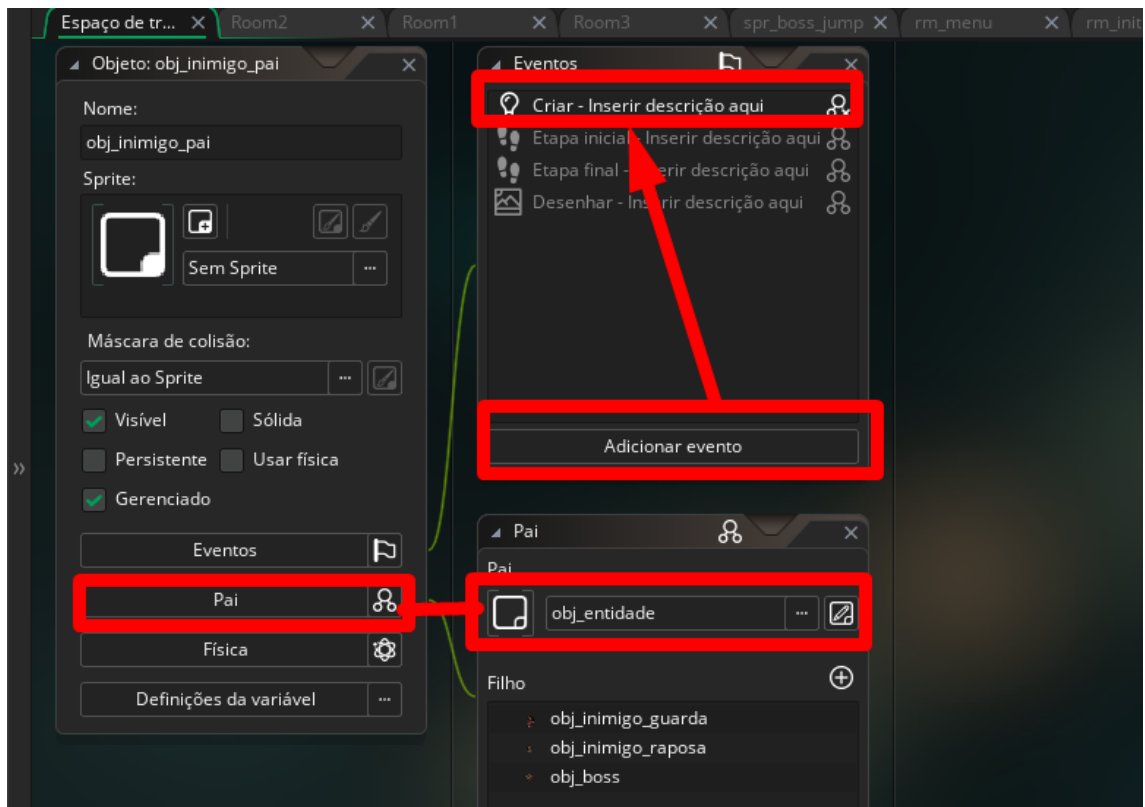
```

```
Desenhar  X  *Criar  X  Etapa  X  Início do cena  X
408 }
409 if (image_index >= 1 && dano == noone && possoatk)
410 {
411     audio_play_sound(Espada, 10, false)
412     dano = instance_create_layer(x + sprite_width/2 + velh * 2, y - sprite_height/2, layer, obj_dano);
413     dano.dano = ataque;
414     dano.pai = id;
415     possoatk = false;
416 }
417
418 //saíndo do estado
419 if (image_index >= image_number - 1)
420 {
421     estado = "pulando"
422     if (image_index > image_number-1)
423     {
424         estado = "parado";
425         possoatk = true;
426         if (dano)
427         {
428             instance_destroy(dano, false);
429             dano = noone;
430         }
431     }
432 }
433 if (chao)
434 {
435     estado = "parado"
436     possoatk = true;
437     if (dano)
438     {
439         instance_destroy(dano, false);
440         dano = noone;
441     }
```

```
443 }
444
445 break;
446
447 #endregion
448 //estado padrão
449 default:
450 {
451     estado = "parado"
452 }
453 }
454
455 if (keyboard_check_released(vk_enter)) || (gamepad_button_check_pressed(0, gp_face4)) game_restart();
456
```

```
Desenhar  X  *Criar  X  Etapa  X  Início do cena  X
1  /// @description Inserir descrição aqui
2  // Você pode escrever seu código neste editor
3
4  var cam = instance_create_layer(x, y, layer, obj_camera);
5  cam.alvo = id;
6
7
```

4 Crie um objeto para impedir que os inimigos causem danos neles mesmos



```

1  // @description Inserir descrição aqui
2  // Você pode escrever seu código neste editor
3
4
5  // Inherit the parent event
6  event_inherited();
7
8
9  //criando o método de ataque
10 //@method atacando()
11 //@args sprite_index image_index_min image_index_max dist_x dist_y, [xscale_dano, yscale_dano], [proximo_estado]
12 atacando = function(sprite_index, _image_index_min, _image_index_max, _dist_x, _dist_y, _xscale_dano, _yscale_dano, _proximo_estado, _som)
13 {
14     // caso a pessoa não passe xscale, eu defino ele como 1
15
16     if (!_xscale_dano) _xscale_dano = 1;
17     if (!_yscale_dano) _yscale_dano = 1;
18     if (_proximo_estado == undefined) _proximo_estado = "parado"
19     _som = asset_sound;
20
21
22     mid_velh = 0;
23     velh = 0;
24     if(sprite_index != _sprite_index)
25     {
26         image_index = 0;
27         possoatk = true;
28         dano = noone;
29     }
30     _sprite_index = _sprite_index;
31
32     if(image_index > image_number - 1)
33     {
34         estado = _proximo_estado;

```

```

35     }
36
37     //criando o dano
38     if (image_index >= _image_index_min && dano == noone && image_index < _image_index_max && possoatk)
39     {
40         dano = instance_create_layer(x + _dist_x, y - _dist_y, layer, obj_dano);
41         dano.dano = ataque;
42         dano.pai = id;
43         dano.image_xscale = _xscale_dano;
44         dano.image_yscale = _yscale_dano;
45         possoatk = false
46     }
47
48     //destruindo o dano
49     if (dano != noone && image_index >= _image_index_max)
50     {
51         instance_destroy(dano);
52         dano = noone;
53     }
54 }

```


- 5 Crie dois objetos para os inimigos, que irão ter os mesmos códigos, porém com alguns valores diferentes (velocidade, massa e dano) e suas sprites próprias

```
Etapa x Desenhar x Criar x
1  /// @description Inserir descrição aqui
2  // Você pode escrever seu código neste editor
3  var chao = place_meeting(x, y + 1, obj_block);
4
5
6  if (!chao)
7  {
8      velv += GRAVIDADE * massa * global.vel_mult;
9  }
10 //if (mouse_check_button_pressed(mb_right))
11 //{
12 //  estado = "ataque";
13 //}
14
15
16 switch(estado)
17 {
18     case "parado":
19     {
20         mid_velh = 0;
21         temporizador_estado++;
22         if (sprite_index != spr_inimigo_raposa_idle)
23         {
24             image_index = 0;
25         }
26         sprite_index = spr_inimigo_raposa_idle;
27
28         if (irandom(temporizador_estado) > 300)
29         {
30             estado = choose("andando", "parado", "andando");
31             temporizador_estado = 0;
32         }
33     }
34     ScrAtkPlayerMelee(obj_player, dist, xscale);
```

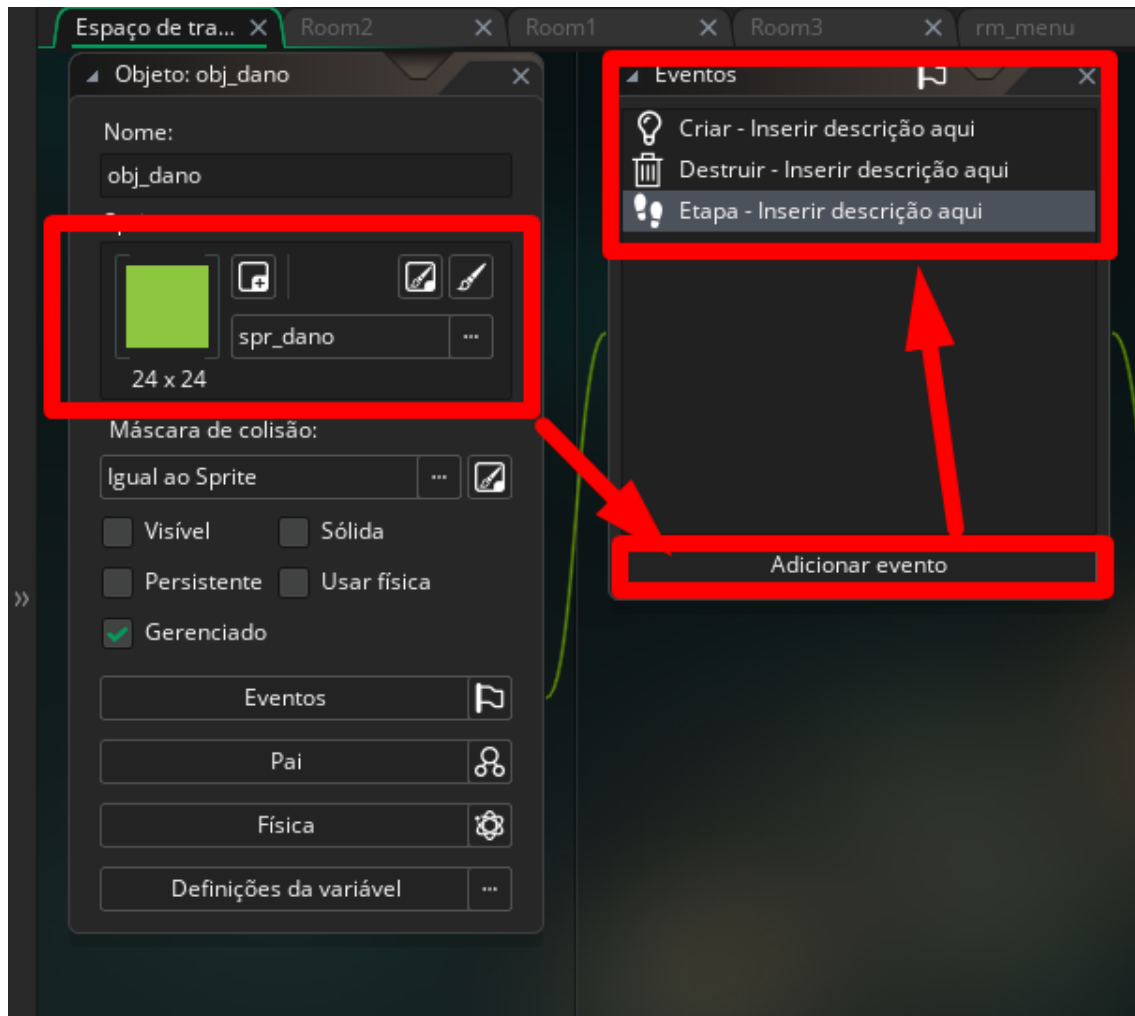
```
Etapa x Desenhar x Criar x
36     break;
37 }
38
39 case "andando":
40 {
41     temporizador_estado++;
42     if (sprite_index != spr_inimigo_raposa_run)
43     {
44         image_index = 0;
45         mid_velh = choose(.4, -.4) * global.vel_mult;
46     }
47     sprite_index = spr_inimigo_raposa_run;
48
49     //inverter a direção ao bater na parede
50     if (place_meeting(x + mid_velh, y, obj_block))
51     {
52         mid_velh *= -1;
53     }
54
55     if (irandom(temporizador_estado) > 300)
56     {
57         estado = choose("parado", "andando", "parado");
58         temporizador_estado = 0;
59     }
60     ScrAtkPlayerMelee(obj_player, dist, xscale);
61
62     break;
63 }
64
65 case "ataque":
66 {
67     //iniciando o delay
68     delay = room_speed * 2
69     mid_velh = 0;
```

```
Etapa x Desenhx x Criar x
70 if(sprite_index != spr_inimigo_raposa_atk)
71 {
72     image_index = 0;
73     possoatk = true;
74     dano = noone;
75 }
76 sprite_index = spr_inimigo_raposa_atk;
77
78 if(image_index > image_number - 1)
79 {
80     estado = "parado";
81 }
82
83 //criando o dano
84 if (image_index >= 6 && dano == noone && image_index < 8 && possoatk)
85 {
86     audio_play_sound(Espada, 10, false);
87     dano = instance_create_layer(x + sprite_width/2, y - sprite_height/3, layer, obj_dano);
88     dano.dano = ataque;
89     dano.pai = id;
90     possoatk = false;
91 }
92
93 //destruindo o dano
94 if (dano != noone && image_index >= 7)
95 {
96     instance_destroy(dano);
97     dano = noone;
98 }
99
100 break;
101 }
102
103 case "hit":
```

```
Etapa x Desenhx x Criar x
104 {
105     if (sprite_index != spr_inimigo_raposa_dano)
106     {
107         image_index=0;
108         //vida_atua --;
109     }
110     audio_play_sound(hit_sound_effect_12445, 8, false, 0.05)
111     sprite_index = spr_inimigo_raposa_dano;
112
113     if (vida_atual > 0)
114     {
115         if (image_index > image_number-1)
116         {
117             estado = "parado";
118         }
119     }
120     else
121     {
122         if (image_index >= 3)
123         {
124             estado = "dead";
125         }
126     }
127     break;
128 }
129
130 case "dead":
131 {
132     mid_velh = 0;
133     if(sprite_index != spr_inimigo_raposa_death)
134     {
135         image_index = 0;
136     }
137     sprite_index = spr_inimigo_raposa_death;
```

