

INSTITUTO FEDERAL – SÃO PAULO



**INSTITUTO
FEDERAL**

São Paulo

Campus
São Paulo

**ENSINO MÉDIO INTEGRADO COM O TÉCNICO DE
DESENVOLVIMENTO DE SISTEMAS**

Diandro Silva Souto Júnior

João Victor Ferreira e Silva

Lorena Vieira Marques

Mirela De Fátima Fernandes Dos Santos

Paulo Henrique Bispo de Jesus Pereira

A Raiz Do Medo

São Paulo, SP

2025

Diandro Silva Souto Júnior

João Victor Ferreira e Silva

Lorena Vieira Marques

Mirela De Fátima Fernandes Dos Santos

Paulo Henrique Bispo de Jesus Pereira

A Raiz do Medo

Documentação de Projeto Interdisciplinar apresentado como requisito parcial para avaliação nas disciplinas de Lógica de Programação, Arquitetura de Computadores e Redes, Princípios de Desenvolvimento Web, Informática e Ferramentas Para Desenvolvimento.

Prof(a). Ana Lucia Grici Zacarin Mamede

Prof(o). André Evandro Lourenço

Prof(o). Antonio Ferreira Viana

Prof(a). Claudete de Oliveira Alves

Prof(a). Claudia Miyuki Werhmuller

Prof(a). Daniela dos Santos Santana

Prof(o). João Antonio Temochko Andre

Prof(a). Paula Neves de Araújo

Prof(o). Paulo Henrique Netto de Alcantara

São Paulo, SP

2025

Sumário

Introdução.....	4
Planejamento do Projeto.....	5
Desenvolvimento.....	6
Design e Arte.....	6
Arquitetura e Código.....	10
O Site de Divulgação.....	12
Protótipo e Testes.....	16
Conclusão.....	17
Referências.....	17
Apêndice.....	18

Introdução

O desenvolvimento do jogo “A Raiz Do Medo” é o resultado direto de uma atividade curricular fundamental na disciplina de Lógica de Programação. O requisito principal desta atividade era a criação de um jogo digital funcional, proporcionando à equipe total autonomia e liberdade criativa quanto ao gênero, temática e dimensão espacial do produto final. O objetivo deste documento é servir como o registro formal e abrangente de todo o processo de desenvolvimento, detalhando o planejamento, o design conceitual, a implementação técnica e os protótipos funcionais que resultaram no jogo.

Após um rigoroso processo de levantamento de dados e discussões internas, a decisão unânime foi pela criação do jogo “A Raiz Do Medo”, um puzzle game de terror psicológico em plataforma 2D, projetado para envolver o jogador em um ciclo de tensão e mistério. A narrativa aprisiona o jogador na pele de Eddie, que acorda em seu próprio quarto, agora irreconhecível e decadente. Sua única rota para a "liberdade" está codificada em complexas equações matemáticas deixadas em cartas enigmáticas. O sucesso em decifrar esses enigmas o permite avançar, mas o fracasso o confronta com uma morte iminente, forçando-o a descobrir a verdadeira "raiz" de seu medo para quebrar o ciclo.

A justificativa para esta escolha temática e de design reside na ambição de combinar a tensão inerente ao gênero de terror psicológico com a lógica fria dos puzzles, onde a matemática se torna a fonte primária de ansiedade e punição. A decisão de desenvolver em duas dimensões 2D foi estratégica, visando otimizar o tempo de produção no GameMaker, permitindo que a equipe focasse intensamente na qualidade do pixel art e na atmosfera imersiva, em vez da complexidade da modelagem 3D. Nosso público-alvo primário são jogadores jovens e adultos a partir de 10 anos que apreciam puzzles complexos e narrativas sombrias, buscando um desafio intelectual combinado com a imersão emocional.

O leitor encontrará, nas seções subsequentes, o detalhamento de todas as fases do projeto. Este documento apresenta o planejamento, o design, o desenvolvimento técnico e os protótipos do jogo "A Raiz Do Medo", abordando o escopo do projeto, o Design Document com regras, personagens, arte e a implementação do código na engine GameMaker.

Planejamento do Projeto

Nosso projeto se concentra na criação de um jogo de terror psicológico onde a premissa principal é a fuga de um hospício. Inicialmente, o jogador enfrenta um desafio de matemática, sendo obrigado a resolver contas para conseguir passar de uma sala para outra. À medida que o jogo avança, a dificuldade aumenta, introduzindo desafios mais complexos como fugir de monstros, resolver enigmas e outras tarefas necessárias para chegar à saída e, finalmente, desvendar o desfecho da história.

Para desenvolver a estética e as mecânicas do jogo, buscamos inspiração em jogos que empregam a técnica de pixel art e utilizam uma visão 2D. As referências mais significativas foram os jogos "Sally Face" e "Five Nights at Freddy's Into The Pit", que serviram como modelos para definir a aparência dos cenários, o design dos sprites e a implementação de certas mecânicas de jogo. No aspecto técnico, o aprendizado se deu por meio de tutoriais focados no GameMaker. Estes foram essenciais para a criação de elementos como o menu e para a compreensão de funcionalidades do engine, como a gestão de "Rooms" e a criação de "Eventos". Embora a maior parte dos tutoriais tenha sido dedicada a ensinar as bases do desenvolvimento, eles foram cruciais para a fundação do projeto.

A escolha do GameMaker como engine foi motivada pela recomendação da professora e por sua notável simplicidade de programação, além de oferecer um vasto conjunto de ferramentas que facilitam o processo de criação de jogos. Naturalmente, optamos por programar em GML (GameMaker Language), pois é a linguagem nativa do engine e nos pareceu ser a opção mais acessível e simples para iniciar o desenvolvimento. Além do GameMaker, utilizamos outras ferramentas importantes, como vídeos do YouTube para aprender a criar funcionalidades como o menu, e alguns sites para obter sprites de monstros e personagens, bem como sons. Por fim, empregamos Inteligências Artificiais com dupla função: auxiliar na criação de cenários e atuar como ferramenta de debugging, ajudando a resolver bugs de código, a entender o propósito de códigos específicos e a identificar as razões por trás de erros de programação.

Desenvolvimento

Design e Arte:

O estilo visual principal escolhido para "A Raiz Do Medo" é o Pixel Art sombrio e atmosférico. Essa escolha foi crucial, pois a natureza do terror psicológico se beneficia imensamente de detalhes que sugerem decadência e abandono, elementos que o pixel art pode entregar com forte impacto emocional. A paleta de cores é dominada por tons dessaturados, cinzas, marrons e pretos profundos, com o uso ocasional de luzes contrastantes como velas ou um rastro de lanterna para guiar o olhar do jogador e aumentar a tensão. O objetivo não é o realismo, mas sim a criação de uma estética de pesadelo que reforça o sentimento de aprisionamento de Eddie.

Os cenários de "A Raiz Do Medo" não são meros planos de fundo; eles são extensões do estado mental de Eddie e ferramentas ativas para reforçar o terror psicológico e a claustrofobia do escape room. O level design em 2D foi construído para ser visualmente opressor, utilizando a estética pixel art sombria para saturar os ambientes com detalhes de decadência e negligência. As imagens a seguir demonstram o design de ambientes, cada um nomeado de acordo com sua função e atmosfera no ciclo do pesadelo.

