

Maze TCC: Uma Jornada Acadêmica

Você sobreviveu ao Ensino Médio. Acha que já zerou a vida? Pense de novo. Você acaba de ser convocado para a arena mais desafiadora, imprevisível e gloriosa de todas: a FACULDADE!

Embarque em uma aventura única! Desafios acadêmicos testarão sua resiliência. Um calouro de faculdade descobrirá as aventuras do mundo acadêmico, com desafios diários como comer em um restaurante universitário meio suspeito, interagindo com os colegas e passando pelos obstáculos criados pelos seus professores.

Crie seu personagem, defina sua trajetória e supere obstáculos com astúcia e bravura. O sucesso do seu TCC está em suas mãos. Prepare-se para marcar sua história na graduação!

Desenvolvedores

- Ageu V. de Melo - ageuvianademelo003@gmail.com
- Arnoldo F. de Sá - arnoldofurtado@gmail.com
- Lucas B. C. Santos - lucasbonan.cs@gmail.com
- GitHub: github.com/Fibonancci/mazetcc

Atributos do Personagem

Estes atributos definem suas capacidades e influenciam sua jornada acadêmica:

- **Estamina:** Seu nível de tensão interna. Prejudica ações, relações e saúde mental.
- **Reputação:** A percepção de professores e colegas sobre você. Impacta oportunidades e relacionamentos.
- **Motivação:** Sua determinação para seguir estudando, mesmo diante de dificuldades.
- **Sabedoria:** Sua capacidade de usar recursos de forma inteligente (ex: Google), priorizar tarefas e evitar armadilhas acadêmicas.
- **Nota:** Seu desempenho nas disciplinas, nem sempre justo, mas sempre presente.

Personagem Inicial: Calouro(a) Inocente

- **Nome:** Rodolfo
- **Curso:** Ciência da Computação
- **Período:** 1º semestre

Atributos Iniciais:

Atributo	Valor	Descrição Rápida
Estamina	25 ▾	Começou animado, prazos ainda não são um problema
Reputação	25 ▾	Poucos contatos, mas sem inimizades
Motivação	25 ▾	Cheio(a) de expectativas e objetivos
Sabedoria	25 ▾	Conhecimento básico, confia em tutoriais online
Nota	10 ▾	Média neutra, sem grandes surpresas acadêmicas

Ideias para o Jogo

- **Festa Secreta no Laboratório:** Um professor permissivo cede o laboratório para "eventos culturais" noturnos. Participar pode render um pendrive com código útil ou

um resfriado.

- **O Ponteiro Fantasma:** Um erro de memória causa bugs aleatórios no mundo real. Um minigame seria necessário para encontrar e corrigir o ponteiro.
- **Prova de Lógica com Pegadinhas:** Questões complexas que prendem o jogador em loops de justificativas se erradas.
- **O Acadêmico da Conspiração:** Um colega com teorias sobre a universidade ser controlada por IA.
- **Profª Lígia, a Sábia:** Respostas enigmáticas, mas pode oferecer acesso a um "livro secreto" mediante resposta correta.
- **O Técnico de TI Invisível:** Surge apenas ao xingar o Wi-Fi em voz alta, com ajuda incerta.
- **O Colega NPC "Vou Trancar":** Presente em todos os semestres, mas nunca desiste do curso.

Interações e Consequências

1. Conversas no RU:

- **Cena:** Debate sobre a capacidade de 100 homens derrotarem um gorila.
- **Opções:**
 - a) Concordar: +Networking, +Diversão
 - b) Questionar: +Conhecimento, -Networking
 - c) Desviar o assunto para o TCC: +Foco, -Diversão

2. Pedir Ticket no RU:

- **Cena:** Colega sem dinheiro pede um ticket.
- **Opções:**
 - a) Oferecer um ticket: +Networking, -Pontos (a serem definidos)
 - b) Recusar: +Foco, -Networking
 - c) Sugerir pedir a outros: +Persuasão (sucesso variável)

3. Problemas de Programação (C):

- **Cena:** Erro de ponteiro.
- **Opções:**
 - a) Revisar o código: +Conhecimento
 - b) Pedir ajuda: +Networking, -Foco
 - c) Tentar outra abordagem: -Conhecimento, maior chance de erro

