

NOVO JOGADOR

JOGAR

SCORE

TACTICS POWER



CENÁRIO

O jogo ocorrerá em um cenário 2D, cuja batalha ocorrerá em formato x1 contra a máquina, no qual, o jogador controlará por turno um avatar de sua equipe, bem como a máquina, num total de 6 avatares já pré-definidos, não poderá ser feita qualquer alteração quanto as características de cada avatar da equipe. O jogo será desenvolvido tendo como base o jogo tactics core.

JOGADORES

Os jogadores possuem nome, vitórias, derrotas e score geral. Ao vencer um combate sua pontuação será calculada em score geral considerando quantidade de movimentos efetuados e velocidade da partida, analogamente em caso de derrota.

BANCO DE DADOS

O jogo possuirá um banco de dados que armazenará as estatísticas dos jogadores, podendo carregar um jogador novo antes de iniciar o jogo.

O JOGO

O combate inicia assim que o jogador confirmar o inicio da partida. Sempre o jogador iniciará o turno e poderá se movimentar, atacar ou passar o turno com o avatar que, ao iniciar a partida, será escolhido aleatoriamente para ambos os jogadores. Caso o jogador efetuar um ataque o mesmo não poderá se movimentar e poderá somente encerrar seu turno. Por se tratar de um jogo

contra a máquina não haverá tempo para cada turno, no entanto haverá um contador de tempo da partida, desde seu início ao termino da mesma.

Cada avatar terá uma pré-visualização de seu deslocamento ou ataque como no seguinte exemplo:



Figura 1 – Exemplo de ataque do avatar, foto do jogo tactics core

Analogamente à figura 1, o jogo mostrará uma barra de vida de cada avatar. Os avatares terão habilidades únicas que os caracterizarão como tanque, mago, curandeiro e etc. Assim como habilidades passivas únicas que poderão ser ativadas caso seja atingido seu pré-requisito. As habilidades dos avatares poderão causar diversos efeitos como: dano crítico ou burn ao longo do tempo, cada ataque será único e logo colocarei nesse documento os cálculos de dano para cada classe, contudo o calculo geral terá como base a seguinte ideia:

$$dano_final = atq - def$$

O ataque (atq) base de um avatar respeitará, de maneira geral, a seguinte formula:

$$atq = calculo\ a\ ser\ trabalhado$$

A defesa (def) base de um avatar respeitará de, de maneira geral, a seguinte formula:

$$def = calculo\ a\ ser\ trabalhado$$

Assim como o ataque e a defesa a cura será:

$$cura = calculo\ a\ ser\ trabalhado$$

O jogo terá como engine o godot.

LISTA DE SUBSTANTIVOS E VERBOS

Substantivos	Atribuição
Cenário	Classe
Jogador	Classe
Avatar	Classe
Nome_Jogador	Atributos do jogador
Vitórias_Jogador	Atributos do jogador
Score_Jogador	Atributos do jogador
Derrotas_Jogador	Atributos do jogador
Combate	Classe
IA	Classe
Jogo	Classe
Pontos_de_movimento	Atributos do avatar
Ataque	Atributos do avatar
Defesa	Atributos do avatar
Vida	Atributos do avatar
Ação	→ verbalizado
Menu	→ verbalizado
Grid Skills	→ Verbalizado
Engine	Classe
Mouse	Classe

Verbos	Atribuição
Criar jogador	Jogador
Cenário inicia o combate	Cenario
Indicar as ações do avatar	Cenario
Atacar avatar	Jogador
Movimentar avatar	Jogador
Passar a vez do avatar	Jogador
Sair do jogo	Jogador
Contador da partida	Cenario
Ver score	Cenario
Dizimar_Jogador	IA

