

Trabalho prático Agentes Inteligentes

Objetivo do Projeto

O nosso objetivo foi aplicar conceitos aprendidos na disciplina de Sistemas Inteligentes, incluindo Máquina de Estado Finito e Árvore de Decisão.

Componentes do Jogo

Personagem Jogável(Player):

O jogador controla um personagem jogável com movimentos básicos.

Possui os modos patrulhar, atirar e super ataque.

As teclas de comando são:

keyboard right - Movimento para o lado direito;

keyboard left - Movimento para o lado esquerdo;

keyboard up - Movimento para o cima;

keyboard down - Movimento para baixo;

Mouse Button left - Ataque comum;

Mouse Button right - Super Ataque;

Personagem NPC 1(Agente RED):

Personagem movimento controlados pela IA.

Possui os modos Patrulhar, Perseguir, Criar Obstáculo, Atacar e Game Over.

Personagem NPC 2(Agente YELLOW):

Personagem movimento controlados pela IA.

Possui os modos Patrulhar, Criar Obstáculo e Atacar.

Elementos do Jogo

Player:

Blue Powerball - Ataque Comum

Superball Blue2 - Super Ataque

Agente Red:

Red Powerball - Ataque comum

Superball Red2 - Objeto Explosivo(Formato Caveira)

Agente Yellow:

Superball Yellow - Objeto Super Ataque(Formato de RAI0)

Superball Yellow 2 - Objeto Vida(Formato de CORAÇÃO)

Jogabilidade, objetivo e lógica básica do jogo.

O objetivo principal do jogo é derrotar o Agente Red, fugindo e esquivando dos ataques e dos objetos explosivos deixados pelo Agente Red.

O segundo objetivo é coletar os itens deixados pelo Agente Yellow, para utilizá-los durante o objetivo 1.

Player:

O player possui uma jogabilidade simples, sendo controlado pelo teclado e mouse. O Player é controlado pelas teclas de seta, e utiliza os botões do mouse para disparar ataques comuns e super ataques.

O Player já iniciou o Game com 5 unidades de vida, 100 unidades de Munição simples e 1 unidade de Super Ataque.

Cada ataque simples do Player elimina 1 unidade de vida do Agente RED, e em cada ataque simples do Agente RED é perdido uma unidade de Vida do Player.

Cada objeto CORAÇÃO que o Player coleta, acrescenta 1 unidade de vida para o Player e cada objeto RAI0 acrescenta 1 unidade de Super Ataque para o Player.

E o Super Ataque do Player tira 5 unidades de Vida do Agente Red.

Agente RED:

O Agente RED possui uma jogabilidade controlada pela IA.

O Agente Red inicia o Game com 5 unidades de vida, com Munição infinita e cria objetos explosivos de forma infinita.

O Agente RED possui ataques simples que tiram 1 unidade de vida do Player, e cria diversos objetos explosivos em formato de caveira durante o jogo, sempre que entrar em contato com o player destrói automaticamente o Player e reseta o GAME. Além de possuir o modo Game Over, quando o Player toca o Agente RED, o agente é destruído automaticamente, e reiniciar o GAME.

Agente YELLOW:

O Agente Yellow possui uma jogabilidade controlada pela IA.

O Agente Yellow não sofre danos e não pode ser destruído.

O Agente Yellow não possui ataques ofensivos, ele age durante o Game como um ajudante do Player.

O Agente Yellow cria objetos para serem coletados pelo Player auxiliando o Player no objetivo de destruir o Agente Red.

Os objetos criados pelo Agente Yellow são:

O coração, que acrescenta 1 unidade de vida para o Player, e o RAIO que acrescenta 1 unidade de Super Ataque para o Player.

Códigos, Eventos, Scripts e Nós

Player:

O GAME possui os seguintes eventos e nós para o Player.

Patrulhar - Possui os comandos básicos para movimentar o Player e os códigos de ataques.

```
// -----Codigo que gera os tiros -----
    if municao_player >= 10{
        if (mouse_check_button(mb_left)) && (cooldown < 1){
            municao_player -= 10;//controla a municao
            alarm[0] = 180;
            instance_create_layer(x, y, "Instances", obj_blue_powerball);
            cooldown = 10;
        }
        // Subtrai 1 da variável "cooldown" de cada quadro do jogo,
        cooldown = cooldown - 1;
    }
    // Controla a municao
    if alarm[0] <= 0{
```

```

        municao_player += 1;
    }
    municao_player = clamp(municao_player, 0, max_municao_player);

    // -----Codigo que gera as SUPERBOLLS -----
    if superball >= 1{
        if (mouse_check_button(mb_right)) && (cooldown < 1){
            superball -= 1;//controla a municao

            instance_create_layer(x, y, "Instances", obj_superball_blue2);
            cooldown = 10;
        }
        // Subtrai 1 da variável "cooldown" de cada quadro do jogo,
        cooldown = cooldown - 1;
    }