```
137     sprite_index = spr_inimigo_raposa_death;
138
139     if (image_index > image_number - 1)
140     {
141         image_speed = 0;
142         image_alpha -= .01;
143     }
144     break;
145 }
146 }
147
148
149
```

6 Crie um objeto parecido com o obj_block, para ser o dano das entidades



Insira os seguintes códigos:

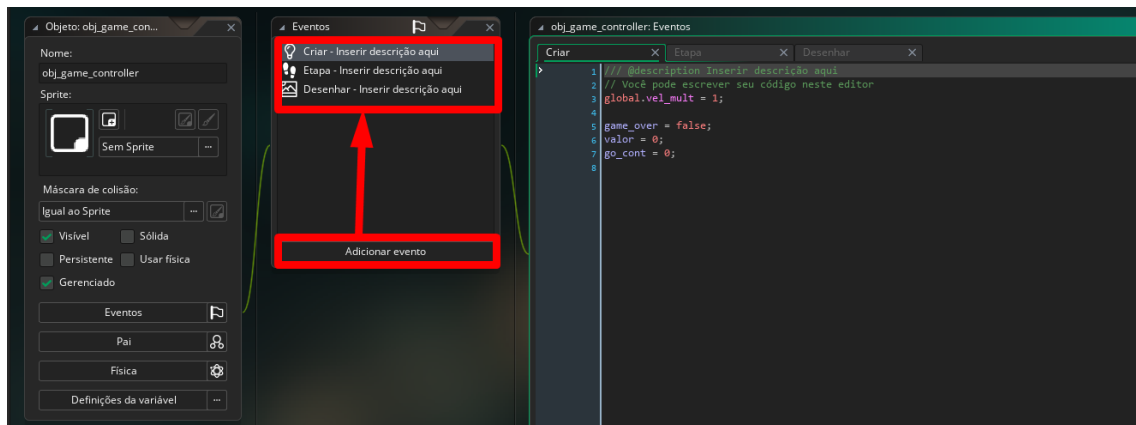
```
obj_dano: Eventos
Criar
1 /// @description Inserir descrição aqui
2 // Você pode escrever seu código neste editor
3 dano = 0;
4 pai = noone;
5 image_alpha = 1;
6
7
8 //lista para salvar em quem posso atacar
9 aplicar_dano = ds_list_create();
Etapa
Destruir
```

```
obj_dano: Eventos
Criar X Etapa X Destruir X
>
1  /// @description Inserir descrição aqui
2  // Você pode escrever seu código neste editor
3  var outro;
4  var outro_lista = ds_list_create();
5  var quant = instance_place_list(x, y, obj_entidade, outro_lista, 0);
6
7
8
9
10 for (var i = 0; i < quant; i++)
11 {
12     //checando o atual
13     var atual = outro_lista[i];
14
15     //checando se o atual está invencível
16     if (atual.invencivel)
17     {
18         continue;
19     }
20
21     if (object_get_parent(atual.object_index) != object_get_parent(pai.object_index))
22     {
23         //checar se posso dar dano
24         //checar se o atual já está na lista
25         var pos = ds_list_find_index(aplicar_dano, atual);
26         if (pos == -1)
27         {
28             ds_list_add(aplicar_dano, atual);
29         }
30     }
31 }
32
```

```
obj_dano: Eventos
Criar X Etapa X Destruir X
34 var tam = ds_list_size(aplicar_dano);
35 for (var i = 0; i < tam; i++)
36 {
37     outro = aplicar_dano[i].id;
38     if (outro.vida_atual > 0)
39     {
40         if (outro.delay <= 0)
41         {
42             outro.estado = "hit";
43             outro.image_index = 0;
44         }
45         outro.vida_atual -= dano;
46         instance_destroy();
47
48         //screenshake
49         if (object_get_parent(outro.object_index) == obj_inimigo_pai)
50         {
51             //screenshake apenas nos inimigos
52             screenshake(0.9);
53
54             //garantindo que o inimigo irá morrer
55             if (outro.vida_atual <= 0)
56             {
57                 outro.estado = "dead";
58             }
59         }
60     }
61 }
62 //destruindo as listas
63 ds_list_destroy(aplicar_dano);
64 ds_list_destroy(outro_lista);
65 instance_destroy();
```

```
obj_dano: Eventos
Criar x Etapa x Destruir x
1 /// @description Inserir descrição aqui
2 // Você pode escrever seu código neste editor
3 if (pai)
4 {
5     pai.dano = noone;
6 }
```

- 7 Agora crie um objeto para controlar algumas funcionalidades do jogo e adicione este objeto em todas as rooms

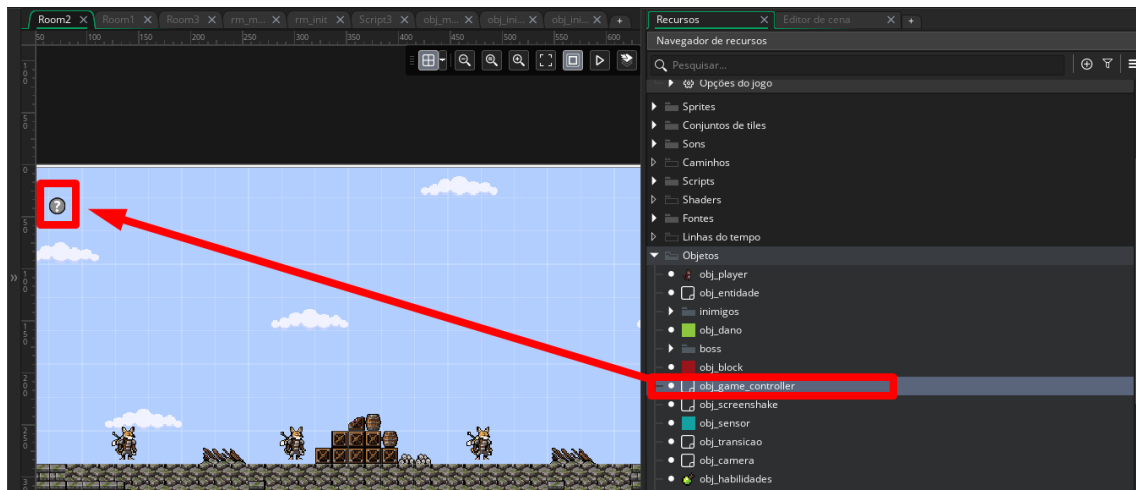


```
obj_game_controller: Eventos
Criar x Etapa x Desenhar x
1 /// @description Inserir descrição aqui
2 // Você pode escrever seu código neste editor
3
4 audio_play_sound(Floresta_01,1,true, 0.05)
5
6
7 if (game_over)
8 {
9     global.vel_mult = .5
10 }
11 else
12 {
13     global.vel_mult = 1;
14 }
15
```