DETALHAMENTO DE REQUISITOS E IDENTIFICAÇÃO DE CLASSES

1. A Classe cenario será responsável por manipular o grid de visualização das habilidades dos avatares;
2. A Classe jogador se responsável por indexar o id para o jogador e armazenar suas informações no banco de dados;
3. A Classe avatar será responsável por atribuir os valores dos atributos base dos avatares e suas respectivas classes e realizar o cálculo de dano das mesmas utilizando argumentos matemáticos propostos em seus métodos;
4. A Classe combate fará um link entre as ações do jogador por meio do avatar e o cenario, verificando cada ação segundo as regras pré-definidas;
5. A Classe jogo será instanciada assim que o jogo for aberto e fará as verificações quanto à inicialização, tamanho da tela e encerramento da aplicação;
6. A Classe IA irá controlar seus avatares de maneira automática com estratégias sinistras para dizimar o jogador;
7. O atributo score do jogador analisará a pontuação ao longo da partida do jogador considerando número de movimentos e o tempo da partida;
8. O atributo pontos de movimento será responsável por delimitar a quantidade de ações que tangem a movimentação do avatar que será previamente mostrado no cenário;
9. A Classe engine será responsável por vincular as animações dos avatares no cenário, bem como controlar a música e efeitos sonoros das skills.