O Recinto do Despertar (Quarto de Eddie)

Este é o ponto de partida e de recomeço do ciclo. O Recinto do Despertar é a representação distorcida do quarto de Eddie. Suas paredes rachadas, cobertas de musgo e teias de aranha, e a luz vermelha fraca pendurada no teto, substituem qualquer vestígio de conforto. O ambiente serve como um gatilho imediato para a ansiedade, pois o jogador reconhece o local, mas o encontra em um estado de profunda e inexplicável deterioração. A única saída é a porta trancada, forçando a interação imediata com a primeira equação. As imagens abaixo ilustram a transformação do familiar para o aterrorizante.



O Salão das Escolhas (Sala da Escadaria)

Este é o elo de transição entre os puzzles e o centro da casa. O Salão das Escolhas é marcado por uma grande escadaria decadente, quadros sombrios e portas duplas entreabertas que convidam o jogador a descer para a próxima fase. A iluminação esverdeada fantasmal das arandelas sugere a presença de algo sobrenatural e iminente. Este cenário, embora mais aberto que o quarto, aumenta a sensação de desespero e perda de orientação espacial, preparando o jogador para a iminência de tomar uma decisão de alto risco.



O Limbo da Tentação (Sala das Três Portas)

Após resolver os problemas matemáticos do segundo nível, Eddie se depara com o Limbo da Tentação. Três portas idênticas se materializam em um ambiente de luz esverdeada e paredes mofadas. Este cenário representa o teste final de sorte e intuição na fase intermediária. Duas portas levam a um "fim" súbito (morte), e apenas uma conduz ao nível inferior. A ausência de puzzles neste local coloca toda a pressão na decisão, transformando o ambiente em um momento crucial de terror psicológico puro



O Quarto do Eco(O quarto de outra Vítima)

O ambiente final é o Círculo da Vítima, uma sala reclusa que não pertence a Eddie, mas sim a uma vítima anterior sequestrada que não conseguiu escapar. A sala, embora semelhante a um quarto, está saturada de símbolos, rabiscos e equações incompletas deixadas nas paredes, representando o legado de desespero e a confusão mental da pessoa aprisionada. Este cenário é o ápice do mistério e do puzzle. É aqui que o jogador resolve a equação final, "Encontre o X e fuja", descobrindo que o código numérico obtido é, na verdade, uma tragédia recorrente.



Arquitetura e Código:

Lógica Principal:

O jogo usa condicionais (if/else) para controlar ações do jogador.

Por exemplo:

if (keyboard_check(vk_right)) faz o personagem andar para a direita.

if (!place_meeting(xnovo, y, Obj_parede)) impede que ele atravesse paredes.

Também usamos

if (keyboard_check_pressed(ord("E"))) para o jogador abrir uma folha (atividades).

Essas condições verificam o que o jogador faz e como o jogo deve reagir.

Laços (repetições):

Usamos while para mover o personagem até encostar em uma parede sem ultrapassar:

```
while (!place_meeting(x + sign(hveloc), y, Obj_parede)) x += sign(hveloc);
```

Esse trecho ajusta a posição do personagem suavemente, sem atravessar o cenário.

Funções e Parâmetros:

Criamos funções internas com o GameMaker, como point_distance(x, y, folha.x, folha.y)

para medir a distância entre o personagem e objetos (como as folhas). Ela é usada, por exemplo, para decidir se o jogador está perto o suficiente para interagir.

Trecho do código importante:

```
if (keyboard_check_pressed(ord("E")) && !global.folha_aberta) {
```

```
var folha = instance_nearest(x, y, obj_folha);
```

```
if (instance_exists(folha)) {
```

```
var dist = point_distance(x, y, folha.x, folha.y);
```

```
if (dist < 80) {
```

```
instance_create_layer(display_get_gui_width()/2,
```

```
display_get_gui_height()/2,
```

```
"GUI".
```

```
obj_folha_aberta);
```

```
global.folha_aberta = true;
```

O que ele faz:

Quando o jogador aperta E perto de uma folha, o jogo abre uma tela de exercícios matemáticos. Essa é uma das partes principais do jogo, pois conecta a exploração com os desafios de matemática.

Resumo:

if/else controla ações e colisões.

while faz não atravessar paredes.

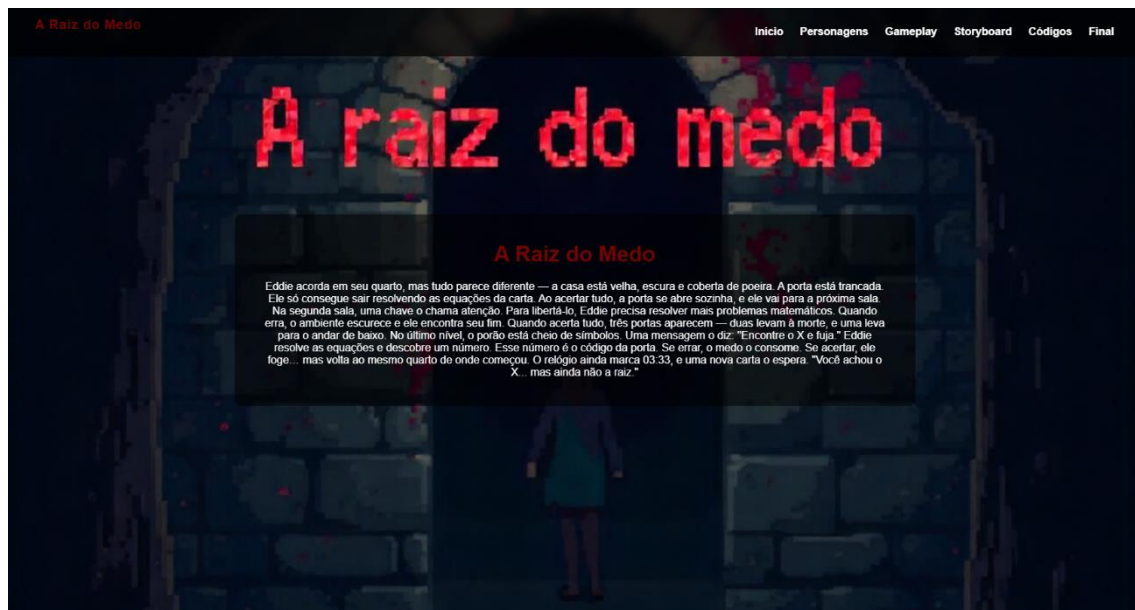
Funções calculam distância e criam objetos.

O código acima é o "cérebro" da interação do jogador com os desafios do jogo.