```
obj_game_controller: Eventos
Criar X Etapa X Desenhar X
3
4
5 //tela de game-over
6 if (game_over)
7 {
8     audio_stop_all();
9     var x1 = camera_get_view_x(view_camera[0]);
10    var w = camera_get_view_width(view_camera[0]);
11    var x2 = x1 + w;
12    var meio_w = x1 + w / 2;
13    var y1 = camera_get_view_y(view_camera[0]);
14    var h = camera_get_view_height(view_camera[0]);
15    var y2 = y1 + h;
16    var meio_h = y1 + h / 2;
17
18    var qtd = h * .15;
19
20
21    valor = lerp(valor, 1, .05);
22
23
24    draw_set_color(c_black);
25    //escurecendo a tela
26    draw_set_alpha(valor - .3);
27    draw_rectangle(x1, y1, x2, y2, false);
28
29    //desenho retangulo superior
30    draw_set_alpha(1);
31    draw_rectangle(x1, y1, x2, y1 + qtd, false);
32
```

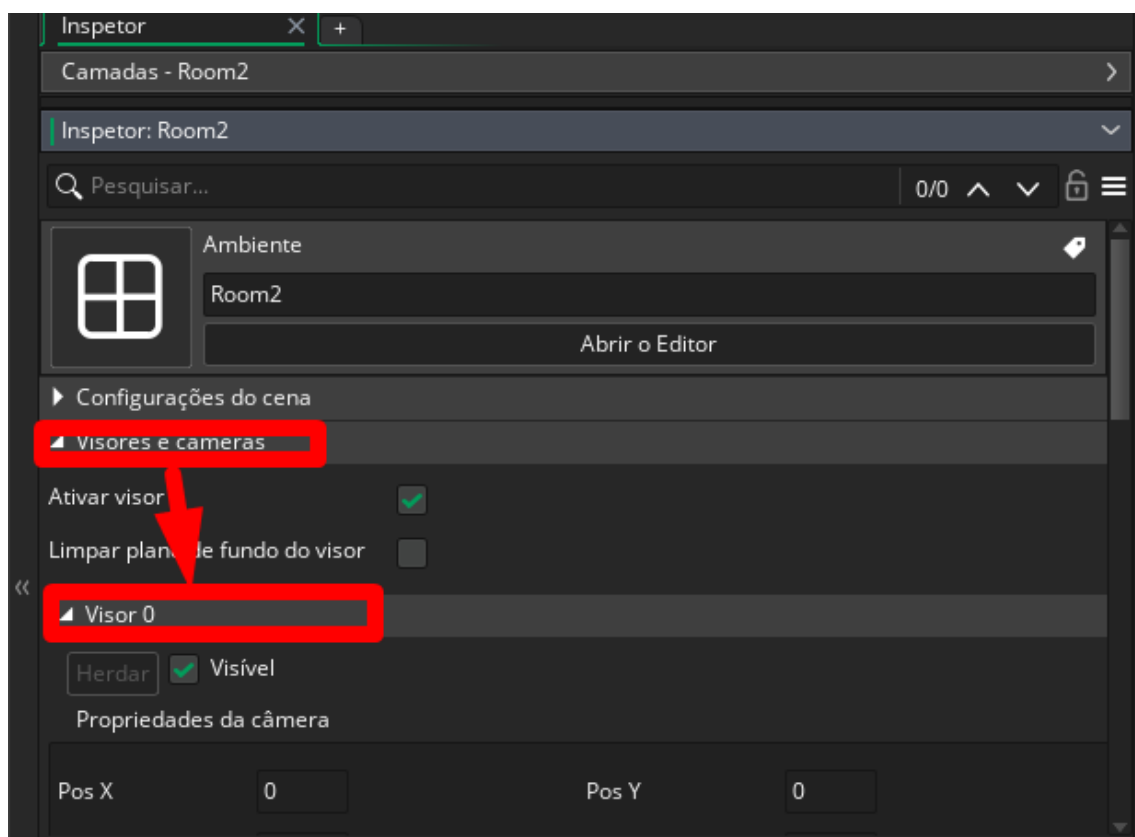
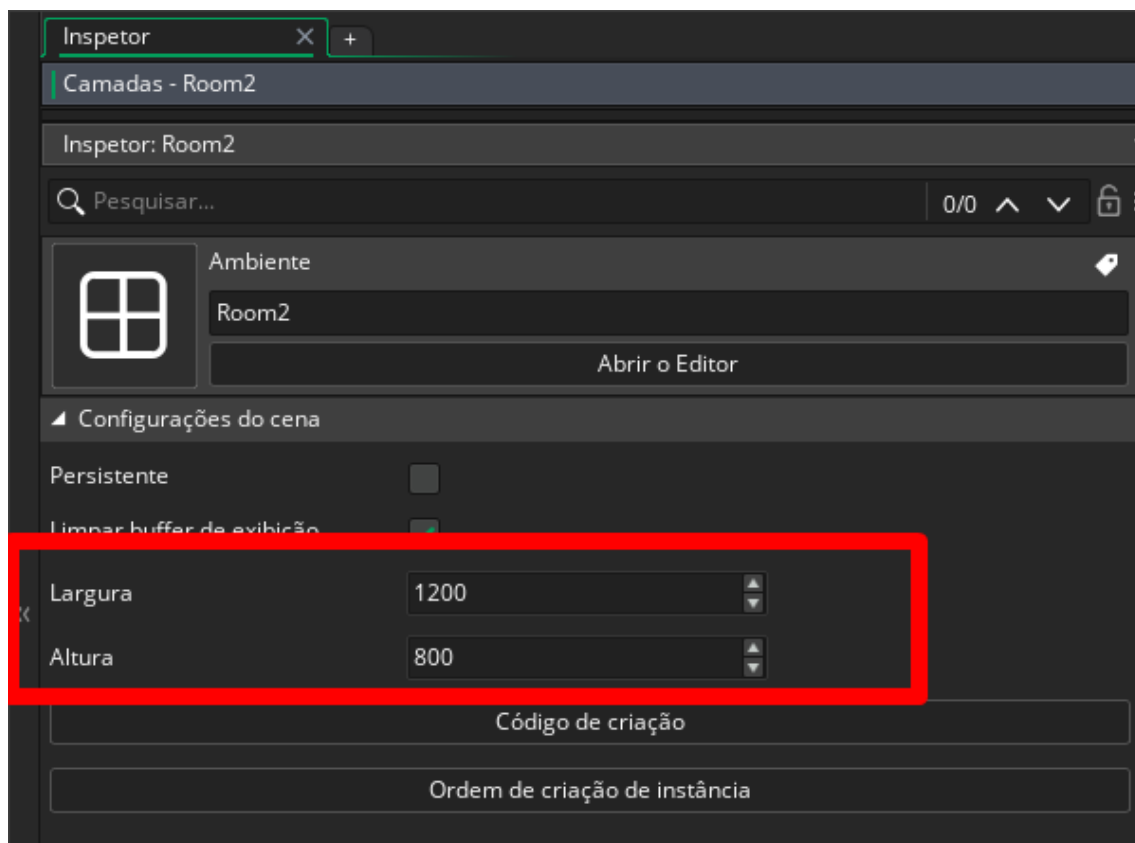
```
obj_game_controller: Eventos
Criar X Etapa X Desenhar X
32
33 //desenho retangulo inferior
34
35 draw_rectangle(x1, y2, x2, y2 - qtd * valor, false);
36
37 draw_set_alpha(1);
38 draw_set_color(-1);
39
40
41 if (valor > .85)
42 {
43     go_cont = lerp(go_cont, 1, .01);
44     //escurecendo game-over
45     draw_set_alpha(go_cont);
46     draw_set_font(fnt_gameover);
47     draw_set_valign(1);
48     draw_set_halign(1);
49     //sombra
50     draw_set_color(c_red);
51     draw_text(meio_w + 1, meio_h + 1, "G a m e - O v e r ");
52     //texto
53     draw_set_color(c_white);
54     draw_text(meio_w, meio_h, "G a m e - O v e r ");
55     draw_set_font(-1);
56
57     draw_text(meio_w, meio_h + 40, "Aperte Enter (ou Y) para reiniciar")
58     draw_set_valign(-1);
59     draw_set_halign(-1);
60 }
```

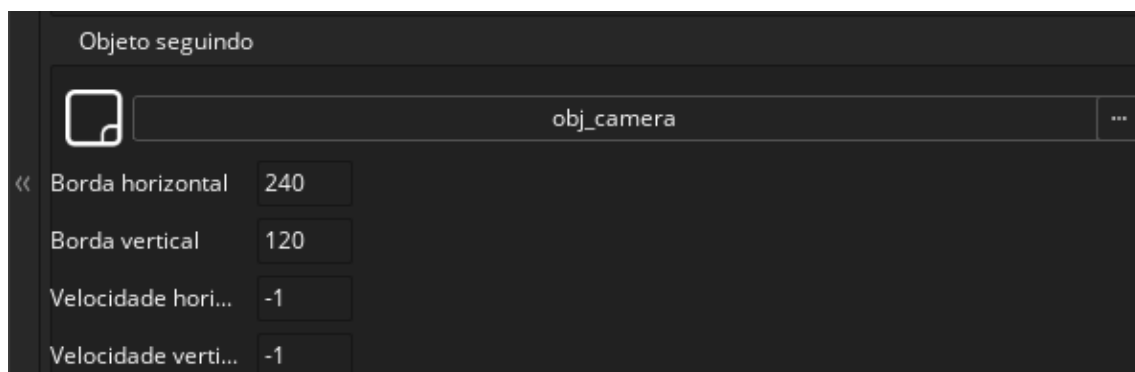
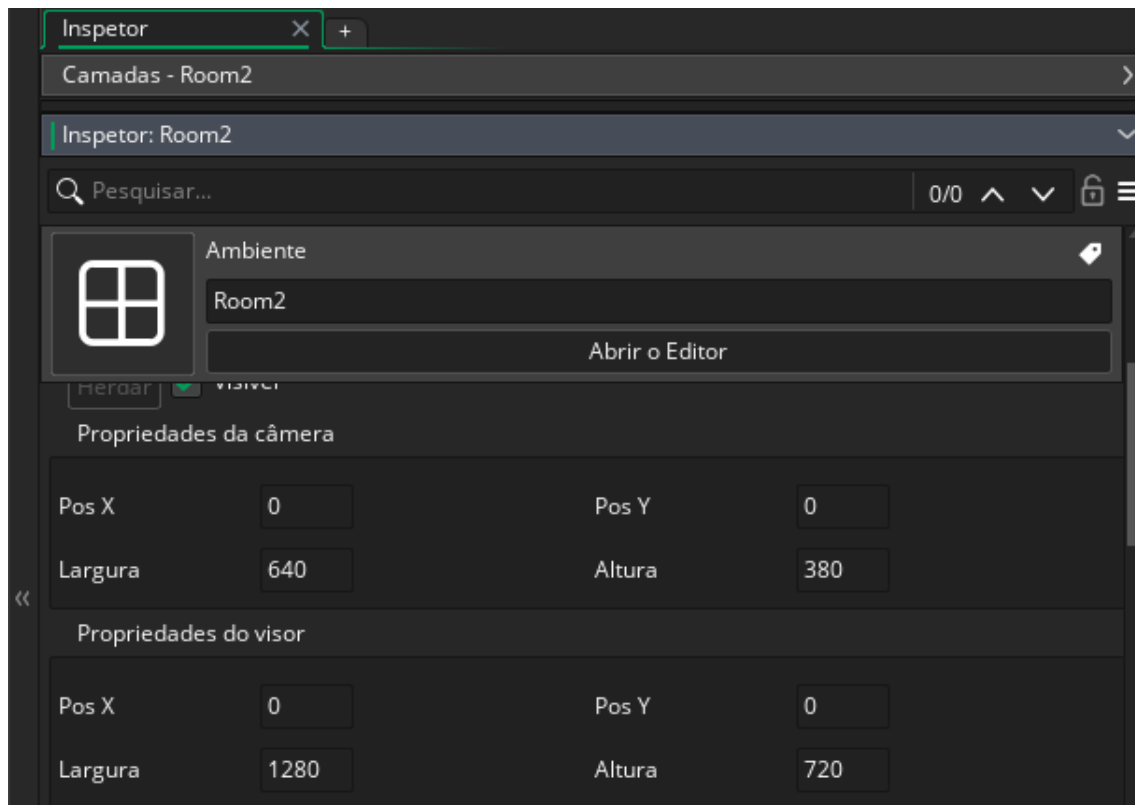
```
61  
62 }  
63 else  
64 {  
65  
66     valor = 0;  
67 }
```



8 Ajuste os visores e o tamanho da room