DIAGRAMA DE CLASSES

pkg

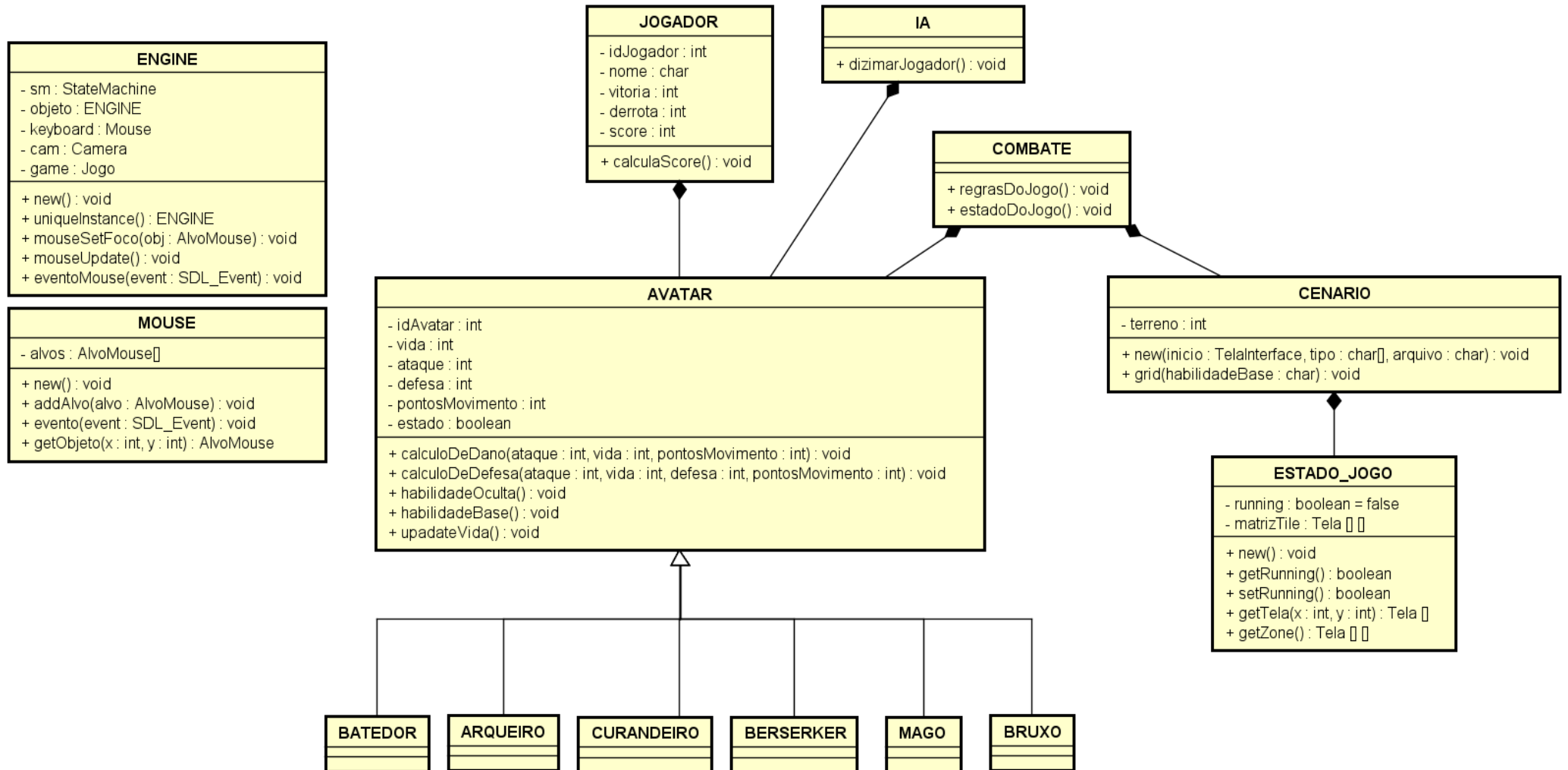


Figura 2 – Diagrama de Classe do jogo

DETALHAMENTO DO DIAGRAMA DE CLASSES

De acordo com o diagrama de classes, há classes que serão modeladas seguindo o padrão MVC nos quais será modeladas da seguinte maneira:

MODEL:

Classe JOGADOR

Classe AVATAR

VIEW:

Classe: CENARIO

CONTROLL:

Classe COMBATE

Classe ENGINE

Classe JOGO

DIAGRAMAS DE CASO DE USO

TELA INICIAL

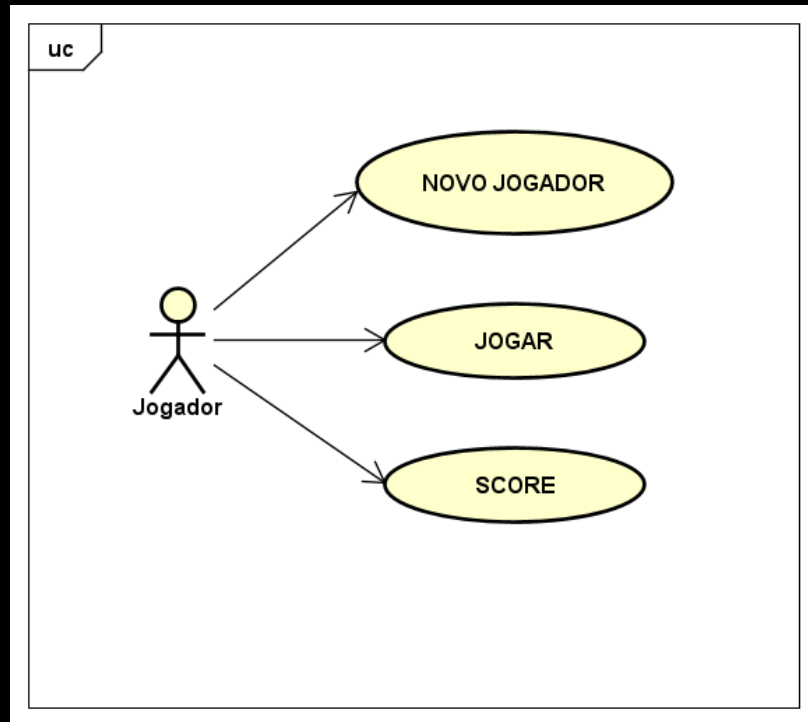


Figura 3 – Tela principal do jogo

TELA NOVO JOGADOR

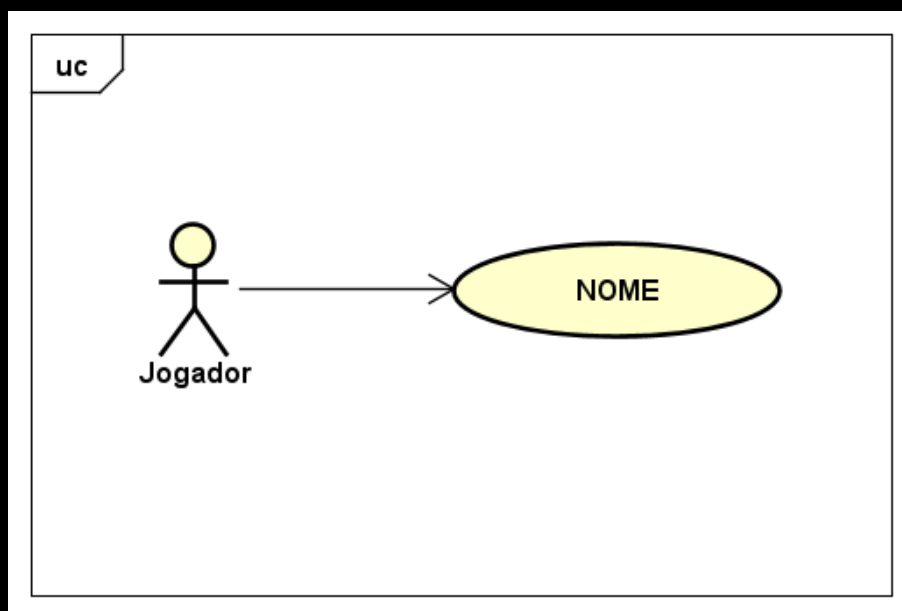


Figura 4 – Tela para inserir novo jogador

COMBATE

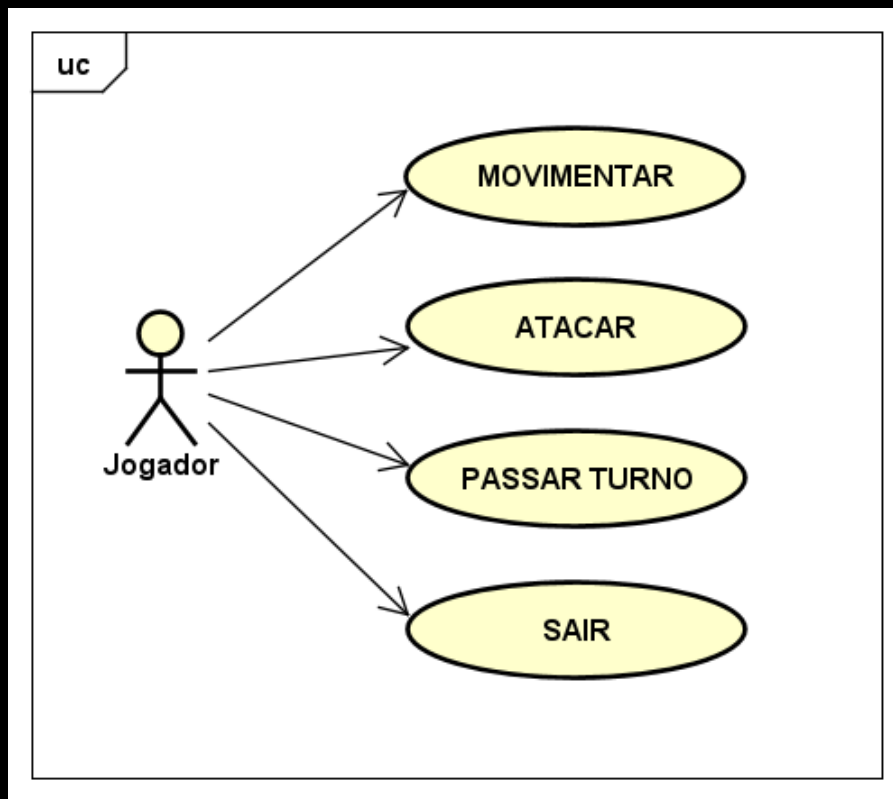


Figura 4 – Tela: combate

As figuras 3,4 e 5 mostram as funcionalidades que haverá em cada tela. De tal forma que na Tela inicial o usuário (jogador) poderá inserir seu nick na opção novo jogador, iniciar o jogo e o score dos jogos anteriores, caso o usuário iniciar o jogo sem personalizar seu nick o jogo fornecerá um de acordo com um contador interno, exemplo: combatente1, combatente2...

A Tela novo jogador é bem intuitiva, na qual aparecerá um espaço para inserir o nick.

A Tela combate terá todo o cenário e o jogador terá as funções para cada um dos seis avatares propostos, cujas funções são de movimentar, atacar, passar o turno ou sair do jogo.