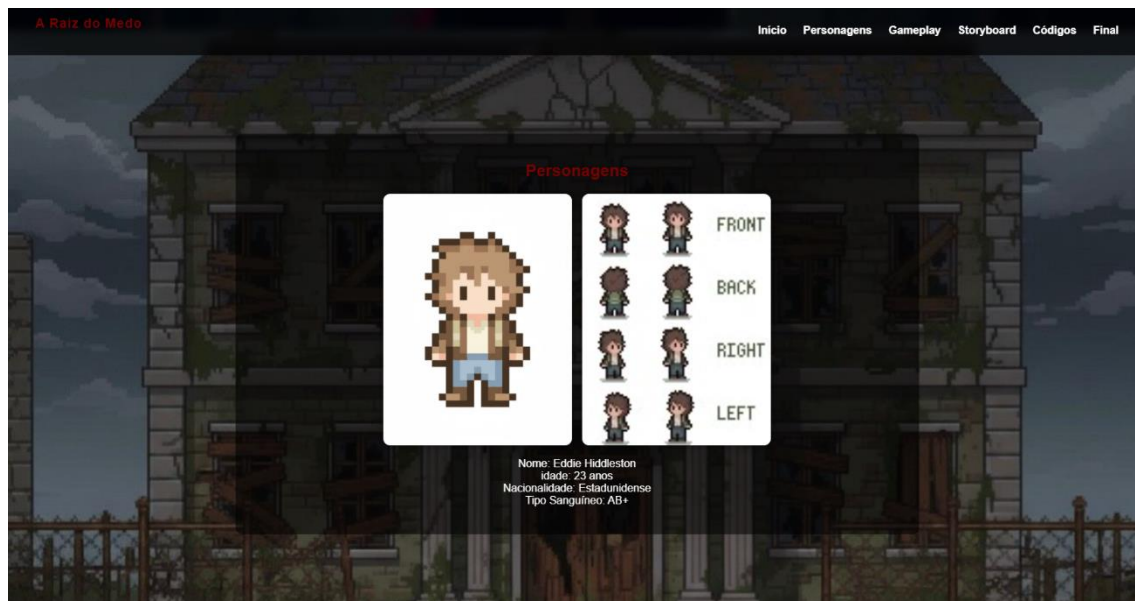
O Site de Divulgação:

O site foi criado com o intuito de divulgar e apresentar algumas informações básicas do jogo como a sinopse e ao nosso protagonista, além de facilitar o acesso para o jogo, deixando mais fácil do jogador poder jogar o jogo em paz.

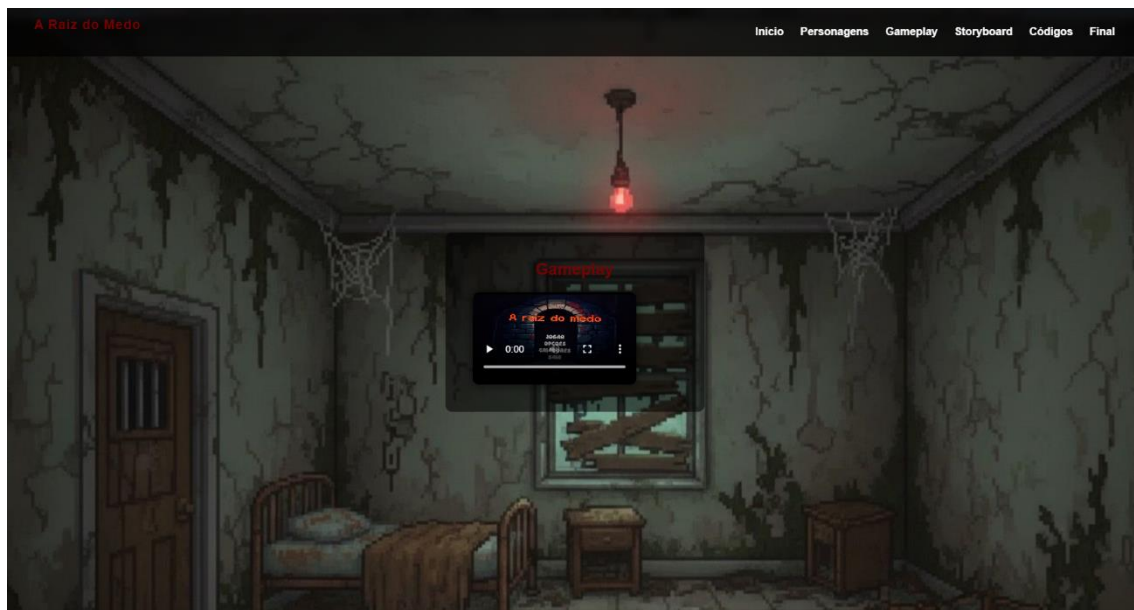
Link do site: <file:///C:/Users/Junior/Documents/Jogo%20site/Site%20jogo%201.html>



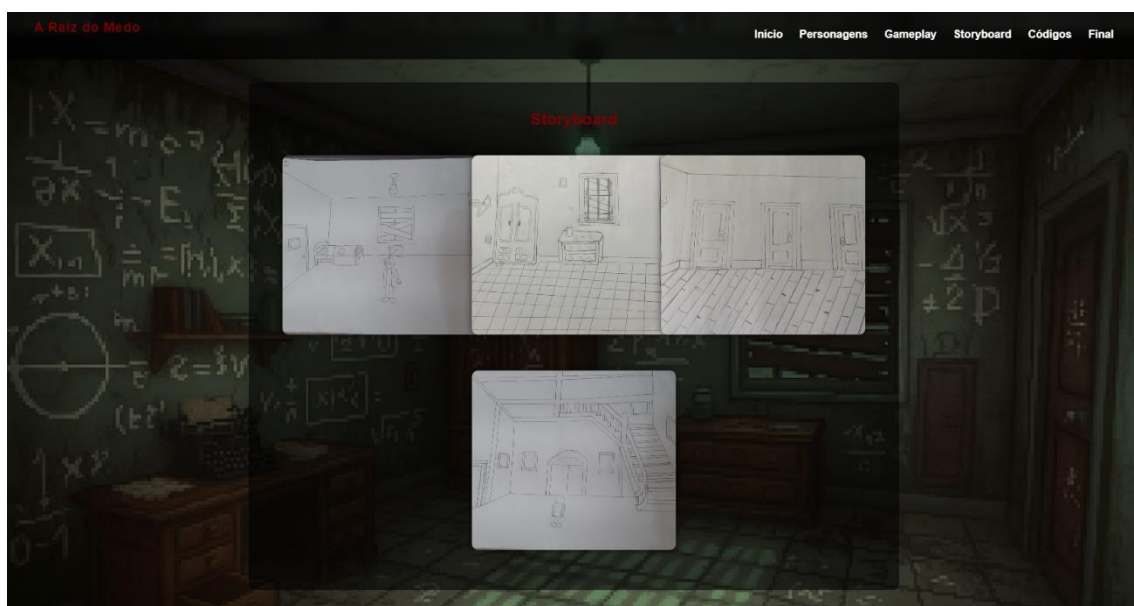
O início tem a função de introduzir o jogo ao jogador, apresentando a sinopse do jogo e chamar a atenção do leitor.



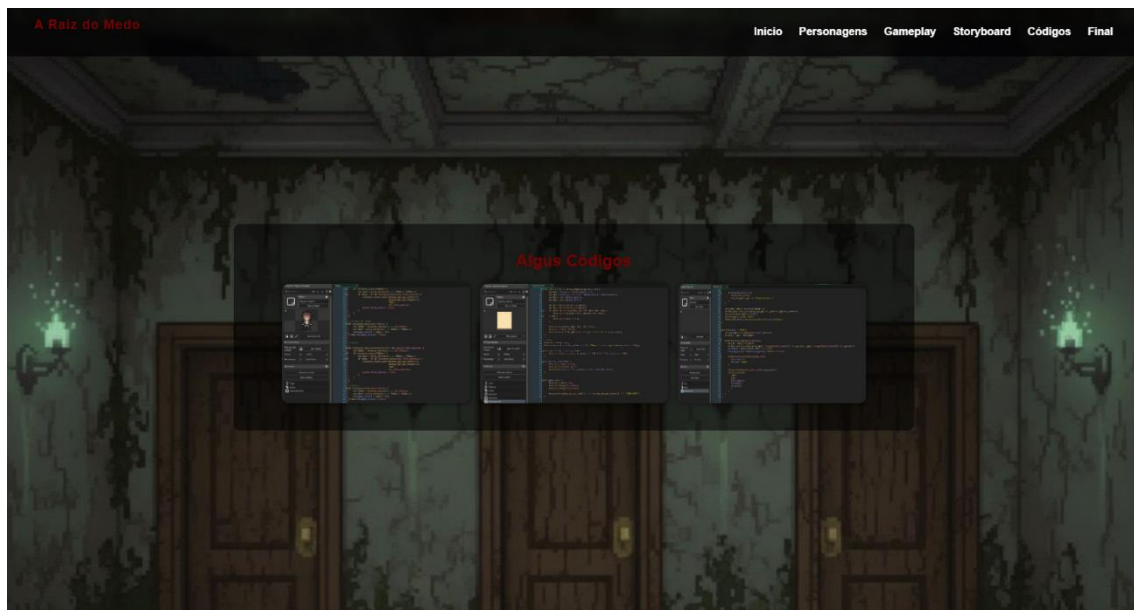
A parte de personagens tem o objetivo de apresentar o nosso protagonista, e dizer algumas informações sobre ele.