4. Questões de Lógica Proposicional:

- **Texto:** Definição da negação em lógica.
- **Pergunta:** Verdadeiro ou falso?
- **Opções:**
 - a) Verdadeiro: +Conhecimento
 - b) Falso: -Conhecimento

5. Convites para Festas:

- **Cena:** Convite para a festa da Atlética.
- **Opções:**
 - a) Ir à festa: +Networking, +Diversão, -Conhecimento (potencial)
 - b) Estudar: +Foco, +Conhecimento, -Networking
 - c) Ir por pouco tempo: +Networking (menor), +Conhecimento (menor), +Diversão (menor)

Extras (Não implementados)

- **Bufs/Debuffs:** Efeitos temporários que alteram atributos (ex: café aumenta o foco, ressaca reduz persuasão e conhecimento).
- **Quizzes com Professores (Estilo Visual Novel):** Responder corretamente a perguntas aumenta conhecimento e amizade com o professor, desbloqueando dicas para o TCC. Exemplo com Profª Fernanda sobre negação lógica.
- **Eventos Aleatórios:** Acontecimentos inesperados com consequências (ex: pane no notebook causa perda temporária de conhecimento não salvo).

Implementação do Jogo Baseado em Labirinto ASCII

Objetivo

Desenvolver a estrutura do jogo utilizando um labirinto pré-definido em formato ASCII como base. O labirinto será representado por uma matriz, facilitando a manipulação e a movimentação do personagem.

Detalhes do Labirinto

- **Formato:** Labirinto ASCII representado por uma matriz de caracteres.
- **Definição:** O labirinto será definido previamente, com paredes, caminhos e pontos de interesse já estabelecidos.
- **Vantagens:**
 - Estrutura simples e fácil de implementar.
 - Permite foco nas mecânicas de jogo e interações.
 - Facilidade de expansão e modificação do labirinto.

Estrutura da Matriz do Labirinto

A matriz representará o labirinto da seguinte forma:

Símbolo	Significado
■	Parede
(espaço)	Caminho Livre
:)	Personagem
■	Ponto de Chegada
※	Pontos de Interesse (itens, NPCs, etc.)

Implementação Inicial

1. **Definir o Labirinto:** Criar uma matriz predefinida com o layout do labirinto usando os símbolos acima.
2. **Representação Visual:** Exibir o labirinto na tela utilizando os caracteres da matriz.
3. **Movimentação do Personagem:** Implementar a movimentação do personagem dentro da matriz, verificando colisões com as paredes ("■").
4. **Interações:** Adicionar interações básicas com pontos de interesse, como coletar itens ou iniciar diálogos.

Próximos Passos

- **Deteção de Colisões:** Implementar uma função robusta para detectar colisões com as paredes.
- **Controle do Personagem:** Criar um sistema de controle que permita ao jogador mover o personagem pelo labirinto.
- **Objetivos do Jogo:** Definir objetivos claros para o jogador dentro do labirinto (ex: encontrar a saída, coletar itens).
- **Elementos Adicionais:** Adicionar elementos como inimigos, puzzles ou eventos aleatórios.

Estruturas de Dados do Jogo

- **Personagem:** Representa o jogador e seus atributos.
 - **nome:** Nome do personagem.
 - **estamina, reputacao, motivacao, sabedoria, nota:** Atributos do personagem.
 - **labirintos_zerados:** Quantidade de labirintos completados.
 - **estado_atual:** Status do personagem (matriculado, jubilado, formado).
- **Ranking:** Armazena os nomes e pontuações dos melhores jogadores.
 - **nomes[TAMANHO_RANKING]:** Array de nomes.
 - **pontuacao[TAMANHO_RANKING]:** Array de pontuações.

- **Desafio**: Detalhes de um desafio específico no jogo.
 - `id`, `nome`, `arquivoTela`, `descricao`: Informações do desafio.
 - `estamina_buff`, `estamina_debuff`, etc.: Modificadores de atributos ao aceitar/resolver o desafio.
 - `opcoes[3][10]`: Matriz para opções de interação do desafio.
 - **ListaDesafios**: Contém uma coleção de objetos **Desafio**.
 - `desafios[MAX_DESAFIOS]`: Array de desafios.
 - `numDesafios`: Número atual de desafios carregados.
 - **SalvarJogo**: Estrutura para gerenciar os dados do jogo salvo.
 - `perfil`: Contém a estrutura **Personagem** a ser salva/carregada.
-

Funções do Game

Funções de **Personagem** (`personagem.h` / `mazetcc.c`)

- `void exibirPersonagem(const Personagem *p)`: Exibe os atributos e informações do personagem.
- `int criarPersonagem(Personagem *p)`: Permite ao jogador criar um novo personagem, preenchendo seus atributos.
- `int verificarMortePersonagem(const Personagem *p)`: Verifica se o personagem atingiu alguma condição de "morte" (provavelmente jubileamento ou game over).

Funções de **Ranking** (`ranking.h` / `mazetcc.c`)

- `void inicializarRanking(Ranking *r)`: Prepara a estrutura do ranking, provavelmente com valores padrão.
- `int adicionarAoRanking(Ranking *r, const Personagem *p)`: Adiciona a pontuação de um personagem ao ranking.
- `int carregarRanking(Ranking *r, const char *nomeArquivo)`: Carrega os dados do ranking de um arquivo.
- `void exibirRanking(const Ranking *r)`: Mostra a lista de jogadores e suas pontuações no ranking.
- `int salvarRanking(Ranking *r, const char *nomeArquivo)`: Salva os dados do ranking em um arquivo.

Funções de **Salvar** (`salvar.h` / `mazetcc.c`)

- `SalvarJogo carregarSave()`: Carrega os dados de um jogo salvo.
- `int salvarSave(SalvarJogo *s)`: Salva o estado atual do jogo.
- `int checarSave(SalvarJogo s)`: Verifica se há um jogo salvo válido.

Funções de **Desafio** (`desafio.h` / `mazetcc.c`)

- `void inicializarListaDesafios(ListaDesafios *lista)`: Inicializa a estrutura da lista de desafios.
- `int carregarTodosDesafios(ListaDesafios *lista, const char *nome_arquivo_desafios_geral)`: Carrega todos os desafios de um arquivo de texto.
- `Desafio *buscarDesafioPorId(ListaDesafios *lista, int id)`: Procura um desafio específico na lista pelo seu ID.
- `void exibirTelaDesafio(const char *nome_arquivo)`: Exibe o conteúdo de um arquivo de texto que representa a tela de um desafio.

Funções de **Tela de Desafio** (tela_desafio.h / mazetcc.c)

- `void exibirTelaResultadoDesafio(int sucesso, const char *mensagemAdicional, int estamina_mod, int reputacao_mod, int motivacao_mod, int sabedoria_mod, int nota_mod)`: Exibe uma tela de resultado de desafio, mostrando sucesso/falha e modificadores de atributos.
- `void exibirTelaDesafioAceitarRecusar(int id_desafio, const char *nome_desafio, const char *descricao_desafio)`: Exibe a tela onde o jogador decide aceitar ou recusar um desafio.
- `int exibirTelaInteracaoDesafio(const char *nome_arquivo_tela)`: Parece exibir uma tela específica de interação para um desafio.
- `void exibirTelaGameOver()`: Mostra a tela de "Fim de Jogo".

Funções de **Labirinto** (labirinto.h / mazetcc.c)

- `void esculpir_passagens(int cx, int cy)`: Função auxiliar para a geração do labirinto.
- `void gerar_mapa_aleatorio()`: Gera o layout do labirinto de forma aleatória.
- `void monta_mapa(Personagem *personagemAtual)`: Monta/exibe o mapa do labirinto com o personagem.
- `int my_getch(void)`: Provavelmente uma função para leitura de caractere sem a necessidade de Enter.
- `int exibirLabirinto(ListaDesafios *listaDeTodosDesafios, Personagem *personagemAtual, SalvarJogo *saveCarregado)`: Função principal para exibir e permitir a interação no labirinto.
- `void desafiarJogador(ListaDesafios *listaDeTodosDesafios, Personagem *personagemAtual)`: Lógica para iniciar um desafio quando o jogador encontra um.
- `void desenhaBarraStatus(const char *atributo, Personagem *personagemAtual)`: Desenha a barra de status de um atributo do personagem.

Funções Utilitárias Gerais (mazetcc.c)

- `void limparBufferEntrada()`: Limpa o buffer do teclado após leituras (como `scanf`).

- `void printLento(const char *texto, int tempo)`: Exibe o texto caractere por caractere com atraso.
 - `void exibirInstrucaoJogo()`: Mostra as instruções de movimentação e elementos do labirinto.
 - `void pausa(int tempo)`: Pausa a execução do programa por um tempo especificado.
 - `void exibirIntruducacaoDoJogo()`: Exibe a introdução narrativa e o título ASCII art do jogo.
 - `void exibirLogo()`: Exibe apenas o logo/título ASCII art.
 - `void aguardarVoltar(int opcaoSaida, const char *msgOpcao)`: Espera o jogador pressionar uma opção para continuar.
 - `int verificarInteiro(int *valor, int min_val, int max_val)`: Valida se uma entrada do usuário é um inteiro dentro de um determinado intervalo.
 - `int exibirCreditos()`: Exibe os créditos do jogo, lendo de um arquivo.
 - `void limparTela()`: Limpa o console/terminal.
 - `void exibirMenu(Ranking *rankingDoJogo, Personagem *personagemAtual, SalvarJogo *saveCarregado, ListaDesafios *listaDeTodosDesafios)`: A função principal que gerencia a navegação do menu do jogo.
-

Telas do Game (Fluxo e Conteúdo)

- **Tela de Introdução (`exibirIntruducacaoDoJogo`):**
 - Exibe a narrativa do jogo ("Você sobreviveu ao Ensino Médio...", "Faculdade!").
 - Apresenta o título do jogo em ASCII Art de forma animada (com `pausa`).
 - Aguarda input para prosseguir para o menu.
- **Tela de Menu (`exibirMenu`):**
 - Mostra o logo/título do jogo.
 - Opções principais:
 - **Novo Jogo**: Inicia um novo jogo, chamando `criarPersonagem`.
 - **Carregar Jogo Salvo**: Carrega um save existente.
 - **Instruções**: Leva à tela de instruções.
 - **Ranking**: Exibe o ranking.
 - **Créditos**: Mostra a tela de créditos.
 - **Sair**: Finaliza o jogo (com possível função de `salvarJogo` antes).
- **Tela de Criação de Personagem (`criarPersonagem`):**
 - Pede o nome do personagem.
 - Permite preencher os atributos iniciais (estamina, reputação, etc.).
 - Salva o personagem (provavelmente chamando `salvarSave`).
- **Tela de Personagem (`exibirPersonagem`):**
 - Apresenta os atributos e informações atuais do personagem.
 - Opções como "Jogar" (leva ao labirinto) ou "Editar" (volta à tela de criação).

- **Tela de Jogo / Labirinto (`exibirLabirinto`):**
 - Exibe o mapa do labirinto (com `monta_mapa`).
 - Mostra informações do personagem (status, atributos) (provavelmente com `desenhaBarraStatus`).
 - Permite a movimentação do personagem.
 - Gerencia interações e desafios (`desafiarJogador`).
 - Oferece opções para "Sair/Salvar".
- **Tela de Desafio (`exibirTelaDesafio`, `exibirTelaDesafioAceitarRecusar`, `exibirTelaInteracaoDesafio`):**
 - Mostra detalhes de um desafio.
 - Apresenta opções de interações do desafio.
 - Exibe os atributos atuais do personagem durante o desafio.
 - Ao finalizar, leva à `exibirTelaResultadoDesafio` e pode chamar `pontuarAtributo` (ou a lógica de modificação de atributos).
- **Tela de Resultado de Desafio (`exibirTelaResultadoDesafio`):**
 - Informação de sucesso ou falha no desafio.
 - Exibe as modificações nos atributos do personagem.
- **Tela de Ranking (`exibirRanking`):**
 - Lista os nomes e pontuações dos melhores jogadores.
- **Tela de Créditos (`exibirCreditos`):**
 - Exibe o conteúdo do arquivo `data/creditos.txt`.
- **Tela de Instruções (`exibirInstrucaoJogo`):**
 - Explica os comandos de movimentação e os símbolos do labirinto.
- **Tela de Game Over (`exibirTelaGameOver`):**
 - Apresentada quando o personagem é "jubilado" ou atinge uma condição de falha.