```

Perseguir - Possui os comandos básicos para movimentar o Player e os códigos de ataques.

```

// -----Codigo que gera os tiros -----
if municao_player >= 10{
    if (mouse_check_button(mb_left)) && (cooldown < 1){
        municao_player -= 10;//controla a municao
        alarm[0] = 180;
        instance_create_layer(x, y, "Instances", obj_blue_powerball);
        cooldown = 10;
    }
    // Subtrai 1 da variável "cooldown" de cada quadro do jogo,
    cooldown = cooldown - 1;
}
// Controla a municao
if alarm[0] <= 0{
    municao_player += 1;
}
municao_player = clamp(municao_player, 0, max_municao_player);

// -----Codigo que gera as SUPERBOLLS -----
if superball >= 1{
    if (mouse_check_button(mb_right)) && (cooldown < 1){
        superball -= 1;//controla a municao

        instance_create_layer(x, y, "Instances", obj_superball_blue2);
        cooldown = 10;
    }
    // Subtrai 1 da variável "cooldown" de cada quadro do jogo,
    cooldown = cooldown - 1;
}

```

Criar Obstáculo - Possui os comandos básicos para movimentar o Player e os códigos de ataques.

```
// -----Codigo que gera os tiros -----
if municao_player >= 10{
    if (mouse_check_button(mb_left)) && (cooldown < 1){
        municao_player -= 10;//controla a municao
        alarm[0] = 180;
        instance_create_layer(x, y, "Instances", obj_blue_powerball);
        cooldown = 10;
    }
    // Subtrai 1 da variável "cooldown" de cada quadro do jogo,
    cooldown = cooldown - 1;
}
// Controla a municao
if alarm[0] <= 0{
    municao_player += 1;
}
municao_player = clamp(municao_player, 0, max_municao_player);

// -----Codigo que gera as SUPERBOLLS -----
if superball >= 1{
    if (mouse_check_button(mb_right)) && (cooldown < 1){
        superball -= 1;//controla a municao

        instance_create_layer(x, y, "Instances", obj_superball_blue2);
        cooldown = 10;
    }
    // Subtrai 1 da variável "cooldown" de cada quadro do jogo,
    cooldown = cooldown - 1;
}
```

Atacar - Possui os comandos básicos para movimentar o Player e os códigos de ataques.

```
// -----Codigo que gera os tiros -----
if municao_player >= 10{
    if (mouse_check_button(mb_left)) && (cooldown < 1){
        municao_player -= 10;//controla a municao
        alarm[0] = 180;
        instance_create_layer(x, y, "Instances", obj_blue_powerball);
        cooldown = 10;
    }
    // Subtrai 1 da variável "cooldown" de cada quadro do jogo,
    cooldown = cooldown - 1;
}
// Controla a municao
if alarm[0] <= 0{
    municao_player += 1;
}
municao_player = clamp(municao_player, 0, max_municao_player);
```

```
// -----Codigo que gera as SUPERBOLLS -----
    if superball >= 1{
        if (mouse_check_button(mb_right)) && (cooldown < 1){
            superball -= 1;//controla a municao

            instance_create_layer(x, y, "Instances", obj_superball_blue2);
            cooldown = 10;
        }
        // Subtrai 1 da variável "cooldown" de cada quadro do jogo,
        cooldown = cooldown - 1;
    }
}
```

Game Over - Possui os comandos básicos para movimentar o Player e os códigos de ataques.

```
// -----Codigo que gera os tiros -----
    if municao_player >= 10{
        if (mouse_check_button(mb_left)) && (cooldown < 1){
            municao_player -= 10;//controla a municao
            alarm[0] = 180;
            instance_create_layer(x, y, "Instances", obj_blue_powerball);
            cooldown = 10;
        }
        // Subtrai 1 da variável "cooldown" de cada quadro do jogo,
        cooldown = cooldown - 1;
    }
    // Controla a municao
    if alarm[0] <= 0{
        municao_player += 1;
    }
    municao_player = clamp(municao_player, 0, max_municao_player);

    // -----Codigo que gera as SUPERBOLLS -----
    if superball >= 1{
        if (mouse_check_button(mb_right)) && (cooldown < 1){
            superball -= 1;//controla a municao

            instance_create_layer(x, y, "Instances", obj_superball_blue2);
            cooldown = 10;
        }
        // Subtrai 1 da variável "cooldown" de cada quadro do jogo,
        cooldown = cooldown - 1;
    }
}
```

Agente RED:

O GAME possui os seguintes eventos e nós para o Agente Red.

Patrulhar - Possui os códigos necessários para gerar a ação de Patrulhar do Agente.

velocidade =2;

```
var dist = point_distance(x, y, destinox, destinoy);

if dist >= 10

    move_towards_point(destinox, destinoy, velocidade)
else{
    randomize()
    destinox  = random_range(64, room_width-64);
    destinoy  = random_range(64, room_height-64);
}
escrever_estado = "PATRULHANDO";
```

Perseguir - Possui os códigos necessários para gerar a ação de Perseguir do Agente.

// Variaveis para Perseguiçao

veloc_perseg = 2;

dist_perseg = 300;

```
//Calcular a distância entre o Agente e o Jogador
if distance_to_object(obj_player) <= dist_perseg{
// Vai na direção do objeto (jogador) na posição x e y, na velocidade 2
    move_towards_point( obj_player.x, obj_player.y, veloc_perseg)
    escrever_estado = "PERSEGUINDO";
}
```

Criar Obstáculo - Possui os códigos necessários para gerar a ação de Criar Obstáculo do Agente.

// Variaveis para Perseguiçao

veloc_perseg = 2;

dist_perseg = 300;

```
//Calcular a distância entre o Agente e o Jogador
if distance_to_object(obj_player) <= dist_perseg{
    if (frequencia > 0) {
        frequencia -= 1;
```

```

    } else {
        // Crie o objeto
        instance_create_layer(x, y, "Instances", obj_superball_red2);
        frequencia = 360; // Configurado novamente para 60 passos (1 segundo a 60 FPS)
    }
    escrever_estado = "CRIA OBSTÁCULO";
}

```

Atacar - Possui os códigos necessários para gerar a ação de Atacar do Agente.

```

veloc_atk = 2;
dist_atk = 100;
//cooldown = 0; //variavel colocada dentro do crear do obj

    if distance_to_object(obj_player) <= dist_atk{
        if cooldown < 1 {
            move_towards_point( obj_player.x, obj_player.y, veloc_atk); // Move o
obj ate o player
            instance_create_layer(x, y, "Instances", obj_red_powerball); //Cria o
powerboll
            cooldown = 50; //Tempo do contador da municao, quanto maior valor,
maior o tempo entre o disparo
        }
        cooldown = cooldown - 1; // O contador recebe o tempo -1
        escrever_estado = "ATACANDO";
    }

```

Game Over - Possui os códigos necessários para gerar a ação de GAME OVER do Agente.

```

// Variaveis para Perseguiacao
veloc_perseg = 2;
dist_perseg = 0;

    //Calcular a distância entre o Agente e o Jogador
    if distance_to_object(obj_player) <= dist_perseg {
        game_restart()

        escrever_estado = "GAMEOVER";
    }

```

Agente YELLOW

O GAME possui os seguintes eventos e nós para o Agente Yellow

Patrulhar - Possui os códigos necessários para gerar a ação de Patrulhar do Agente.

```
velocidade =2;

var dist = point_distance(x, y, destinox, destinoy);

if dist >= 10

    move_towards_point(destinox, destinoy, velocidade)
else{
    randomize()
    destinox  = random_range(64, room_width-64);
    destinoy  = random_range(64, room_height-64);
}
escrever_estado = "PATRULHANDO";
```

Perseguir - Possui os códigos necessários para gerar a ação de Perseguir do Agente.

```
// Variaveis para Perseguicao
veloc_perseg = 2;
dist_perseg = 300;

//Calcular a distância entre o Agente e o Jogador
if distance_to_object(obj_player) <= dist_perseg{
// Vai na direção do objeto (jogador) na posição x e y, na velocidade 2
    move_towards_point( obj_player.x, obj_player.y, veloc_perseg)
    escrever_estado = "PERSEGUINDO";
}
```

Criar Obstáculo - Possui os códigos necessários para gerar a ação de Criar Obstáculo do Agente.

```
// Variaveis para Perseguicao
veloc_perseg = 2;
dist_perseg = 300;

//Calcular a distância entre o Agente e o Jogador
if distance_to_object(obj_player) <= dist_perseg{
if (frequencia > 0) {
    frequencia -= 1;
} else {
```

```

// Crie o objeto
instance_create_layer(x, y, "Instances", obj_superball_yellow2);
frequencia = 360; // Configurado novamente para 60 passos (1 segundo a 60 FPS)
}
escrever_estado = "VIDA";

}

```

Atacar - Possui os códigos necessários para gerar a ação de Atacar do Agente.

```

veloc_atk = 2;
dist_atk = 100;
//cooldown = 0; //variavel colocada dentro do crear do obj

if distance_to_object(obj_player) <= dist_atk{
    if (frequencia > 0) {
        frequencia -= 1;
    } else {
        // Crie o objeto
        instance_create_layer(x, y, "Instances", obj_superball_yellow);
        frequencia = 360; // Configurado novamente para 60 passos (1 segundo a
60 FPS)
    }
    escrever_estado = "SUPERBALL";
}
}

```

Game Over - Possui os códigos necessários para gerar a ação de GAME OVER do Agente.

```

// Variaveis para Perseguiçao
veloc_perseg = 2;

```

```
dist_perseg = 0;
```

```
//Calcular a distância entre o Agente e o Jogador  
if distance_to_object(obj_player) <= dist_perseg {  
    game_restart()  
  
    escrever_estado = "GAMEOVER";  
}
```

Diagrama Decision Tree

Base para criar a árvore e os nós de ação e nós de decisão.