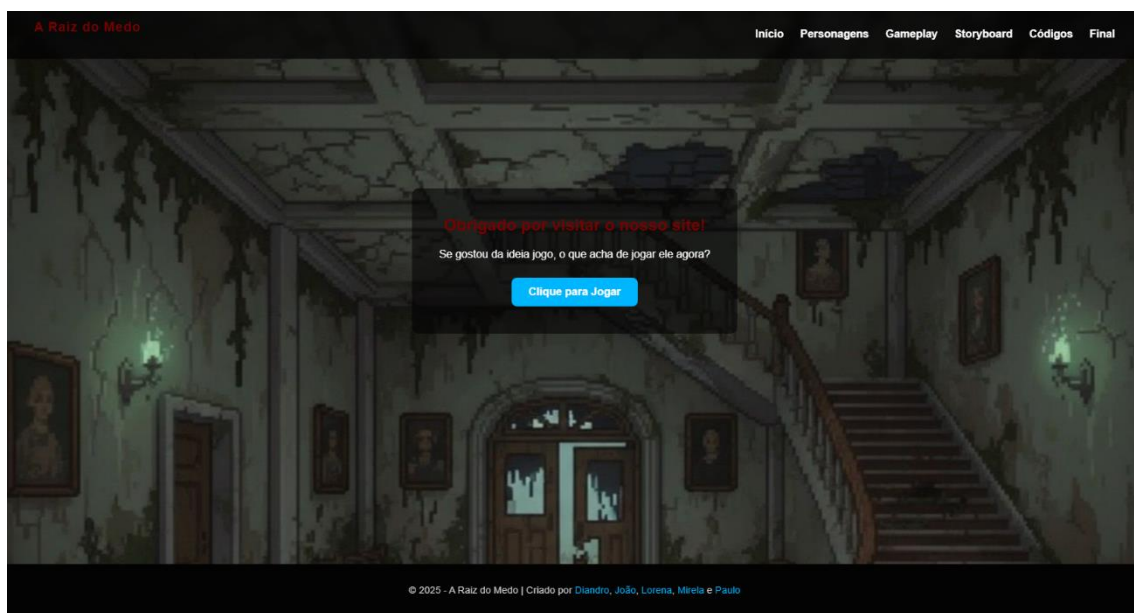
Essa parte tem o intuito de mostrar toda a gameplay do jogo, para o jogador ter alguma ideia de como jogar e não se perder durante a jogatina.



A parte dos Storyboards tem a função de mostrar toda a ideia dos cenários de antes da criação do jogo.



O intuito dessa parte é mostrar um pouco dos códigos de alguns sprites do jogo, para dar uma ideia ao jogador do processo de criação do jogo, e mostrar a complexidade dele.



E o final tem o objetivo de facilitar o acesso do jogador ao jogo, facilitando muito o seu trabalho em baixar o jogo, é só clicar no botão e ele já está pronto para jogar.

O site teve como principais características de design, as div's para centralizar o conteúdo do site, o cabeçalho para facilitar o usuário a mexer no site, e principalmente o efeito Parallax, que serve para embelezar o site e dar uma sensação de imersão.

Protótipo e Teste

A gente começou os testes do jogo pelo menu, e essa parte foi bem tranquila. Os códigos funcionaram de primeira e os bugs que apareceram eram coisinhas simples, mais ligadas a ajustar o que ficava em cada lugar ou a configuração das opções. Foi fácil de resolver e deu para seguir em frente sem grandes problemas nessa questão.

Porem quando fomos programar o jogo em si foi mais complicado e começamos a encontrar alguns bugs. Como não tínhamos muita prática com algumas partes do código, vários bugs começaram a surgir. Um em particular deu um trabalho enorme: era um erro que fazia uma folha no cenário se duplicar na tela sem parar. A gente perdeu algumas horas tentando entender o que estava causando aquilo e, principalmente, como consertar de vez essa falha chata.

Depois de resolvermos o erro no código que causava o motivo dessa duplicação, o resto dos testes ficou mais leve. Encontramos outros bugs, claro, mas nada que se comparasse ao tempo e ao estresse que passamos com a folha. Nenhum outro problema foi tão grande ou demorou tanto para ser resolvido, o que nos permitiu seguir com o desenvolvimento com mais tranquilidade.



Conclusão

O desenvolvimento deste jogo foi uma experiência fantástica que nos ajudou a crescer muito. Conseguimos melhorar nossas habilidades de programação, aprendendo a escrever códigos mais limpos e, o que é mais importante, a montar sistemas complexos dentro do jogo. Além disso, tivemos que forçar a barra para ter e organizar ideias para o game, o que foi um ótimo exercício de criatividade. E, claro, aprendemos na prática a importância de trabalhar bem em equipe e de manter a comunicação em dia.

No início, nossos planos eram enormes, criativos até demais. A gente queria colocar no jogo um monte de coisas: muitas animações, quebra-cabeças bem difíceis e uma variedade grande de monstros para o jogador enfrentar. No entanto, percebemos que o tempo que tínhamos era apertado e que o nosso nível de experiência não seria suficiente para entregar tudo isso com qualidade. Em vez de ficarmos frustrados, fizemos o que era preciso: pegamos todas aquelas ideias incríveis e adaptamos elas, criando versões mais simples e viáveis que cabiam no nosso prazo. Isso nos ensinou muito sobre a realidade de um projeto.

A maior dificuldade que encontramos no caminho foi a falta de comunicação clara entre a gente. Como não tínhamos um ritmo constante de conversas e notícias sobre o andamento das tarefas, isso fez com que uma parte importante do jogo atrasasse bastante. Esse desliz acabou nos afetando um pouco, mas serviu como uma grande lição. Depois de notarmos esse erro, passamos a conversar muito mais e, felizmente, tudo se resolveu. A parte em que mais nos divertimos em todo esse processo foi, sem dúvida, programar o jogo em si. Foi uma experiência inédita e muito valiosa. Tudo o que aprendemos sobre códigos e a lógica de games vai ser extremamente útil nos nossos próximos trabalhos e projetos. Se tivéssemos um mês a mais, usaríamos esse tempo para dar um polimento geral, melhorando as telas (o menu, o pause) e adicionando mais conteúdo: novos jeitos de jogar, quebra-cabeças mais elaborados, mais monstros, mais personagens e uma história bem mais desenvolvida.