Obs: você pode deixa-las da maneira que desejar, porém esta é uma recomendação de tamanhos para as rooms e os visores para que fiquem agradáveis para os jogadores

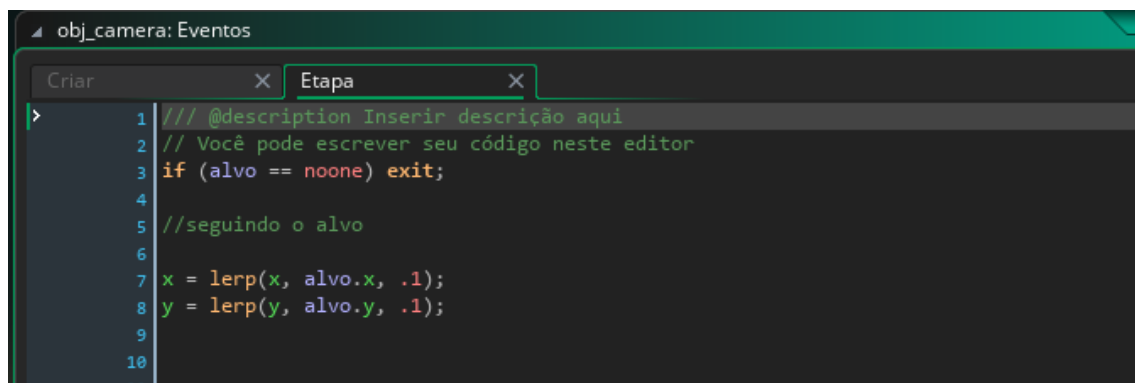
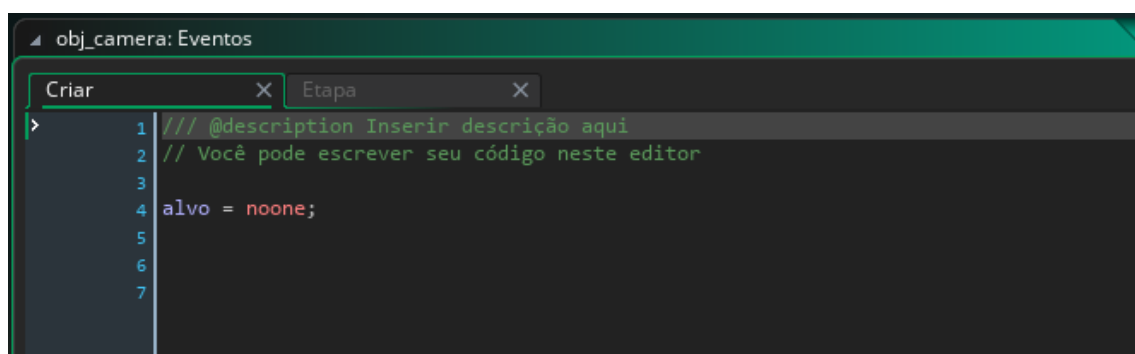
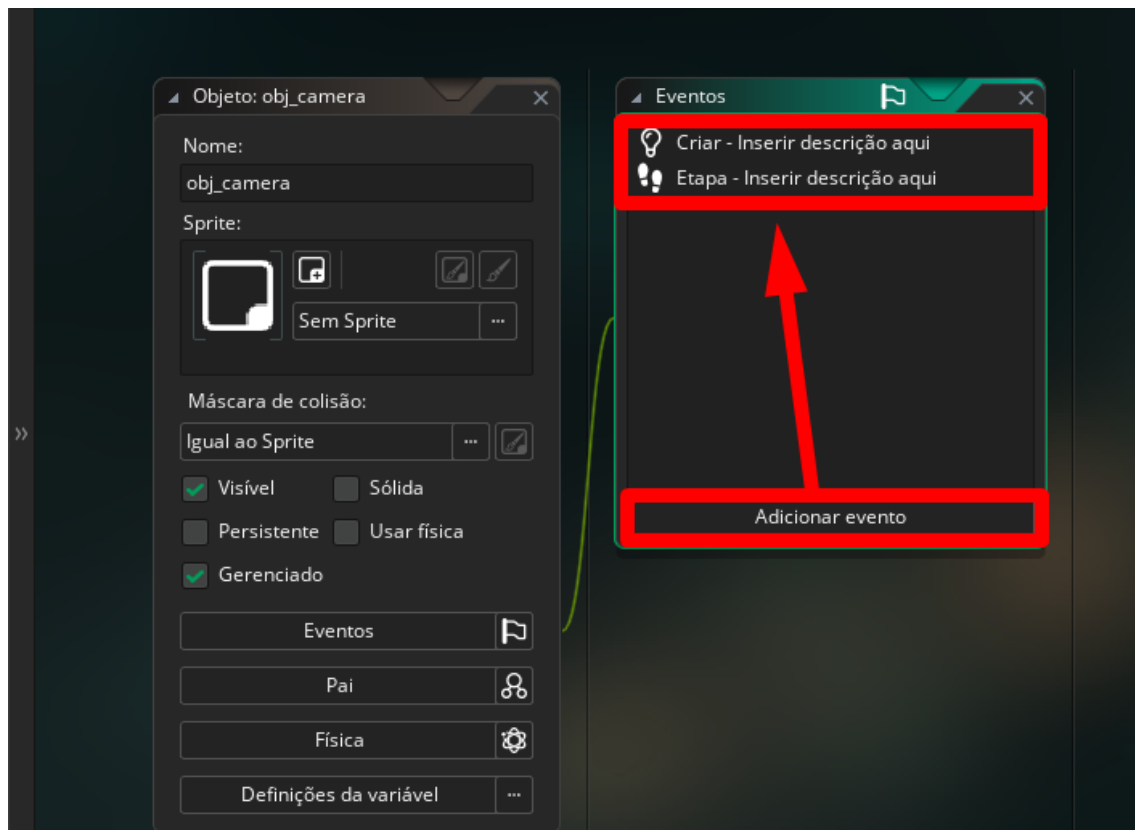




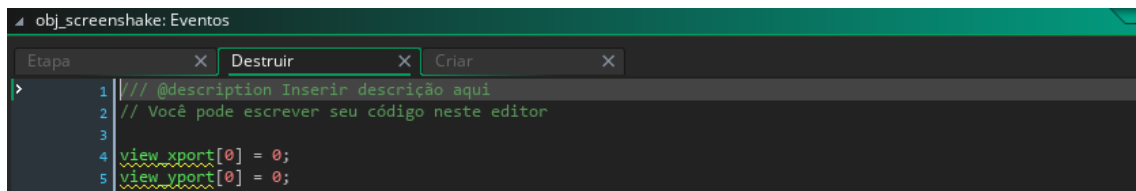
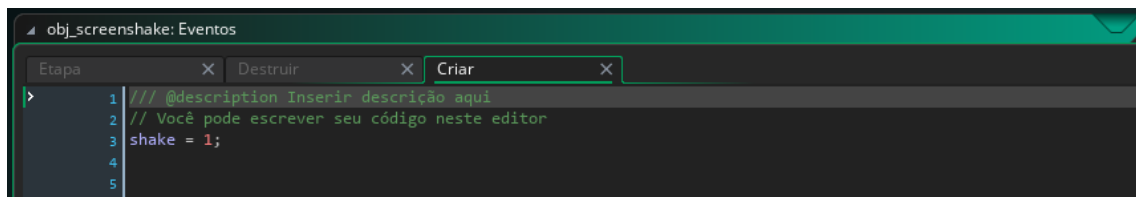
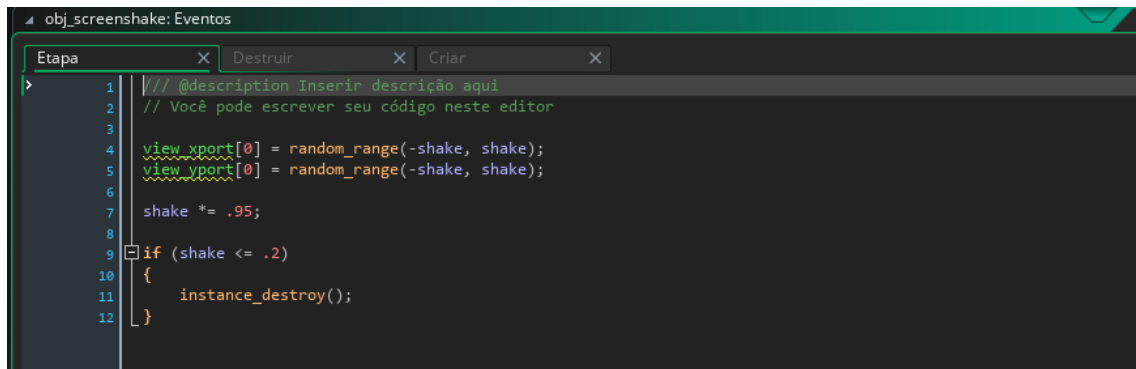
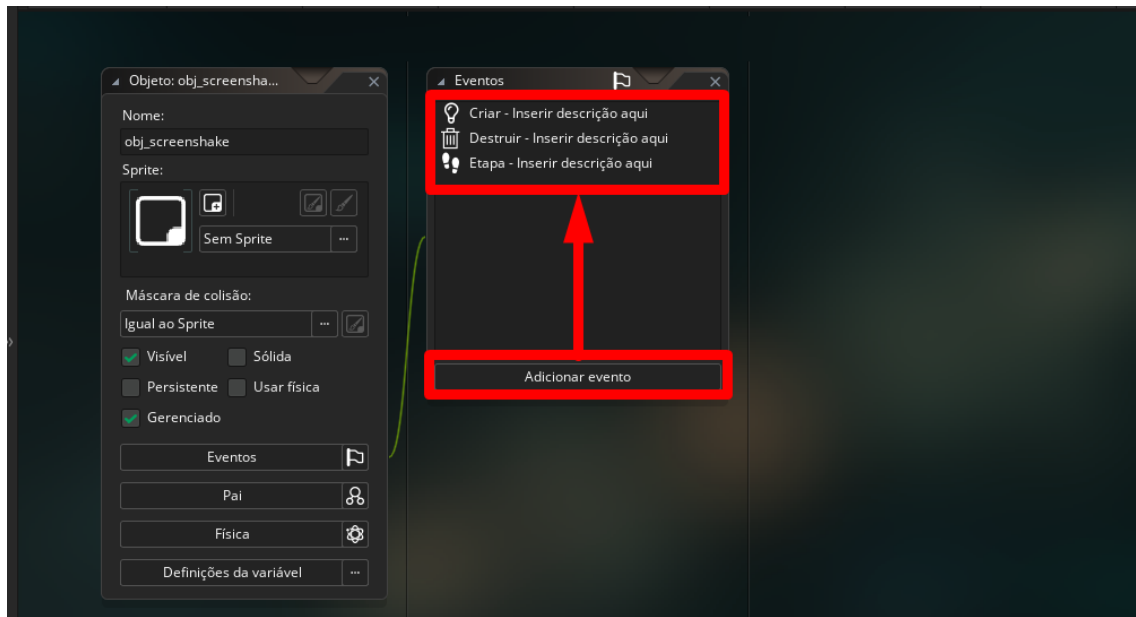
Faça essas modificações em todas as rooms do seu jogo

9 Criando a câmera

-Crie um objeto para colocar os códigos para a câmera

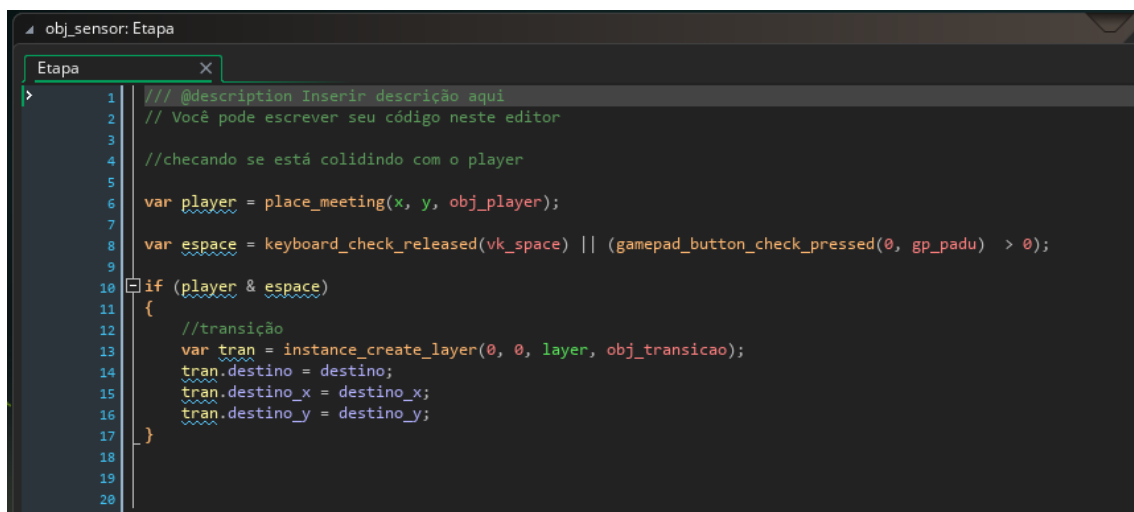
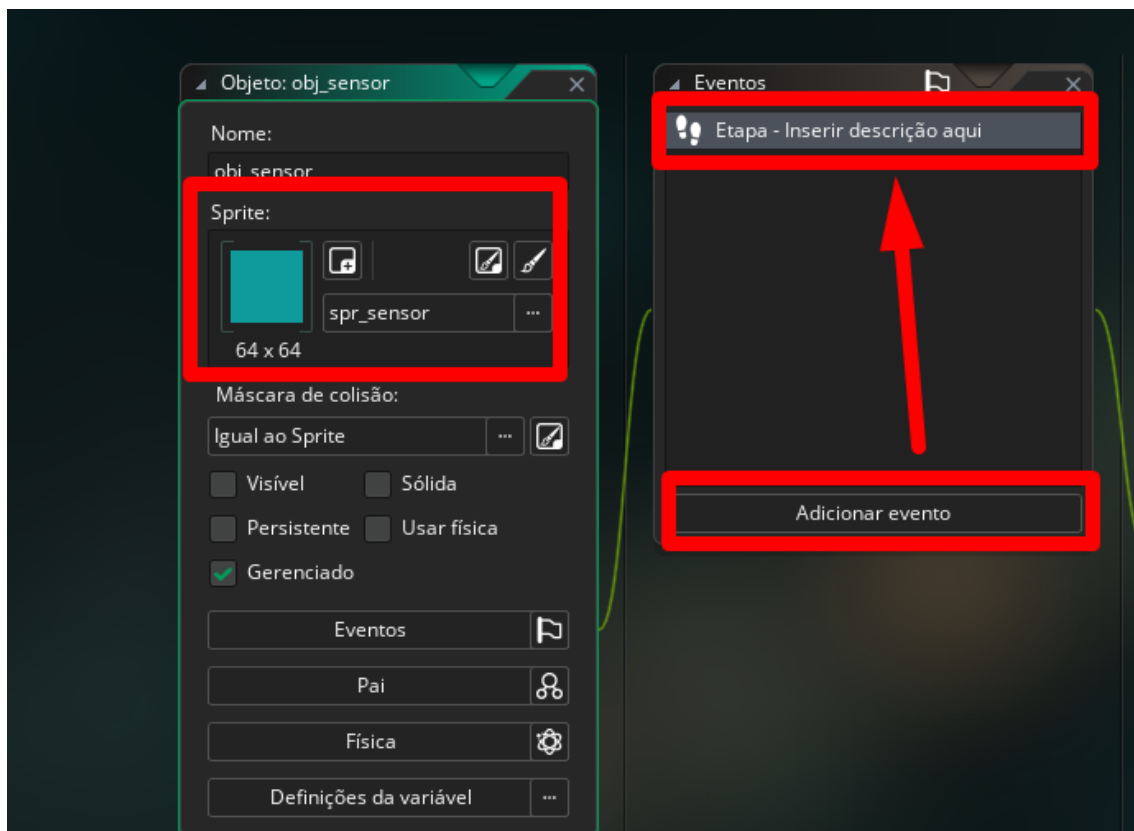


-Crie um objeto para fazermos esse efeito



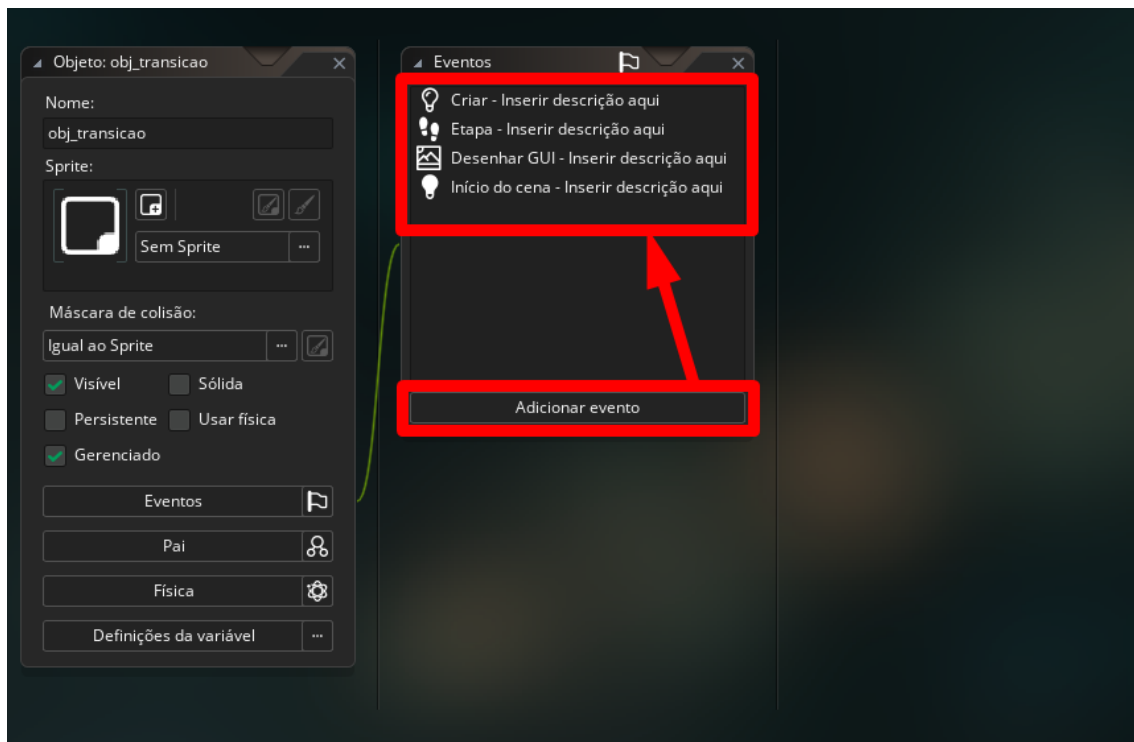
11 Vamos criar sensores para o player conseguir mudar de fase ao apertar um botão

-Esses sensores devem ficar em lugares do mapa que você achar melhor, o player deve ir até eles para mudar de fase.

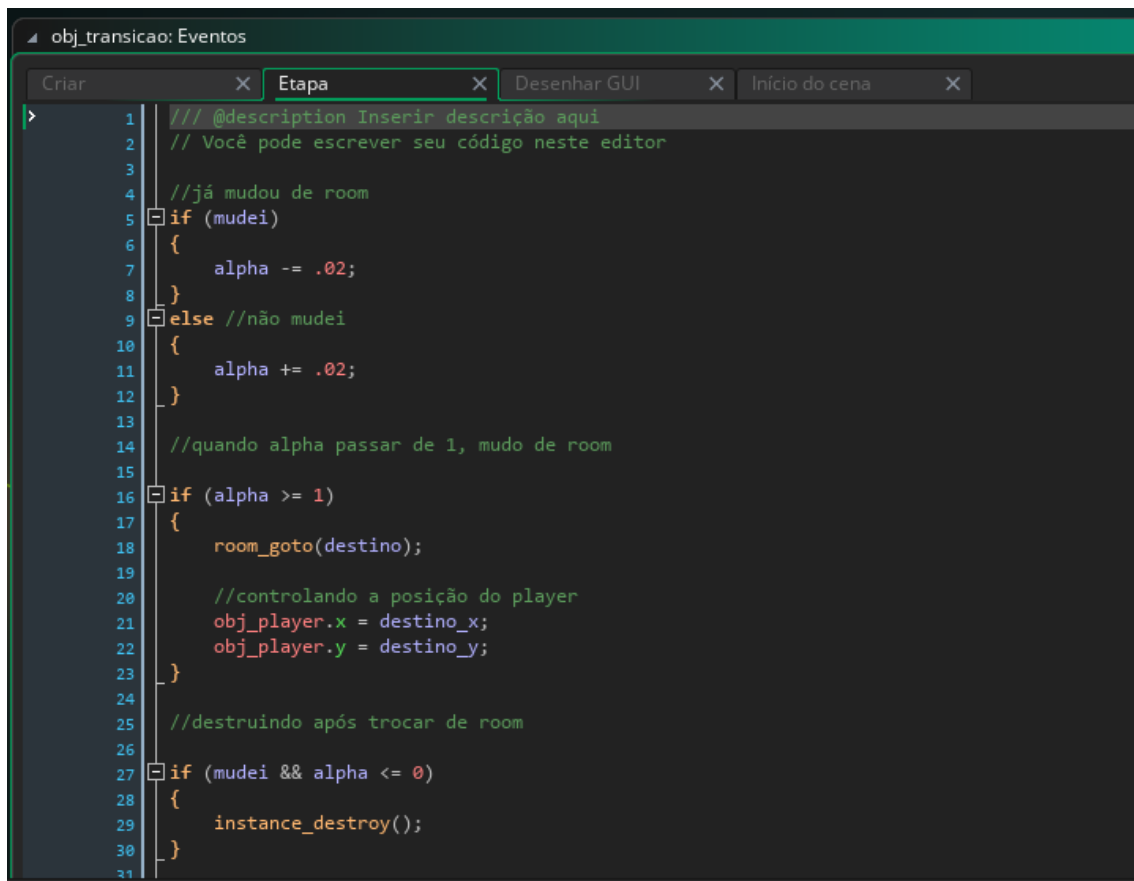


12 Agora precisamos de um efeito de transição ao trocar de fase

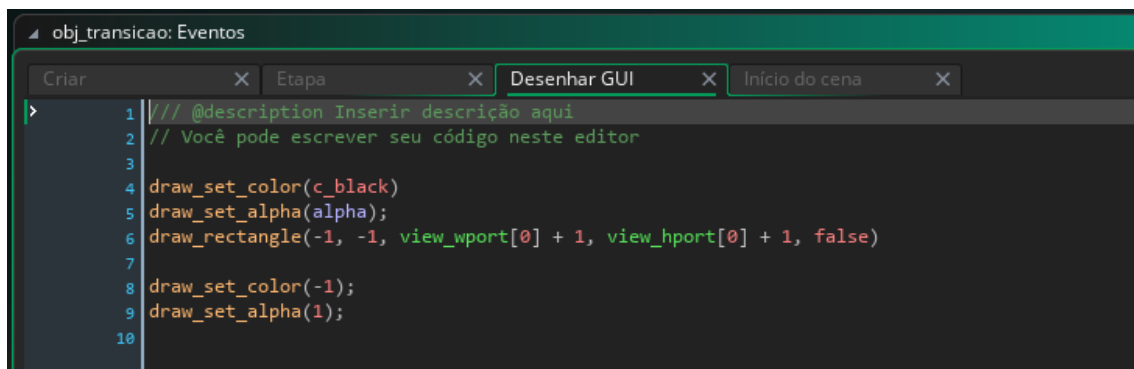
-Crie outro objeto para fazer isso



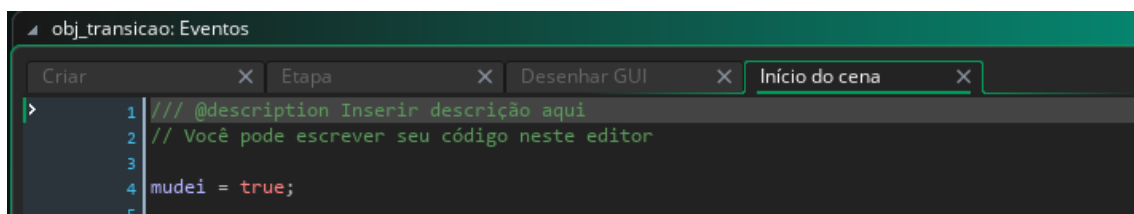
```
obj_transicao: Eventos
Criar
1 | /// @description Inserir descrição aqui
2 | // Você pode escrever seu código neste editor
3 |
4 | //transparência
5 | alpha = 0;
6 |
7 | //verificar se já mudei de room
8 | mudei = false;
9 |
10 |
```



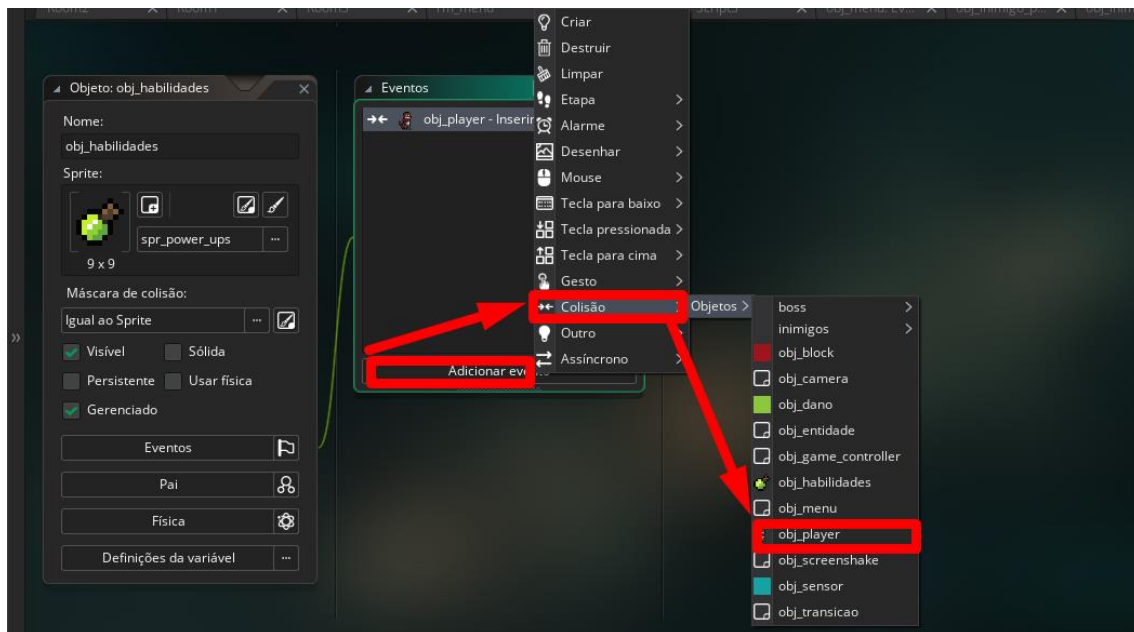
```
1  /// @description Inserir descrição aqui
2  // Você pode escrever seu código neste editor
3
4  //já mudou de room
5  if (mudei)
6  {
7      alpha -= .02;
8  }
9  else //não mudei
10 {
11     alpha += .02;
12 }
13
14 //quando alpha passar de 1, mudo de room
15
16 if (alpha >= 1)
17 {
18     room_goto(destino);
19
20     //controlando a posição do player
21     obj_player.x = destino_x;
22     obj_player.y = destino_y;
23 }
24
25 //destruindo após trocar de room
26
27 if (mudei && alpha <= 0)
28 {
29     instance_destroy();
30 }
31
```



```
1  /// @description Inserir descrição aqui
2  // Você pode escrever seu código neste editor
3
4  draw_set_color(c_black)
5  draw_set_alpha(alpha);
6  draw_rectangle(-1, -1, view_wport[0] + 1, view_hport[0] + 1, false)
7
8  draw_set_color(-1);
9  draw_set_alpha(1);
10
```



```
1  /// @description Inserir descrição aqui
2  // Você pode escrever seu código neste editor
3
4  mudei = true;
5
```



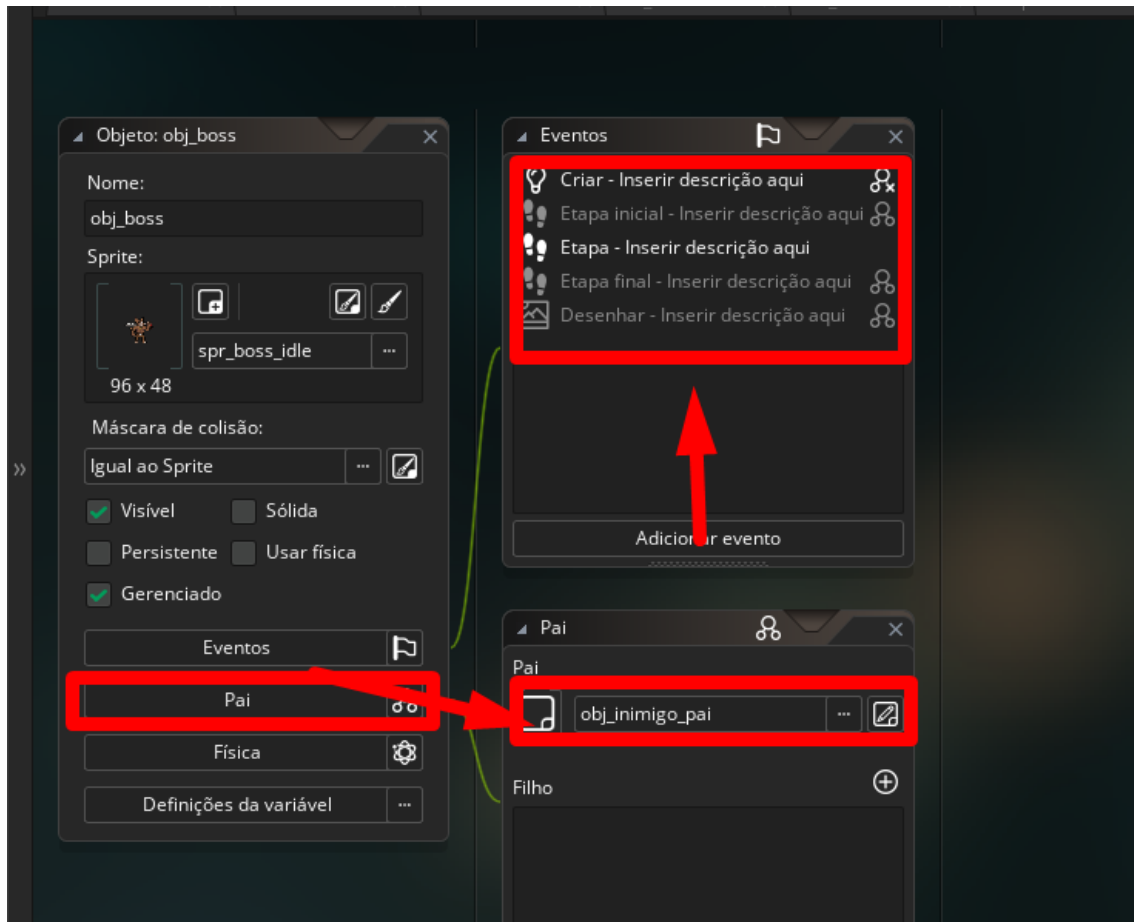
```

1  /// @description Inserir descrição aqui
2  // Você pode escrever seu código neste editor
3
4  global.power_ups[indice] = true;
5  audio_play_sound(short_success_sound_glockenspiel_treasure_video_game_6346, 9, false);
6  instance_destroy();
7
8
9
10

```

14 Boss

Crie um objeto para ele. Ele possui códigos parecidos com os outros inimigos, porém com algumas alterações:



```
obj_boss: Eventos
Etapa
Criar
7
8
9
10
11
12 // Inherit the parent event
13 event_inherited();
14
15 //atributos do boss
16 vida_max = 20 * global.dificuldade;
17 vida_atual = vida_max;
18
19 max_velh = 2;
20 max_velv = 3;
21 dash_vel = 4;
22
23 temporizador_estado = 0;
24
25 dist = 60;
26 dano = noone;
27 ataque = 2;
28
29 massa = 3;
30
31 taunt_delay = room_speed * 2;
32 taunt_timer = taunt_delay;
33
34
35 atk = irandom(2); //0, 1 ou 2
36
```



```
obj_boss: Eventos
Etapas
  Criar
  1 /// @description Inserir descrição aqui
  2 // Você pode escrever seu código neste editor
  3
  4
  5 var chao = place_meeting(x, y + 1, obj_block);
  6
  7 if (!chao)
  8 {
  9     velv += GRAVIDADE * massa * global.vel_mult;
  10 }
  11
  12 //state machine
  13 switch(estado)
  14 {
  15     case "parado":
  16     {
  17         //criar a lógica do estado parado
  18         //condição para trocar de estado
  19         if (sprite_index != spr_boss_idle)
  20         {
  21             sprite_index = spr_boss_idle;
  22             image_index = 0;
  23         }
  24
  25         //condições para trocar de estados
  26         //checando se o player está na tela
  27         if (instance_exists(obj_player))
  28         {
  29             var _dist = point_distance(x, y, obj_player.x, obj_player.y);
  30
  31             if (_dist < 300)
  32             {
  33                 estado = "andando";
  34             }
  35         }
  36
  37         break;
  38     }
  39
  40     case "andando":
  41     {
  42         screenshake(1);
  43         //criar a lógica do estado andando
  44
  45         if (sprite_index != spr_boss_run)
  46         {
  47             audio_play_sound(wet_metal_footsteps_32703, 6, false, 0.05)
  48             sprite_index = spr_boss_run;
  49             image_index = 0;
  50         }
  51
  52         //perseguir o player
  53         if (instance_exists(obj_player))
  54         {
  55             //minha distancia para o player
  56             var _dist = point_distance(x, y, obj_player.x, obj_player.y)
  57             var _dir = point_direction(x, y, obj_player.x, obj_player.y)
  58
  59             velv = 0;
  60             velh = 0;
  61             velx = 0;
  62             vely = 0;
  63             velr = 0;
  64             velb = 0;
  65             velg = 0;
  66             velg2 = 0;
  67             velg3 = 0;
  68             velg4 = 0;
  69             velg5 = 0;
  70             velg6 = 0;
  71             velg7 = 0;
  72             velg8 = 0;
  73             velg9 = 0;
  74             velg10 = 0;
  75             velg11 = 0;
  76             velg12 = 0;
  77             velg13 = 0;
  78             velg14 = 0;
  79             velg15 = 0;
  80             velg16 = 0;
  81             velg17 = 0;
  82             velg18 = 0;
  83             velg19 = 0;
  84             velg20 = 0;
  85             velg21 = 0;
  86             velg22 = 0;
  87             velg23 = 0;
  88             velg24 = 0;
  89             velg25 = 0;
  90             velg26 = 0;
  91             velg27 = 0;
  92             velg28 = 0;
  93             velg29 = 0;
  94             velg30 = 0;
  95             velg31 = 0;
  96             velg32 = 0;
  97             velg33 = 0;
  98             velg34 = 0;
  99             velg35 = 0;
  100             velg36 = 0;
  101             velg37 = 0;
  102             velg38 = 0;
  103             velg39 = 0;
  104             velg40 = 0;
  105             velg41 = 0;
  106             velg42 = 0;
  107             velg43 = 0;
  108             velg44 = 0;
  109             velg45 = 0;
  110             velg46 = 0;
  111             velg47 = 0;
  112             velg48 = 0;
  113             velg49 = 0;
  114             velg50 = 0;
  115             velg51 = 0;
  116             velg52 = 0;
  117             velg53 = 0;
  118             velg54 = 0;
  119             velg55 = 0;
  120             velg56 = 0;
  121             velg57 = 0;
  122             velg58 = 0;
  123             velg59 = 0;
  124             velg60 = 0;
  125             velg61 = 0;
  126             velg62 = 0;
  127             velg63 = 0;
  128             velg64 = 0;
  129             velg65 = 0;
  130             velg66 = 0;
  131             velg67 = 0;
  132             velg68 = 0;
  133             velg69 = 0;
  134             velg70 = 0;
  135             velg71 = 0;
  136             velg72 = 0;
  137             velg73 = 0;
  138             velg74 = 0;
  139             velg75 = 0;
  140             velg76 = 0;
  141             velg77 = 0;
  142             velg78 = 0;
  143             velg79 = 0;
  144             velg80 = 0;
  145             velg81 = 0;
  146             velg82 = 0;
  147             velg83 = 0;
  148             velg84 = 0;
  149             velg85 = 0;
  150             velg86 = 0;
  151             velg87 = 0;
  152             velg88 = 0;
  153             velg89 = 0;
  154             velg90 = 0;
  155             velg91 = 0;
  156             velg92 = 0;
  157             velg93 = 0;
  158             velg94 = 0;
  159             velg95 = 0;
  160             velg96 = 0;
  161             velg97 = 0;
  162             velg98 = 0;
  163             velg99 = 0;
  164             velg100 = 0;
  165             velg101 = 0;
  166             velg102 = 0;
  167             velg103 = 0;
  168             velg104 = 0;
  169             velg105 = 0;
  170             velg106 = 0;
  171             velg107 = 0;
  172             velg108 = 0;
  173             velg109 = 0;
  174             velg110 = 0;
  175             velg111 = 0;
  176             velg112 = 0;
  177             velg113 = 0;
  178             velg114 = 0;
  179             velg115 = 0;
  180             velg116 = 0;
  181             velg117 = 0;
  182             velg118 = 0;
  183             velg119 = 0;
  184             velg120 = 0;
  185             velg121 = 0;
  186             velg122 = 0;
  187             velg123 = 0;
  188             velg124 = 0;
  189             velg125 = 0;
  190             velg126 = 0;
  191             velg127 = 0;
  192             velg128 = 0;
  193             velg129 = 0;
  194             velg130 = 0;
  195             velg131 = 0;
  196             velg132 = 0;
  197             velg133 = 0;
  198             velg134 = 0;
  199             velg135 = 0;
  200             velg136 = 0;
  201             velg137 = 0;
  202             velg138 = 0;
  203             velg139 = 0;
  204             velg140 = 0;
  205             velg141 = 0;
  206             velg142 = 0;
  207             velg143 = 0;
  208             velg144 = 0;
  209             velg145 = 0;
  210             velg146 = 0;
  211             velg147 = 0;
  212             velg148 = 0;
  213             velg149 = 0;
  214             velg150 = 0;
  215             velg151 = 0;
  216             velg152 = 0;
  217             velg153 = 0;
  218             velg154 = 0;
  219             velg155 = 0;
  220             velg156 = 0;
  221             velg157 = 0;
  222             velg158 = 0;
  223             velg159 = 0;
  224             velg160 = 0;
  225             velg161 = 0;
  226             velg162 = 0;
  227             velg163 = 0;
  228             velg164 = 0;
  229             velg165 = 0;
  230             velg166 = 0;
  231             velg167 = 0;
  232             velg168 = 0;
  233             velg169 = 0;
  234             velg170 = 0;
  235             velg171 = 0;
  236             velg172 = 0;
  237             velg173 = 0;
  238             velg174 = 0;
  239             velg175 = 0;
  240             velg176 = 0;
  241             velg177 = 0;
  242             velg178 = 0;
  243             velg179 = 0;
  244             velg180 = 0;
  245             velg181 = 0;
  246             velg182 = 0;
  247             velg183 = 0;
  248             velg184 = 0;
  249             velg185 = 0;
  250             velg186 = 0;
  251             velg187 = 0;
  252             velg188 = 0;
  253             velg189 = 0;
  254             velg190 = 0;
  255             velg191 = 0;
  256             velg192 = 0;
  257             velg193 = 0;
  258             velg194 = 0;
  259             velg195 = 0;
  260             velg196 = 0;
  261             velg197 = 0;
  262             velg198 = 0;
  263             velg199 = 0;
  264             velg200 = 0;
  265             velg201 = 0;
  266             velg202 = 0;
  267             velg203 = 0;
  268             velg204 = 0;
  269             velg205 = 0;
  270             velg206 = 0;
  271             velg207 = 0;
  272             velg208 = 0;
  273             velg209 = 0;
  274             velg210 = 0;
  275             velg211 = 0;
  276             velg212 = 0;
  277             velg213 = 0;
  278             velg214 = 0;
  279             velg215 = 0;
  280             velg216 = 0;
  281             velg217 = 0;
  282             velg218 = 0;
  283             velg219 = 0;
  284             velg220 = 0;
  285             velg221 = 0;
  286             velg222 = 0;
  287             velg223 = 0;
  288             velg224 = 0;
  289             velg225 = 0;
  290             velg226 = 0;
  291             velg227 = 0;
  292             velg228 = 0;
  293             velg229 = 0;
  294             velg230 = 0;
  295             velg231 = 0;
  296             velg232 = 0;
  297             velg233 = 0;
  298             velg234 = 0;
  299             velg235 = 0;
  300             velg236 = 0;
  301             velg237 = 0;
  302             velg238 = 0;
  303             velg239 = 0;
  304             velg240 = 0;
  305             velg241 = 0;
  306             velg242 = 0;
  307             velg243 = 0;
  308             velg244 = 0;
  309             velg245 = 0;
  310             velg246 = 0;
  311             velg247 = 0;
  312             velg248 = 0;
  313             velg249 = 0;
  314             velg250 = 0;
  315             velg251 = 0;
  316             velg252 = 0;
  317             velg253 = 0;
  318             velg254 = 0;
  319             velg255 = 0;
  320             velg256 = 0;
  321             velg257 = 0;
  322             velg258 = 0;
  323             velg259 = 0;
  324             velg260 = 0;
  325             velg261 = 0;
  326             velg262 = 0;
  327             velg263 = 0;
  328             velg264 = 0;
  329             velg265 = 0;
  330             velg266 = 0;
  331             velg267 = 0;
  332             velg268 = 0;
  333             velg269 = 0;
  334             velg270 = 0;
  335             velg271 = 0;
  336             velg272 = 0;
  337             velg273 = 0;
  338             velg274 = 0;
  339             velg275 = 0;
  340             velg276 = 0;
  341             velg277 = 0;
  342             velg278 = 0;
  343             velg279 = 0;
  344             velg280 = 0;
  345             velg281 = 0;
  346             velg282 = 0;
  347             velg283 = 0;
  348             velg284 = 0;
  349             velg285 = 0;
  350             velg286 = 0;
  351             velg287 = 0;
  352             velg288 = 0;
  353             velg289 = 0;
  354             velg290 = 0;
  355             velg291 = 0;
  356             velg292 = 0;
  357             velg293 = 0;
  358             velg294 = 0;
  359             velg295 = 0;
  360             velg296 = 0;
  361             velg297 = 0;
  362             velg298 = 0;
  363             velg299 = 0;
  364             velg300 = 0;
  365             velg301 = 0;
  366             velg302 = 0;
  367             velg303 = 0;
  368             velg304 = 0;
  369             velg305 = 0;
  370             velg306 = 0;
  371             velg307 = 0;
  372             velg308 = 0;
  373             velg309 = 0;
  374             velg310 = 0;
  375             velg311 = 0;
  376             velg312 = 0;
  377             velg313 = 0;
  378             velg314 = 0;
  379             velg315 = 0;
  380             velg316 = 0;
  381             velg317 = 0;
  382             velg318 = 0;
  383             velg319 = 0;
  384             velg320 = 0;
  385             velg321 = 0;
  386             velg322 = 0;
  387             velg323 = 0;
  388             velg324 = 0;
  389             velg325 = 0;
  390             velg326 = 0;
  391             velg327 = 0;
  392             velg328 = 0;
  393             velg329 = 0;
  394             velg330 = 0;
  395             velg331 = 0;
  396             velg332 = 0;
  397             velg333 = 0;
  398             velg334 = 0;
  399             velg335 = 0;
  400             velg336 = 0;
  401             velg337 = 0;
  402             velg338 = 0;
  403             velg339 = 0;
  404             velg340 = 0;
  405             velg341 = 0;
  406             velg342 = 0;
  407             velg343 = 0;
  408             velg344 = 0;
  409             velg345 = 0;
  410             velg346 = 0;
  411             velg347 = 0;
  412             velg348 = 0;
  413             velg349 = 0;
  414             velg350 = 0;
  415             velg351 = 0;
  416             velg352 = 0;
  417             velg353 = 0;
  418             velg354 = 0;
  419             velg355 = 0;
  420             velg356 = 0;
  421             velg357 = 0;
  422             velg358 = 0;
  423             velg359 = 0;
  424             velg360 = 0;
  425             velg361 = 0;
  426             velg362 = 0;
  427             velg363 = 0;
  428             velg364 = 0;
  429             velg365 = 0;
  430             velg366 = 0;
  431             velg367 = 0;
  432             velg368 = 0;
  433             velg369 = 0;
  434             velg370 = 0;
  435             velg371 = 0;
  436             velg372 = 0;
  437             velg373 = 0;
  438             velg374 = 0;
  439             velg375 = 0;
  440             velg376 = 0;
  441             velg377 = 0;
  442             velg378 = 0;
  443             velg379 = 0;
  444             velg380 = 0;
  445             velg381 = 0;
  446             velg382 = 0;
  447             velg383 = 0;
  448             velg384 = 0;
  449             velg385 = 0;
  450             velg386 = 0;
  451             velg387 = 0;
  452             velg388 = 0;
  453             velg389 = 0;
  454             velg390 = 0;
  455             velg391 = 0;
  456             velg392 = 0;
  457             velg393 = 0;
  458             velg394 = 0;
  459             velg395 = 0;
  460             velg396 = 0;
  461             velg397 = 0;
  462             velg398 = 0;
  463             velg399 = 0;
  464             velg400 = 0;
  465             velg401 = 0;
  466             velg402 = 0;
  467             velg403 = 0;
  468             velg404 = 0;
  469             velg405 = 0;
  470             velg406 = 0;
  471             velg407 = 0;
  472             velg408 = 0;
  473             velg409 = 0;
  474             velg410 = 0;
  475             velg411 = 0;
  476             velg412 = 0;
  477             velg413 = 0;
  478             velg414 = 0;
  479             velg415 = 0;
  480             velg416 = 0;
  481             velg417 = 0;
  482             velg418 = 0;
  483             velg419 = 0;
  484             velg420 = 0;
  485             velg421 = 0;
  486             velg422 = 0;
  487             velg423 = 0;
  488             velg424 = 0;
  489             velg425 = 0;
  490             velg426 = 0;
  491             velg427 = 0;
  492             velg428 = 0;
  493             velg429 = 0;
  494             velg430 = 0;
  495             velg431 = 0;
  496             velg432 = 0;
  497             velg433 = 0;
  498             velg434 = 0;
  499             velg435 = 0;
  500             velg436 = 0;
  501             velg437 = 0;
  502             velg438 = 0;
  503             velg439 = 0;
  504             velg440 = 0;
  505             velg441 = 0;
  506             velg442 = 0;
  507             velg443 = 0;
  508             velg444 = 0;
  509             velg445 = 0;
  510             velg446 = 0;
  511             velg447 = 0;
  512             velg448 = 0;
  513             velg449 = 0;
  514             velg450 = 0;
  515             velg451 = 0;
  516             velg452 = 0;
  517             velg453 = 0;
  518             velg454 = 0;
  519             velg455 = 0;
  520             velg456 = 0;
  521             velg457 = 0;
  522             velg458 = 0;
  523             velg459 = 0;
  524             velg460 = 0;
  525             velg461 = 0;
  526             velg462 = 0;
  527             velg463 = 0;
  528             velg464 = 0;
  529             velg465 = 0;
  530             velg466 = 0;
  531             velg467 = 0;
  532             velg468 = 0;
  533             velg469 = 0;
  534             velg470 = 0;
  535             velg471 = 0;
  536             velg472 = 0;
  537             velg473 = 0;
  538             velg474 = 0;
  539             velg475 = 0;
  540             velg476 = 0;
  541             velg477 = 0;
  542             velg478 = 0;
  543             velg479 = 0;
  544             velg480 = 0;
  545             velg481 = 0;
  546             velg482 = 0;
  547             velg483 = 0;
  548             velg484 = 0;
  549             velg485 = 0;
  550             velg486 = 0;
  551             velg487 = 0;
  552             velg488 = 0;
  553             velg489 = 0;
  554             velg490 = 0;
  555             velg491 = 0;
  556             velg492 = 0;
  557             velg493 = 0;
  558             velg494 = 0;
  559             velg495 = 0;
  560             velg496 = 0;
  561             velg497 = 0;
  562             velg498 = 0;
  563             velg499 = 0;
  564             velg500 = 0;
  565             velg501 = 0;
  566             velg502 = 0;
  567             velg503 = 0;
  568             velg504 = 0;
  569             velg505 = 0;
  570             velg506 = 0;
  571             velg507 = 0;
  572             velg508 = 0;
  573             velg509 = 0;
  574             velg510 = 0;
  575             velg511 = 0;
  576             velg512 = 0;
  577             velg513 = 0;
  578             velg514 = 0;
  579             velg515 = 0;
  580             velg516 = 0;
  581             velg517 = 0;
  582             velg518 = 0;
  583             velg519 = 0;
  584             velg520 = 0;
  585             velg521 = 0;
  586             velg522 = 0;
  587             velg523 = 0;
  588             velg524 = 0;
  589             velg525 = 0;
  590             velg526 = 0;
  591             velg527 = 0;
  592             velg528 = 0;
  593             velg529 = 0;
  594             velg530 = 0;
  595             velg531 = 0;
  596             velg532 = 0;
  597             velg533 = 0;
  598             velg534 = 0;
  599             velg535 = 0;
  600             velg536 = 0;
  601             velg537 = 0;
  602             velg538 = 0;
  603             velg539 = 0;
  604             velg540 = 0;
  605             velg541 = 0;
  606             velg542 = 0;
  607             velg543 = 0;
  608             velg544 = 0;
  609             velg545 = 0;
  610             velg546 = 0;
  611             velg547 = 0;
  612             velg548 = 0;
  613             velg549 = 0;
  614             velg550 = 0;
  615             velg551 = 0;
  616             velg552 = 0;
  617             velg553 = 0;
  618             velg554 = 0;
  619             velg555 = 0;
  620             velg556 = 0;
  621             velg557 = 0;
  622             velg558 = 0;
  623             velg559 = 0;
  624             velg560 = 0;
  625             velg561 = 0;
  626             velg562 = 0;
  627             velg563 = 0;
  628             velg564 = 0;
  629             velg565 = 0;
  630             velg566 = 0;
  631             velg567 = 0;
  632             velg568 = 0;
  633             velg569 = 0;
  634             velg570 = 0;
  635             velg571 = 0;
  636             velg572 = 0;
  637             velg573 = 0;
  638             velg574 = 0;
  639             velg575 = 0;
  640             velg576 = 0;
  641             velg577 = 0;
  642             velg578 = 0;
  643             velg579 = 0;
  644             velg580 = 0;
  645             velg581 = 0;
  646             velg582 = 0;
  647             velg583 = 0;
  648             velg584 = 0;
  649             velg585 = 0;
  650             velg586 = 0;
  651             velg587 = 0;
  652             velg588 = 0;
  653             velg589 = 0;
  654             velg590 = 0;
  655             velg591 = 0;
  656             velg592 = 0;
  657             velg593 = 0;
  658             velg594 = 0;
  659             velg595 = 0;
  660             velg596 = 0;
  661             velg597 = 0;
  662             velg598 = 0;
  663             velg599 = 0;
  664             velg600 = 0;
  665             velg601 = 0;
  666             velg602 = 0;
  667             velg603 = 0;
  668             velg604 = 0;
  669             velg605 = 0;
  670             velg606 = 0;
  671             velg607 = 0;
  672             velg608 = 0;
  673             velg609 = 0;
  674             velg610 = 0;
  675             velg611 = 0;
  676             velg612 = 0;
  677             velg613 = 0;
  678             velg614 = 0;
  679             velg615 = 0;
  680             velg616 = 0;
  681             velg617 = 0;
  682             velg618 = 0;
  683             velg619 = 0;
  684             velg620 = 0;
  685             velg621 = 0;
  686             velg622 = 0;
  687             velg623 = 0;
  688             velg624 = 0;
  689             velg625 = 0;
  690             velg626 = 0;
  691             velg627 = 0;
  692             velg628 = 0;
  693             velg629 = 0;
  694             velg630 = 0;
  695             velg631 = 0;
  696             velg632 = 0;
  697             velg633 = 0;
  698             velg634 = 0;
  699             velg635 = 0;
  700             velg636 = 0;
  701             velg637 = 0;
  702             velg638 = 0;
  703             velg639 = 0;
  704             velg640 = 0;
  705             velg641 = 0;
  706             velg642 = 0;
  707             velg643 = 0;
  708             velg644 = 0;
  709             velg645 = 0;
  710             velg646 = 0;
  711             velg647 = 0;
  712             velg648 = 0;
  713             velg649 = 0;
  714             velg650 = 0;
  715             velg651 = 0;
  716             velg652 = 0;
  717             velg653 = 0;
  718             velg654 = 0;
  719             velg655 = 0;
  720             velg656 = 0;
  721             velg657 = 0;
  722             velg658 = 0;
  723             velg659 = 0;
  724             velg660 = 0;
  725             velg661 = 0;
  726             velg662 = 0;
  727             velg663 = 0;
  728             velg664 = 0;
  729             velg665 = 0;
  730             velg666 = 0;
  731             velg667 = 0;
  732             velg668 = 0;
  733             velg669 = 0;
  734             velg670 = 0;
  735             velg671 = 0;
  736             velg672 = 0;
  737             velg673 = 0;
  738             velg674 = 0;
  739             velg675 = 0;
  740             velg676 = 0;
  741             velg677 = 0;
  742             velg678 = 0;
  743             velg679 = 0;
  744             velg680 = 0;
  745             velg681 = 0;
  746             velg682 = 0;
  747             velg683 = 0;
  748             velg684 = 0;
  749             velg685 = 0;
  750             velg686 = 0;
  751             velg687 = 0;
  752             velg688 = 0;
  753             velg689 = 0;
  754             velg690 = 0;
  755             velg691 = 0;
  756             velg692 = 0;
  757             velg693 = 0;
  758             velg694 = 0;
  759             velg695 = 0;
  760             velg696 = 0;
  761             velg697 = 0;
  762             velg698 = 0;
  763             velg699 = 0;
  764             velg700 = 0;
  765             velg701 = 0;
  766             velg702 = 0;
  767             velg703 = 0;
  768             velg704 = 0;
  769             velg705 = 0;
  770             velg706 = 0;
  771             velg707 = 0;
  772             velg708 = 0;
  773             velg709 = 0;
  774             velg710 = 0;
  775             velg711 = 0;
  776             velg712 = 0;
  777             velg713 = 0;
  778             velg714 = 0;
  779             velg715 = 0;
  780             velg716 = 0;
  781             velg717 = 0;
  782             velg718 = 0;
  783             velg719 = 0;
  784             velg720 = 0;
  785             velg721 = 0;
  786             velg722 = 0;
  787             velg723 = 0;
  788             velg724 = 0;
  789             velg725 = 0;
  790             velg726 = 0;
  791             velg727 = 0;
  792             velg728 = 0;
  793             velg729 = 0;
  794             velg730 = 0;
  795             velg731 = 0;
  796             velg732 = 0;
  797             velg733 = 0;
  798             velg734 = 0;
  799             velg735 = 0;
  800             velg736 = 0;
  801             velg737 = 0;
```

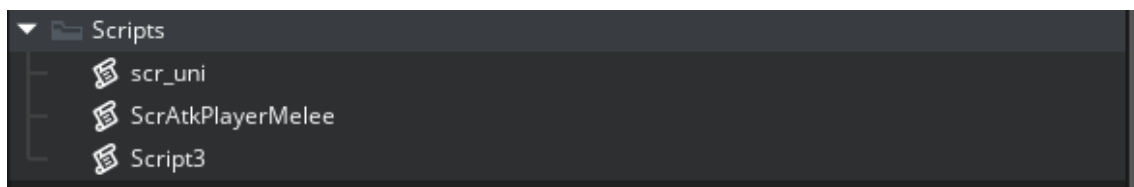
```
obj_boss: Eventos
Etapa
60 if (_dist > 30)
61 {
62     //velocidade
63     velh = lengthdir_x(max_velh, _dir);
64 }
65 else
66 {
67     //chegou perto, para de andar e ataca
68     velh = 0;
69     estado = "ataque";
70     //escolhendo o ataque
71     atk = irandom(2);
72 }
73
74 }
75
76 break;
77 }
78
79 case "ataque":
80 {
81
82
83     //criando o substate do ataque do boss
84     switch(atk)
85     {
86         //primeiro ataque
87         case 0:
88             atacando(spr_boss_atk_1, 2, 6, sprite_width / 2, - sprite_height/3, 2, 2, "taunt", Espada);
89             break;
```

```
obj_boss: Eventos
Etapa
90
91     //segundo ataque
92     case 1:
93         atacando(spr_boss_atk_2, 3, 6, sprite_width/2, - sprite_height/3, 6, 3, "taunt", Espada);
94         screenshake(2);
95         break;
96
97     //terceiro ataque
98     case 2:
99         atacando(spr_boss_atk_3, 2, 6, sprite_width / 2, - sprite_height/3, 4, 2, "taunt", Espada);
100         break;
101
102     }
103     break;
104 }
105
106 case "hit":
107 {
108     if (sprite_index != spr_boss_hurt)
109     {
110         image_index=0;
111         //vida_atua --;
112     }
113     audio_play_sound(hit_sound_effect_12445, 8, false, 0.05);
114     sprite_index = spr_boss_hurt;
115
116     if (vida_atual > 0)
117     {
118         if (image_index > image_number-1)
119         {
```

```
obj_boss: Eventos
Etapas
Criar
119 {
120     estado = "parado";
121 }
122 }
123 else
124 {
125     if (image_index >= 3)
126     {
127         estado = "dead";
128     }
129 }
130 break;
131 }
132
133 case "taunt":
134 {
135     taunt_timer--;
136     //definindo sprite
137     if (sprite_index != spr_boss_jump)
138     {
139         sprite_index = spr_boss_jump;
140         image_index = 0;
141     }
142     //condição para sair do estado
143     //player atacou
144     if (taunt_timer <= 0)
145     {
146         taunt_timer = taunt_delay;
147         estado = "parado"
148     }
149 }
```

```
149 break;
150 }
151
152 case "dead":
153 {
154     mid_velh = 0;
155     if(sprite_index != spr_boss_death)
156     {
157         image_index = 0;
158     }
159     sprite_index = spr_boss_death;
160
161     if (image_index > image_number - 1)
162     {
163         image_speed = 0;
164         image_alpha -= .01;
165     }
166
167 }
168 }
```

15 Vamos escrever alguns scripts, então na pasta dos scripts, crie 3



```
scr_uni.gml
1 #macro GRAVIDADE .3
2
```

```

ScrAtkPlayerMelee... X
1 // @arg player
2 // @arg dist
3 // @arg xscale
4 function ScrAtkPlayerMelee(argument0, argument1, argument2)
5 {
6     var outro = argument0;
7     var dist = argument1;
8     var xscale = argument2;
9
10    var player = collision_line(x, y - sprite_height/2, x + (dist * xscale), y - sprite_height/2, obj_player, 0, 1);
11
12    if (player)
13    {
14        estado = "ataque"
15    }
16 }

```

```

Script3.gml
1 // Os recursos de script mudaram para a v2.3.0; veja
2 // https://help.yoyogames.com/hc/en-us/articles/360005277377 para obter mais informações
3 function scr_functions(){
4
5 }
6
7 global.dificuldade = 1;
8
9 //enumerador para definir as minhas ações
10 enum menu_acoes
11 {
12     roda_metodo,
13     carrega_menu,
14     ajustes_menu
15 }
16 enum menus_lista
17 {
18     principal,
19     opcoes,
20     tela,
21     dificuldade
22 }
23
24 //screenshake
25 // @function screenshake(valor_da_tremida)
26 // @arg forçada_tremida
27
28 function screenshake(_treme)
29 {
30     var shake = instance_create_layer(0, 0, "instances", obj_screenshake);
31     shake.shake = _treme;
32 }
33

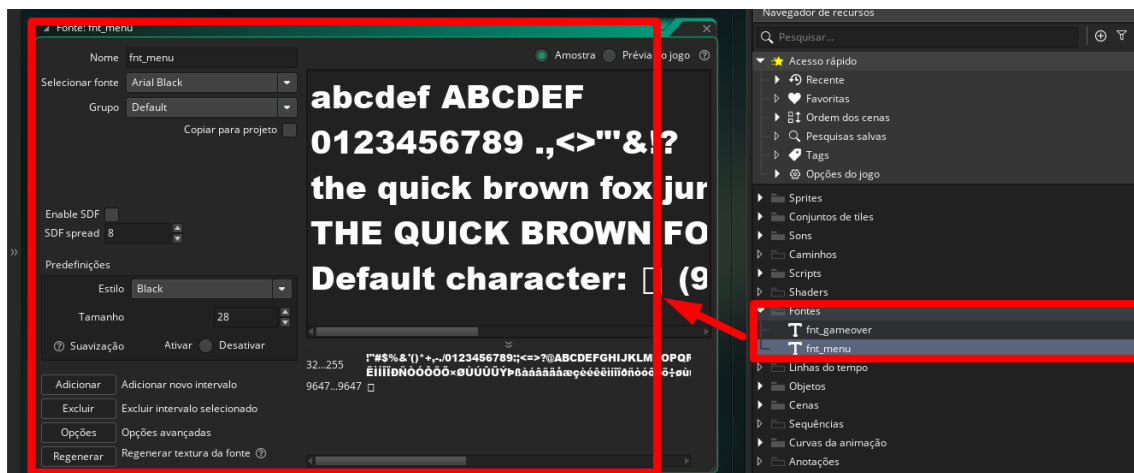
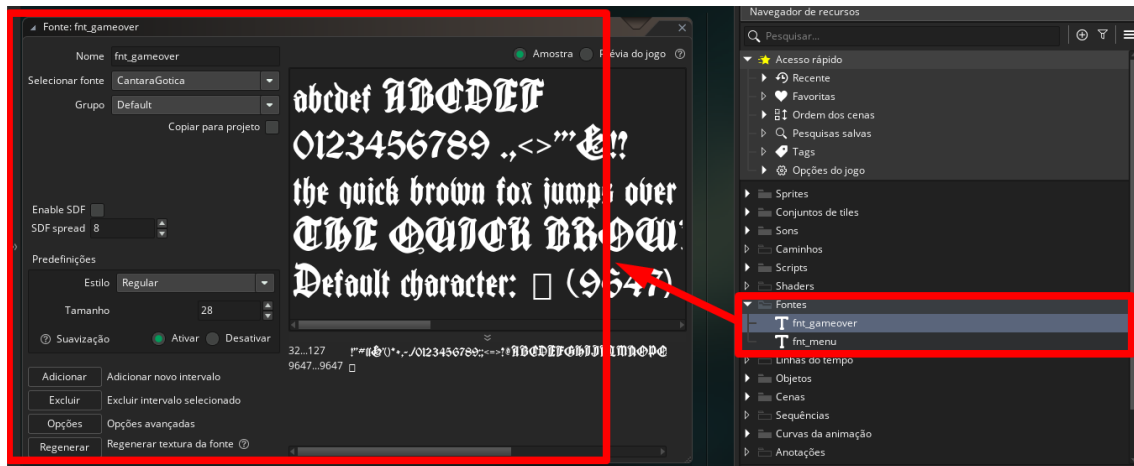
```

```

34
35 //define align
36 // @function define_align(vertical, horizontal)
37
38 function define_align(_ver, _hor)
39 {
40     draw_set_valign(_hor);
41     draw_set_valign(_ver);
42 }
43
44 //pegar o valor da curva de animação
45
46 // @function valor_ac(animation_curve, _animar, [canal])
47 function valor_ac(_anim, _animar = false, _chan = 0)
48 {
49     //posição da animação
50     static _pos = 0, _val = 0;
51
52     //aumentando o valor do pos
53     _pos += delta_time / 1000000;
54
55     if (_animar) _pos = 0;
56
57     //pegando o valor do canal
58     var _canal = animcurve_get_channel(_anim, _chan);
59     _val = animcurve_channel_evaluate(_canal, _pos);
60
61     return _val;
62 }

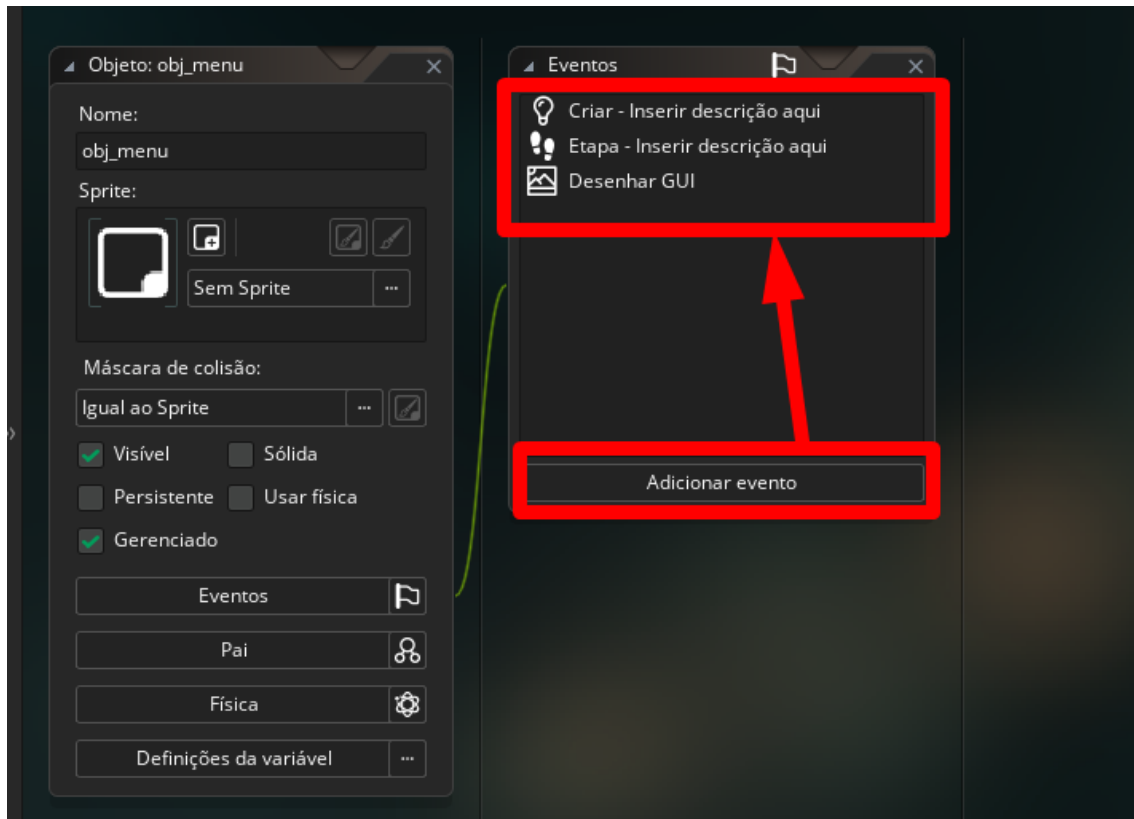
```

16 Importe duas fontes (tela de game-over e para o menu)



17 Menu

-Crie um objeto para fazer o menu



```
Criar | Etapa | Desenhar GUI |
1  /// @description Inserir descrição aqui
2  // Você pode escrever seu código neste editor
3
4  //criando menu
5
6  //seleção do menu
7  sel = 0;
8  marg_val= 32;
9  marg_total = 32;
10
11 //controlando a página do menu
12 pag = 0;
13
14 #region métodos
15 //desenha menu
16 desenha_menu = function(_menu)
17 {
18
19 //definindo fonte
20 draw_set_font(fnt_menu);
21 //alinhando o texto
22 define_align(0, 0);
23
24 //desenhando o menu
25 //pegando o tamanho do método
26 var _qtd = array_length(_menu);
27 //pegando a altura da tela
28 var _alt = display_get_gui_height();
29 //pegando a largura
30 var _larg = display_get_gui_width()
31
32 //definindo espaço entre linhas
33 var _espaco_y = string_height("I") + 16;
34 var _alt_menu = _espaco_y * _qtd;
```

```
Criar x Etapa x Desenhando GUI x
36 //desenhando as opções
37 for (var i = 0; i < _qtd; i++)
38 {
39     var _cor = c_white, _marg_x = 0;
40     //desenhando o item do menu
41
42     var _texto = _menu[i][0];
43
44     //checando se a seleção esta no texto atual
45     if (menus_sel[pag] == i)
46     {
47         _cor = c_red;
48         _marg_x = marg_val;
49     }
50
51     draw_text_color(20 + _marg_x, (_alt/2) - _alt_menu / 2 + (i * _espaco_y), _texto, _cor, _cor, _cor, _cor, 1);
52 }
53
54 //desenha o outro lado do menu (as opções quando elas existirem)
55 for (var i = 0; i < _qtd; i++)
56 {
57     //checar se eu preciso desenhar as opções de fato
58     switch(_menu[i][1])
59     {
60         case menu.acoes.ajustes.menu:
61         {
62             //desenhando as opções do lado direito
63             //salvando índice que estou
64             var _indice = _menu[i][3];
65             var _txt = _menu[i][4][_indice];
66         }
67     }
68 }
```

```
Criar x Etapa x Desenhando GUI x
69 //Só posso ir para a esquerda somente se eu não estiver no índice 0
70 var _esq = _indice > 0 ? "<<" : "";
71 //Só posso ir para a direita somente se eu não estiver no final do vetor
72 var _dir = _indice < array_length(_menu[i][4]) - 1 ? ">>" : "";
73
74 var _cor = c_white;
75 //se estou mexendo nessa opção, mudo de cor
76 if (alterando && menus_sel[pag] == i) _cor = c_red;
77
78 draw_text_color(_larg/2, (_alt/2) - _alt_menu / 2 + (i * _espaco_y), _esq + _txt + _dir, _cor, _cor, _cor, _cor, 1);
79
80 break;
81 }
82 }
83
84 //resetando os draws_sets
85 draw_set_font(-1);
86
87 define_align(-1, -1);
88
89 }
90
91 //controlando o menu
92 controla_menu = function(_menu)
93 {
94     var _up, _down, _avanca, _recua, _left, _right;
95
96     var _sel = menus_sel[pag];
97
98     static _animar = false
99 }
```

```
Criar x Etapa x Desenhando GUI x
103 _up = keyboard_check_pressed(vk_up) || (gamepad_button_check_pressed(0, gp_padu) > 0);
104 _down = keyboard_check_pressed(vk_down) || (gamepad_button_check_pressed(0, gp_padd) > 0);
105 _avanca = keyboard_check_released(vk_enter) || (gamepad_button_check_pressed(0, gp_facel) > 0);
106 _recua = keyboard_check_pressed(vk_escape) || (gamepad_button_check_pressed(0, gp_shoulderl) > 0);
107 _left = keyboard_check_pressed(vk_left) || (gamepad_button_check_pressed(0, gp_padr1) > 0);
108 _right = keyboard_check_pressed(vk_right) || (gamepad_button_check_pressed(0, gp_padr) > 0);
109
110 //checando se eu não estou alterando as opções do jogo
111 if (!alterando)
112 {
113     if (_up or _down)
114     {
115         //mudando o valor do sel
116         menus_sel[pag] += _down - _up;
117
118         //limitando o sel dentro do vetor
119         var _tam = array_length(_menu) - 1;
120         menus_sel[pag] = clamp(menus_sel[pag], 0, _tam);
121
122         _animar = true;
123     }
124 }
125 else
126 {
127     //ouseja, eu estou alterando as opções
128     _animar = false;
129
130     //se apertar para a esquerda ou para a direita, eu mexo nas opções
131     if (_right or _left)
132     {
133         //achando meu limite
134         var _limite = array_length(_menu[_sel][4]) - 1;
135     }
136 }
```

```

137 // mudando meu indice
138 menus[pag][_sel][3] += _right - _left;
139 // garantindo que eu não vou sair do limite
140 menus[pag][_sel][3] = clamp(menus[pag][_sel][3], 0, _limite);
141 }
142 }
143
144 //O que fazer quando apertar o enter
145 if (_avanca)
146 {
147     switch(_menu[_sel][1])
148     {
149         // caso seja 0 ele roda um metodo
150         case menu.acoes.roda_metodo: _menu[_sel][2](); break;
151         // mudar o valor da pag
152         case menu.acoes.carrega_menu: pag = _menu[_sel][2]; break;
153         case menu.acoes.ajustes_menu:
154             alterando = !alterando;
155
156         // rodando o metodo
157         if (!alterando)
158         {
159             var _arg = _menu[_sel][3]
160             _menu[_sel][2](_arg);
161         }
162
163         break;
164     }
165 }
166 }
167
168

```

```

167
168
169
170     if (_animar)
171     {
172         marg_val = marg_total * valor_ac(ac_marg, _up ^^ _down);
173     }
174 }
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200

```

```

199     }
200 }
201
202 ajusta_dificuldade = function(_valor)
203 {
204     switch(_valor)
205     {
206         case 0: global.dificuldade = .5; break;
207         case 1: global.dificuldade = 1; break;
208         case 2: global.dificuldade = 1.5; break;
209         case 3: global.dificuldade = 3; break;
210     }
211 }
212 #endregion
213
214 //quando eu apertar enter no iniciar, ele vai rodar um metodo
215 //quando eu apertar enter no opções, ele vai carregar o menu de opções
216 //quando eu apertar enter no sair ele vai rodar o método fechar jogo
217
218 //texto - ação - conteúdo da ação
219
220 menu_principal = [
221     ["Iniciar", menu_acoes.roda_metodo, inicia_jogo],
222     ["Opções", menu_acoes.carrega_menu, menu_lista.opcoes],
223     ["Sair", menu_acoes.roda_metodo, fecha_jogo]
224 ];
225
226 menu_opcoes = [
227     ["Tipo de janela", menu_acoes.carrega_menu, menu_lista.tele],
228     ["Dificuldade", menu_acoes.carrega_menu, menu_lista.dificuldade],
229     ["Controles", menu_acoes.roda_metodo, test],
230     ["Voltar", menu_acoes.carrega_menu, menu_lista.principal]
231 ];

```



```

232
233 menu_dificuldades = [
234     ["Dificuldade", menu_acoes.ajustes_menu, test, 1, ["Fácil", "Normal", "Difícil", "Impossível"]],
235     ["Voltar", menu_acoes.carrega_menu, menus.lista.opcoes]
236 ];
237
238 menu_tela = [
239     ["Tipo de tela", menu_acoes.ajustes_menu, ajusta_tela, 1, ["Tela cheia", "janela"]],
240     ["Voltar", menu_acoes.carrega_menu, menus.lista.opcoes]
241 ];
242
243 //salvando os menus
244
245 menus = [menu_principal, menu_opcoes, menu_tela, menu_dificuldades];
246
247 //salvando a seleção de cada menu
248 menus_sel = array_create(array_length(menus), 0);
249
250 alterando = false;
251

```

```

Criar  X  Etapa  X  Desenhando GUI  X
1 /// @description Inserir descrição aqui
2 // Você pode escrever seu código neste editor
3
4 //alterando a seleção+;
5
6 //pegando as teclas
7 controla_menu(menus[pag]);

```

```

Criar  X  Etapa  X  Desenhando GUI  X
1 desenha_menu(menus[pag]);

```

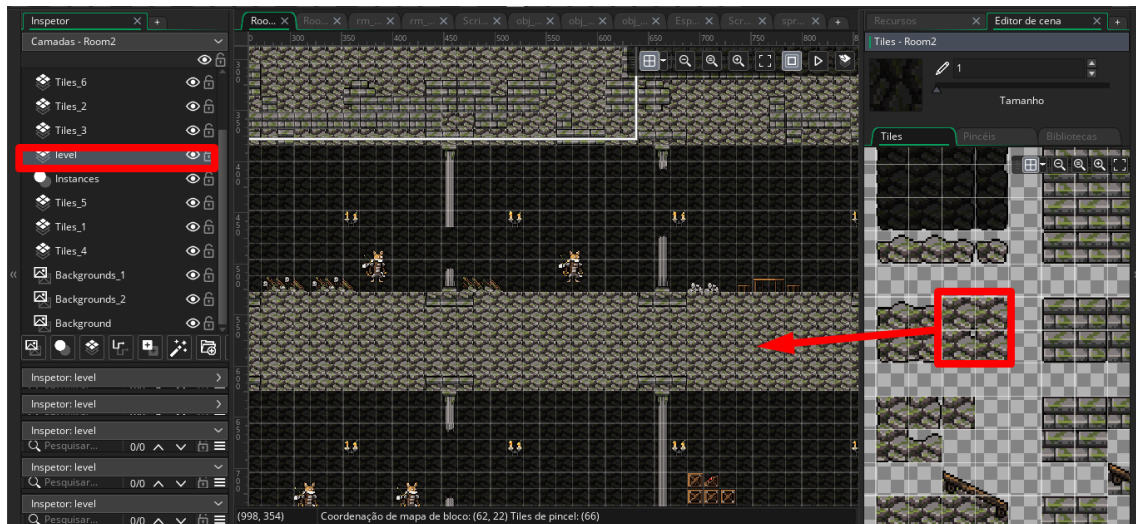
18 Crie uma curva de animação a sua escolha



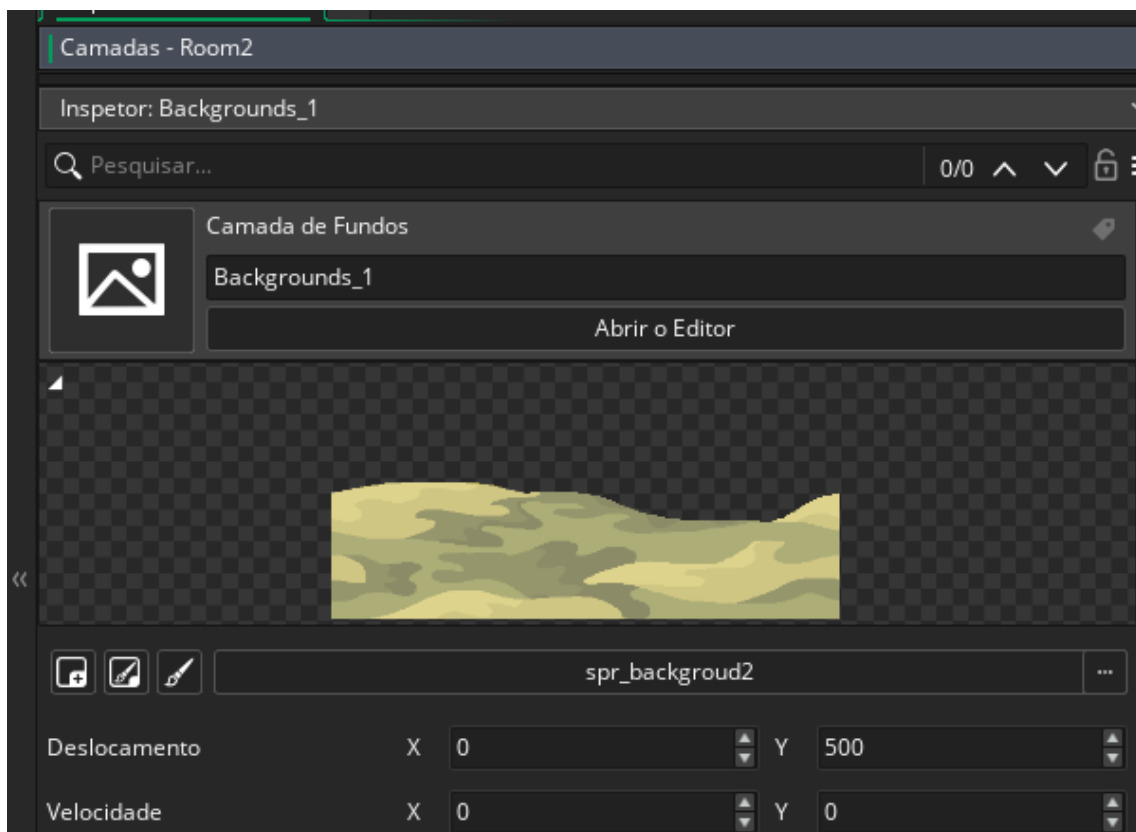
19 Finalizando o jogo

-Coloque os tilessets nas rooms

-Crie várias camadas de tilessets para colocar no seu jogo (Camadas que ficam acima tem prioridades para aparecer)



-Mude o background



-Repita esse processo em todas as rooms

Fim do manual