Referências

Pause System:

NILK. Como fazer sistema de PAUSE (Game Maker Studio 2). YouTube, 29 dez. 2023. 1 vídeo (9min48s). Disponível em: https://www.youtube.com/watch?v=b6_YlQZ674w. Acesso em: 4 nov. 2025.

Game-Maker Tutorial:

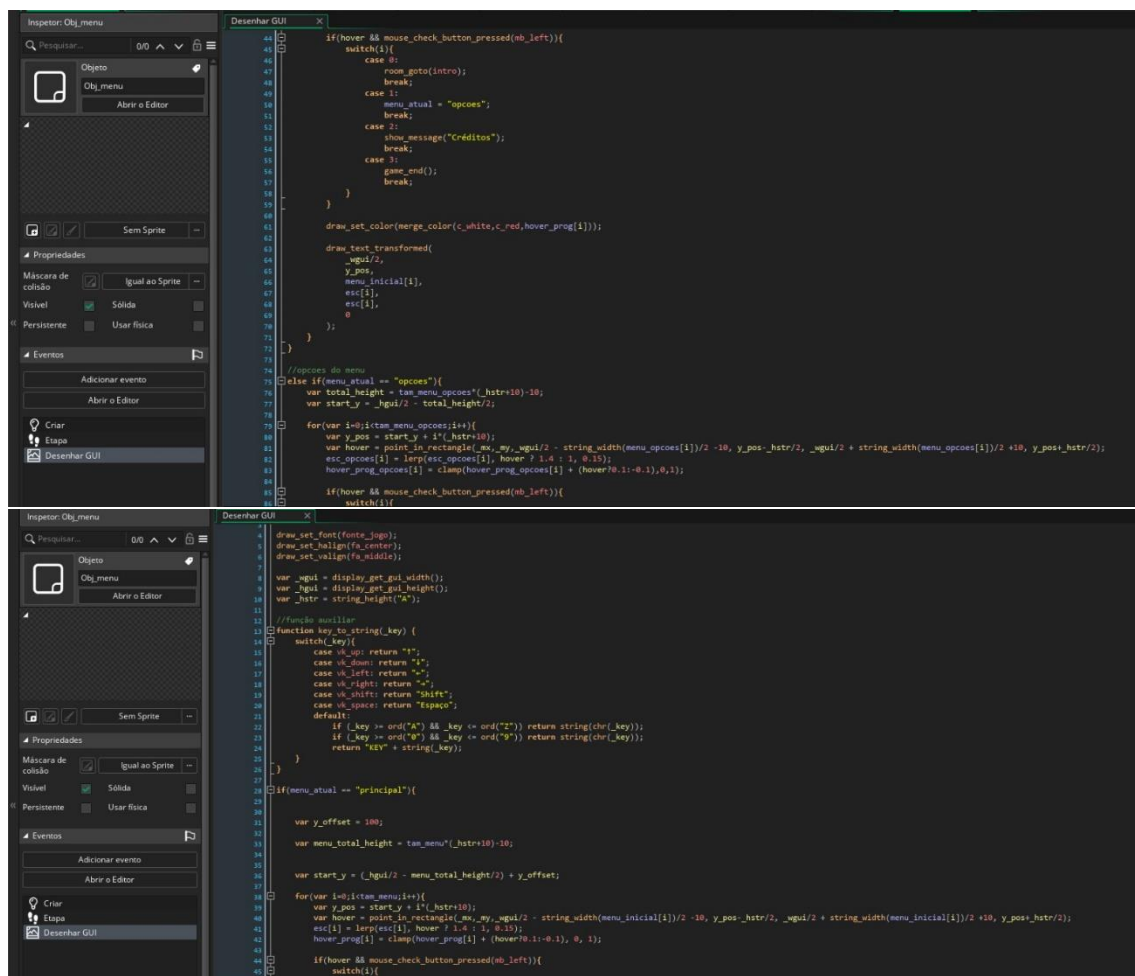
NILK. Como usar o GAME-MAKER - Tutorial Básico. YouTube, 20 abr. 2023. 1 vídeo (10min37s). Disponível em: <https://www.youtube.com/watch?v=ypo83R7slg0>. Acesso em: 26 out. 2025.

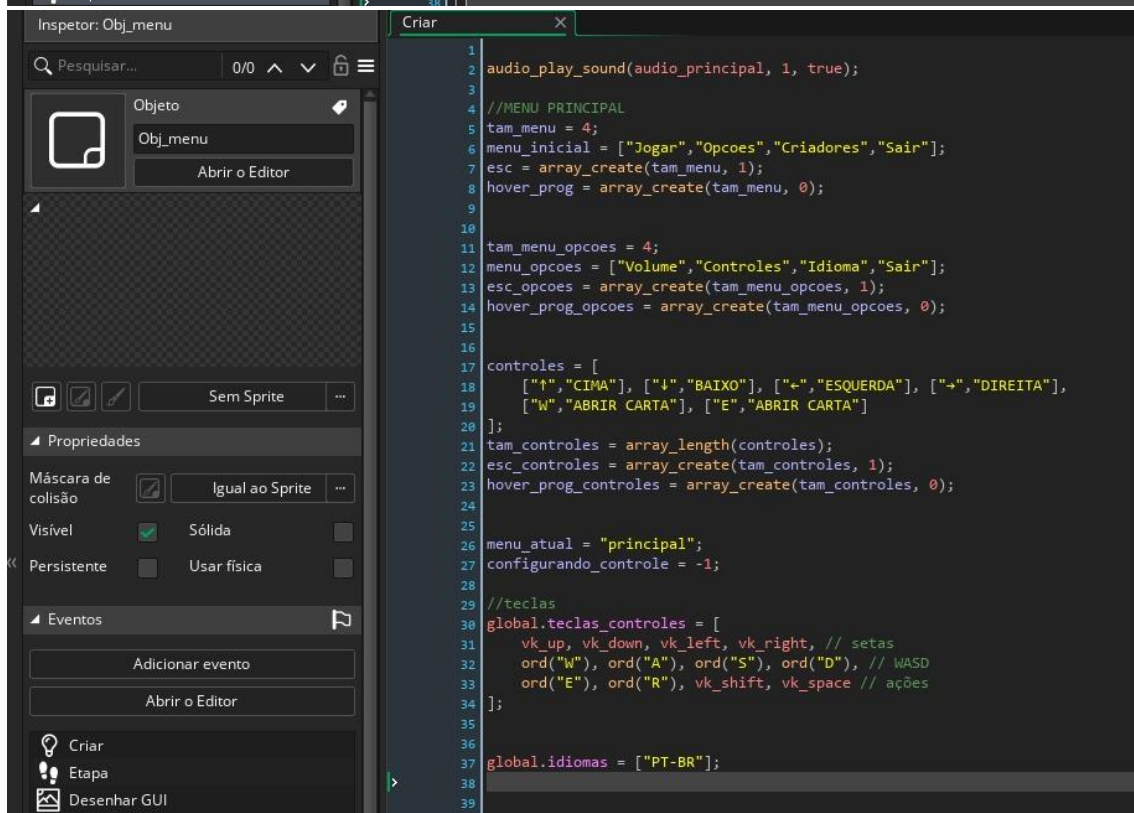
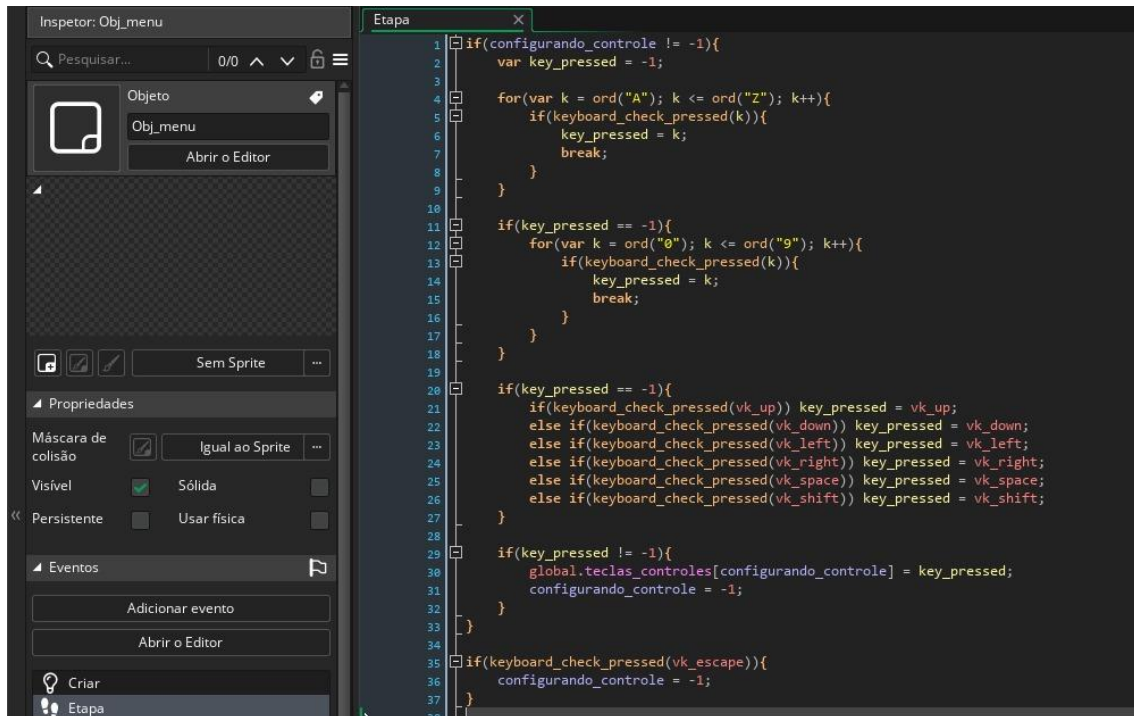
Menu Tutorial:

NILK. Como fazer MENU (Game Maker Studio 2). YouTube, 29 dez. 2023. 1 vídeo (9min21s). Disponível em: https://www.youtube.com/watch?v=ZaYA4ZDCb_Y. Acesso em: 31 out. 2025.

Apêndice

Códigos:






Inspetor: Obj_personagem

Objeto

Obj_personagem

Abrir o Editor



parado_frente

...

Propriedades

Máscara de colisão ☒ spr_colisao

Visível ☒ Sólida

Persistente ☐ Usar física

Eventos

Adicionar evento

Abrir o Editor

Criar

Etapa

Desenhar GUI

Etapa

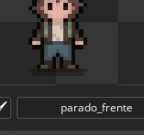
```
42 if (instance_exists(folha)) {
43     var dist = point_distance(x, y, folha.x, folha.y);
44     if (dist < 80 && !instance_exists(obj_folha_aberta)) {
45         instance_create_layer(display_get_gui_width()/2,
46                               display_get_gui_height()/2,
47                               "GUI",
48                               obj_folha_aberta);
49         global.folha_aberta = true;
50     }
51 }
52 }
53
54 // E para ler folha
55 if (instance_exists(obj_folha)) {
56     var folha = instance_nearest(x, y, obj_folha);
57     var dist = point_distance(x, y, folha.x, folha.y);
58     mensagem_visivel = (dist < 80);
59 } else mensagem_visivel = false;
60
61 //FOLHA2
62
63 if (keyboard_check_pressed(ord("W")) && !global.folha_aberta2) {
64     var folha = instance_nearest(x, y, obj_folha2);
65     if (instance_exists(folha)) {
66         var dist = point_distance(x, y, folha.x, folha.y);
67         if (dist < 80 && !instance_exists(obj_folha_aberta2)) {
68             instance_create_layer(display_get_gui_width()/2,
69                                   display_get_gui_height()/2,
70                                   "GUI",
71                                   obj_folha_aberta2);
72             global.folha_aberta2 = true;
73         }
74     }
75 }
76
77 //tecla W
78 if (instance_exists(obj_folha2)) {
79     var folha = instance_nearest(x, y, obj_folha2);
80     var dist = point_distance(x, y, folha.x, folha.y);
81     mensagem_visivel = (dist < 80);
82 } else mensagem_visivel = false;
83
84
```

Inspetor: Obj_personagem

Objeto

Obj_personagem

Abrir o Editor



parado_frente

...

Propriedades

Máscara de colisão ☒ spr_colisao

Visível ☒ Sólida

Persistente ☐ Usar física

Eventos

Adicionar evento

Abrir o Editor

Criar

Etapa

Desenhar GUI

Etapa

```
1 if (global.pause) exit;
2
3 var direita = keyboard_check(vk_right);
4 var esquerda = keyboard_check(vk_left);
5 var cima = keyboard_check(vk_up);
6 var baixo = keyboard_check(vk_down);
7
8 var hveloc = 0;
9 var vveloc = 0;
10
11 if (direita) { hveloc = velocidade; direc = 1; sprite_index = personagem_direita; }
12 else if (esquerda) { hveloc = -velocidade; direc = 2; sprite_index = personagem_esquerda; }
13 else if (cima) { vveloc = -velocidade; direc = 3; sprite_index = personagem_atras; }
14 else if (baixo) { vveloc = velocidade; direc = 4; sprite_index = personagem_frente; }
15 else {
16     if (direc == 1) sprite_index = parado_direita;
17     else if (direc == 2) sprite_index = parado_esquerda;
18     else if (direc == 3) sprite_index = parado_atras;
19     else if (direc == 4) sprite_index = parado_frente;
20 }
21
22 var xnovo = x + hveloc;
23 var ynovo = y + vveloc;
24
25 if (!place_meeting(xnovo, y, Obj_pared)) x = xnovo;
26 else { while(!place_meeting(x + sign(hveloc), y, Obj_pared)) x += sign(hveloc); hveloc = 0; }
27
28 if (!place_meeting(x, ynovo, Obj_pared)) y = ynovo;
29 else { while(!place_meeting(x, y + sign(vveloc), Obj_pared)) y += sign(vveloc); vveloc = 0; }
30
31
32 var movendo = (hveloc != 0) || (vveloc != 0);
33 if (movendo) {
34     if (!audio_is_playing(audio_passos)) audio_play_sound(audio_passos, 1, true);
35 } else {
36     audio_stop_sound(audio_passos);
37 }
38
39 //para abrir primeira folha
40 if (keyboard_check_pressed(ord("E")) && !global.folha_aberta) {
41     var folha = instance_nearest(x, y, obj_folha);
42     if (instance_exists(folha)) {
43         var dist = point_distance(x, y, folha.x, folha.y);
44     }
45 }
```

Inspetor: Obj_personagem


Pesquisar...




0/0

Objeto

Obj_personagem

Abrir o Editor






parado_frente

...

Propriedades

Máscara de colisão

spr_colisao

Visível☒ Sólida☐

Persistente☐ Usar física☐

Eventos

Adicionar evento

Abrir o Editor

Criar

Criar

```
1 //variáveis globais
2 if (!variable_global_exists("folha_aberta")) {
3     global.folha_aberta = false;
4 }
5
6 if (!variable_global_exists("folha_aberta2")) {
7     global.folha_aberta2 = false;
8 }
9
10
11 if (!variable_global_exists("folha_audio_tocada")) {
12     global.folha_audio_tocada = false;
13 }
14 direita = 0;
15 esquerda = 0;
16
17 cima = 0;
18 baixo = 0;
19 gravidade = 0.08;
20
21 direc = 4;
22
23 hveloc = 0; //velocidade horizontal
24 vveloc = 0; //velocidade vertical
25 velocidade = 1.2
26
27 vidas = 3;
28
```

Inspetor: obj_folha_aberta


Pesquisar...




0/0

Objeto

obj_folha_aberta

Abrir o Editor






folha_aberta

...

Propriedades

Máscara de colisão

Igual ao Sprite

Visível☒ Sólida☐

Persistente☐ Usar física☐

Eventos

Adicionar evento

Abrir o Editor

Criar

Destruir

Etapa

Alarme 0

Desenhar

Desenhar GUI

*Desenhar GUI

```
8 for (var i = 0; i < array_length(q.opcoes); i++) {
9     var bx1 = painel_x - botao_largura / 2;
10    var by1 = painel_y + 40 + (i * (botao_altura + botao_espaco));
11    var bx2 = bx1 + botao_largura;
12    var by2 = by1 + botao_altura;
13
14    var mx = device_mouse_x_to_gui(0);
15    var my = device_mouse_y_to_gui(0);
16    if (point_in_rectangle(mx, my, bx1, by1, bx2, by2)) {
17        draw_set_color(make_color_rgb(200, 255, 200));
18    } else {
19        draw_set_color(c_white);
20    }
21
22    draw_rectangle(bx1, by1, bx2, by2, false);
23    draw_set_color(c_black);
24    draw_text(bx1 + 70, by1 + 11, string(i + 1) + " " + q.opcoes[i]);
25 }
26
27 //tempo
28 draw_set_color(c_red);
29 draw_text(painel_x + 60, painel_y - 120, "Tempo: " + string(round(tempo_atual / 60)));
30
31 for (var i = 0; i < vidas; i++) {
32     draw_sprite(spr_coracao, 0, painel_x - 100 + (i * 25), painel_y - 140);
33 }
34
35 if (mostrar_resultado) {
36     draw_set_color(c_white);
37     draw_set_font(font_intro);
38     draw_text(painel_x + 10, painel_y + 220, resultado_texto);
39 }
40
41
42 if (game_over) {
43     draw_set_color(c_red);
44     draw_set_halign(fa_center);
45     draw_set_valign(fa_middle);
46
47     draw_text(display_get_gui_width() / 2, display_get_gui_height() / 2, "GAME OVER");
48 }
```


Inspetor: obj_folha_aberta

Pesquisar...

0/0

^

v


🔒

☰

Objeto

obj_folha_aberta

Abrir o Editor



folha_aberta

...

Propriedades

Máscara de colisão

☒ Igual ao Sprite

...

Visível

☒ Sólida

Persistente

☐ Usar física

Eventos

Adicionar evento

Abrir o Editor

Criar

Destruir

Etapa

Alarme 0

Desenhar

Desenhar GUI

*Desenhar GUI

```
1 draw_sprite_ext(sprite_index, 0, painel_x, painel_y, 1, 1, 0, c_white, 1);
2
3 var q = questoes[questao_atual];
4 draw_set_color(c_black);
5 draw_text(painel_x - 10, painel_y - 40, string(questao_atual + 1) + ") " + q.pergunta);
6
7 //botões
8 for (var i = 0; i < array_length(q.opcoes); i++) {
9     var bx1 = painel_x - botao_largura / 2;
10    var by1 = painel_y + 40 + (i * (botao_altura + botao_espaco));
11    var bx2 = bx1 + botao_largura;
12    var by2 = by1 + botao_altura;
13
14    var mx = device_mouse_x_to_gui(0);
15    var my = device_mouse_y_to_gui(0);
16    if (point_in_rectangle(mx, my, bx1, by1, bx2, by2)) {
17        draw_set_color(make_color_rgb(200, 255, 200));
18    } else {
19        draw_set_color(c_white);
20    }
21
22    draw_rectangle(bx1, by1, bx2, by2, false);
23    draw_set_color(c_black);
24    draw_text(bx1 + 70, by1 + 11, string(i + 1) + ") " + q.opcoes[i]);
25
26 //tempo
27 draw_set_color(c_red);
28 draw_text(painel_x + 60, painel_y - 120, "Tempo: " + string(round(tempo_atual / 60)));
29
30 for (var i = 0; i < vidas; i++) {
31     draw_sprite(spr_coracao, 0, painel_x - 100 + (i * 25), painel_y - 140);
32 }
33
34 if (mostrar_resultado) {
35     draw_set_color(c_white);
36     draw_set_font(font_intro);
37     draw_text(painel_x + 10, painel_y + 220, resultado_texto);
38 }
39
40 if (game_over) {
41     draw_set_color(c_red);
42 }
43 }
```

Inspetor: obj_folha_aberta

Pesquisar...

0/0

^

v


🔒

☰

Objeto

obj_folha_aberta

Abrir o Editor



folha_aberta

...

Propriedades

Máscara de colisão

☒ Igual ao Sprite

...

Visível

☒ Sólida

Persistente

☐ Usar física

Eventos

Adicionar evento

Abrir o Editor

Criar

Destruir

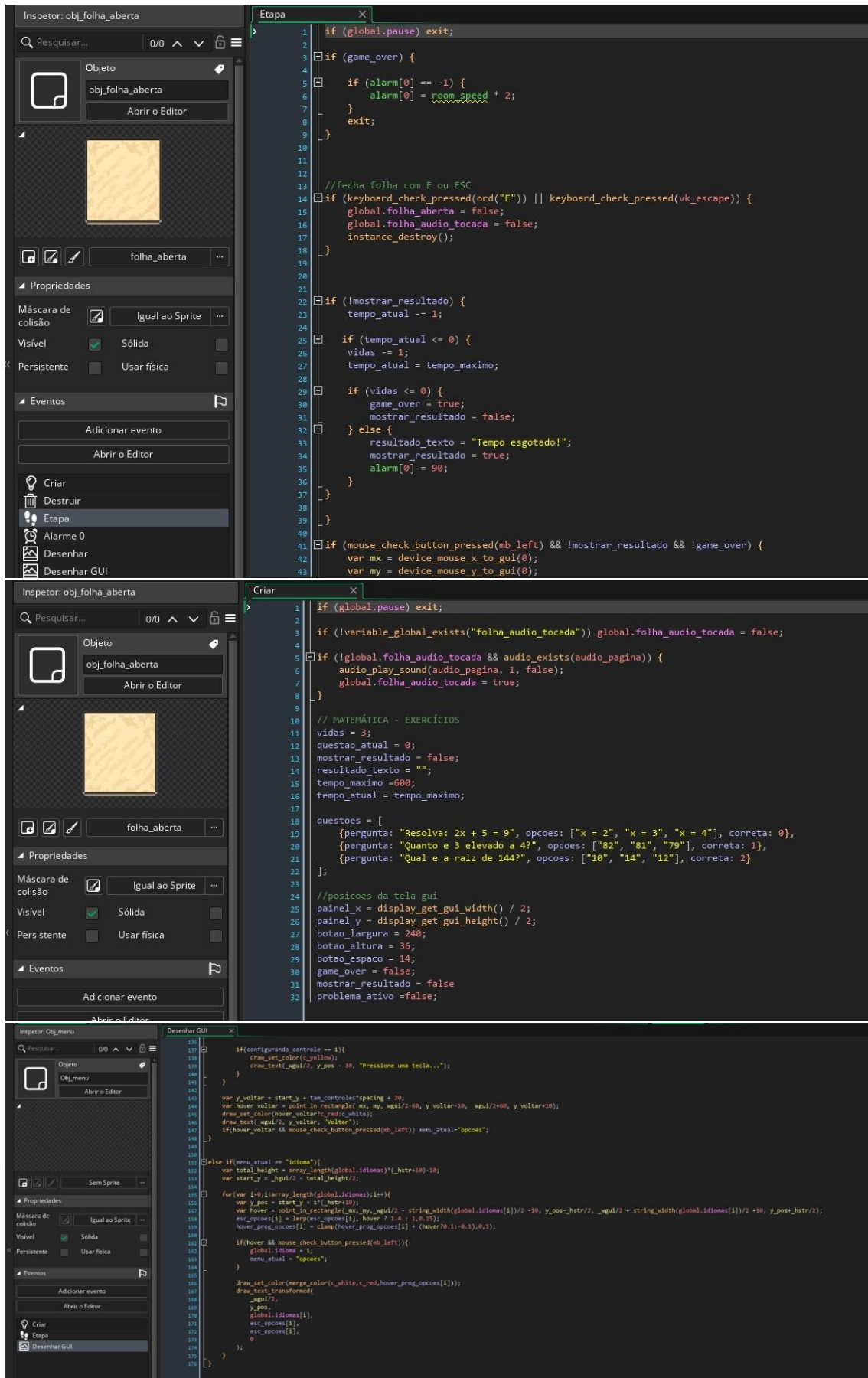
Etapa

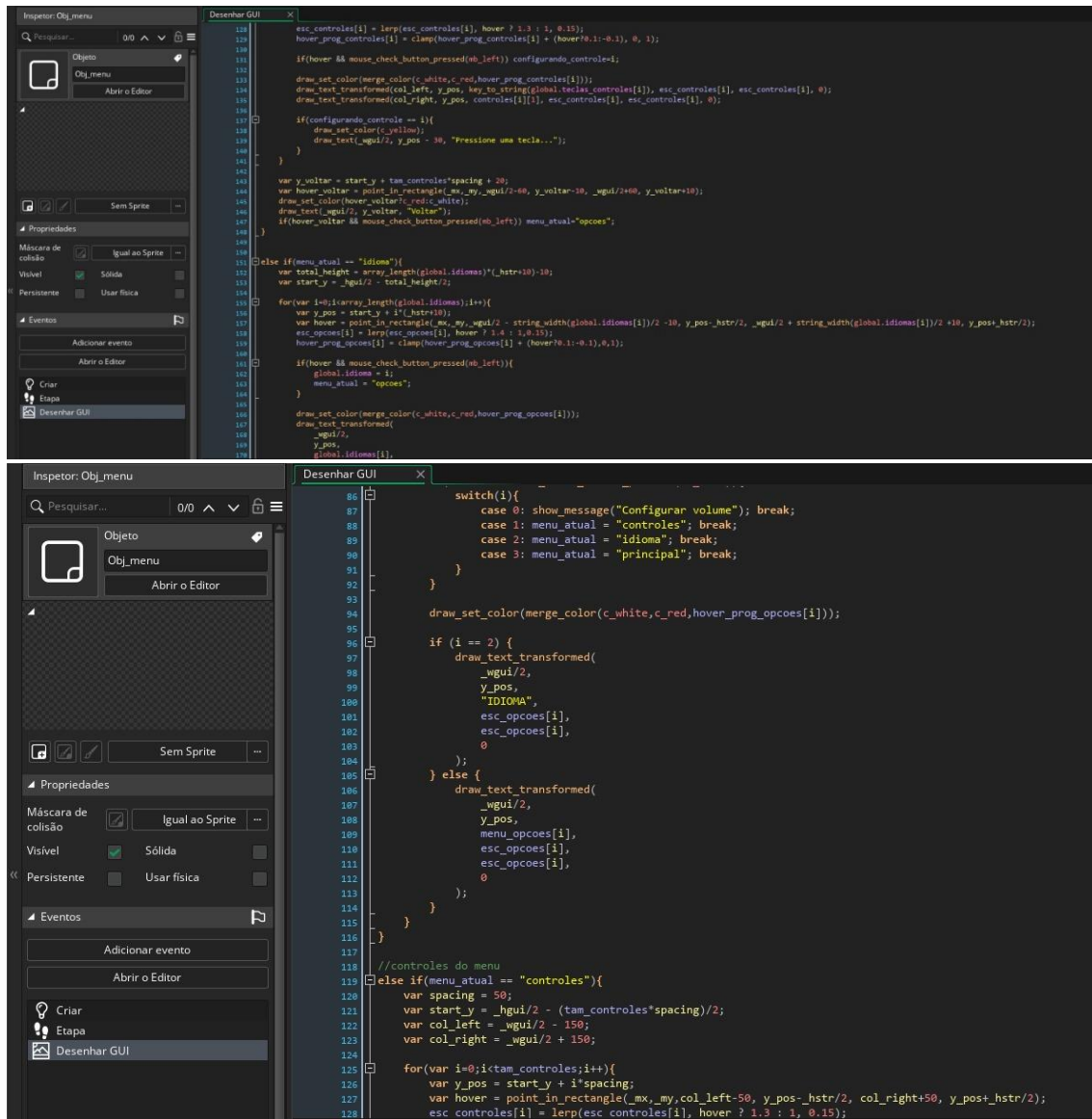
Alarme 0

Desenhar

Etapa

```
38 }
39
40 if (mouse_check_button_pressed(mb_left) && !mostrar_resultado && !game_over) {
41     var mx = device_mouse_x_to_gui(0);
42     var my = device_mouse_y_to_gui(0);
43     var q = questoes[questao_atual];
44
45     for (var i = 0; i < array_length(questoes); i++) {
46         var bx1 = painel_x - botao_largura / 2;
47         var by1 = painel_y + 40 + (i * (botao_altura + botao_espaco));
48         var bx2 = bx1 + botao_largura;
49         var by2 = by1 + botao_altura;
50
51         if (point_in_rectangle(mx, my, bx1, by1, bx2, by2)) {
52             if (i == q.correta) {
53                 resultado_texto = "Correto!";
54                 questao_atual += 1;
55
56                 if (questao_atual >= array_length(questoes)) {
57                     resultado_texto = "Todas respondidas!";
58                     room_goto(segundo_nivel);
59                     questao_atual = 0;
60                 }
61             } else {
62                 resultado_texto = "Errado!";
63                 vidas -= 1;
64                 if (vidas <= 0) {
65                     game_over = true;
66                     mostrar_resultado = false;
67                     room_goto(menu);
68                 }
69             }
70         }
71     }
72     mostrar_resultado = true;
73     alarm[0] = 90;
74     tempo_atual = tempo_maximo;
75
76 }
77
78 }
```





Storyboard:

