Instituto Federal de São Paulo (IFSP)
Curso: Desenvolvimento de Sistemas

Arthur Egidio Vieira Herruzo
Lucas Henrique Rodeiro da Silva
Miguel Ferreira da Silva

Math Championship

São Paulo
2025

Arthur Egidio Vieira Herruzo
Lucas Henrique Rodeiro da Silva
Miguel Ferreira da Silva

Math Championship

Documentação de Projeto Interdisciplinar apresentado como requisito parcial para avaliação nas disciplinas de Programação Web, Lógica de Programação, Arquitetura de Computadores e Redes, Matemática e IFDS.
Professores: Ana Lucia Grici Zacarin Mamede, Claudete de Oliveira Alves, Claudia Miyuki Werhmuller, Antonio Ferreira Viana, Daniela dos Santos Santana, André Evandro Lourenço, João Antonio Temochko Andre, Paula Neves de Araújo, Paulo Henrique Netto de Alcantara e Gislene Pereira de Oliveira Martins.
São Paulo
2025

Sumário

1. INTRODUÇÃO 4
2. PLANEJAMENTO DO PROJETO 5
3. DESENVOLVIMENTO 6
3.1 Design e Arte (A 'Cara' do Jogo) 6
3.2 Arquitetura e Código (O 'Cérebro' do Jogo) 10
3.3 O Site de Divulgação (HTML/CSS) 14
4. PROTÓTIPO E TESTES 15
5. CONCLUSÃO 16
REFERÊNCIAS 17

1. INTRODUÇÃO

O desafio proposto pelos professores consistiu na criação de um jogo, para podermos aprendermoa melhor lógica, foi recomendado escolher uma matéria, decidimos escolher matemática pois é um bom tema paara jogos e nós gostamos.

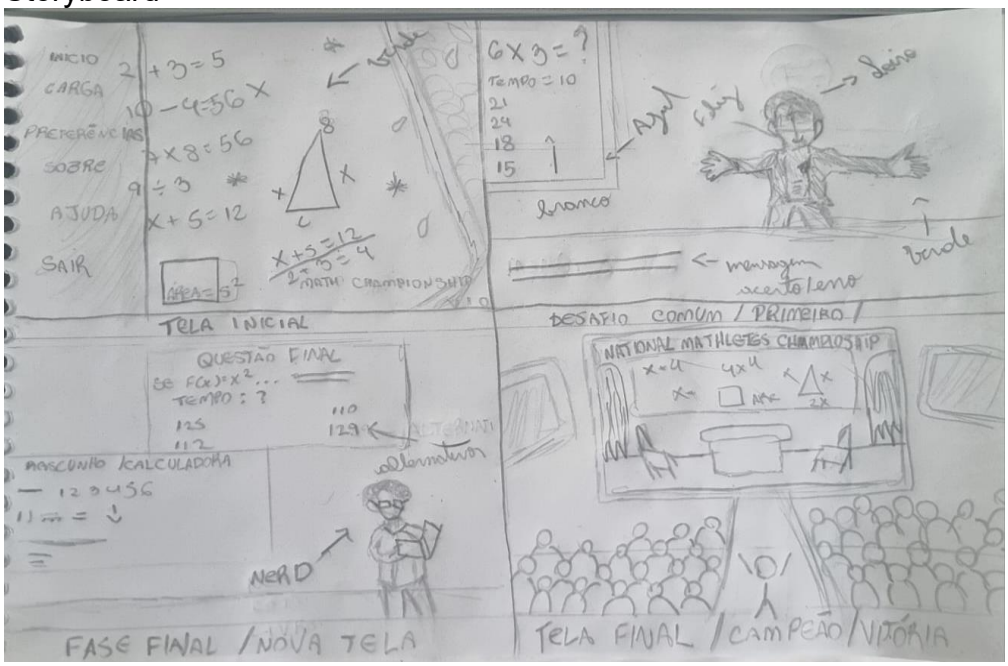
O jogo 'Math Championship' foi desenvolvido com o intuito de desafiar o raciocínio lógico e matemático dos jogadores por meio de fases progressivas. O protagonista é um aluno do primeiro ano do ensino médio apaixonado por desafios e lógica, que decide participar do Campeonato Anual de Matemática para provar seu valor. Enfrentando rivais mais experientes, ele avança fase após fase, mostrando inteligência e determinação.

O público-alvo do jogo são estudantes do ensino médio interessados em desafiar suas habilidades de raciocínio e matemática de forma divertida. Este documento apresenta o planejamento, design, desenvolvimento técnico e protótipos do jogo, servindo como registro do processo de criação e aprendizado obtido durante o projeto.

URL: https://miguelferreira300.github.io/site/

2. PLANEJAMENTO DO PROJETO

Pesquisas Realizadas: a equipe se baseou no Estilo de jogo de Doki Doki Literature Club! Pois esse é um dos jogos de visual-novel mais famosos.
Escopo do Jogo: 'Math Championship' é uma visual novel com progressão por fases. Mecânicas principais incluem: leitura de diálogos, testes matemáticos em formato de minijogos e um sistema de fases eliminatórias até a final. O jogo foi pensado para ser curto e escalável para mais fases futuramente.
Storyboard

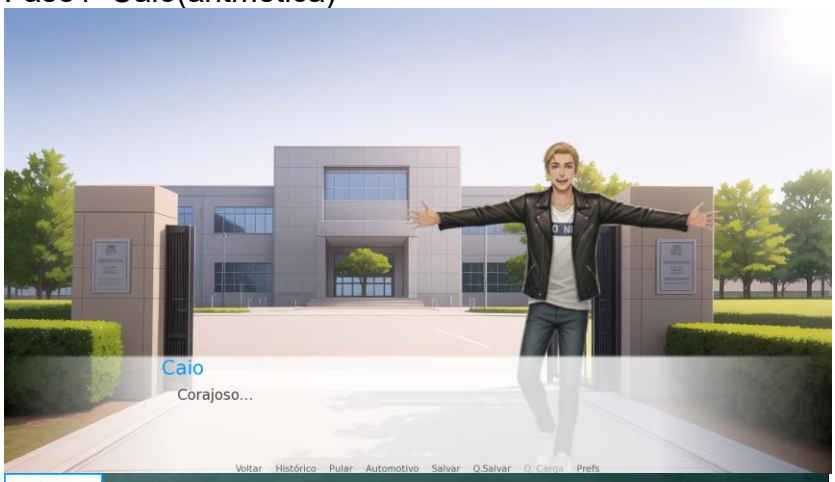


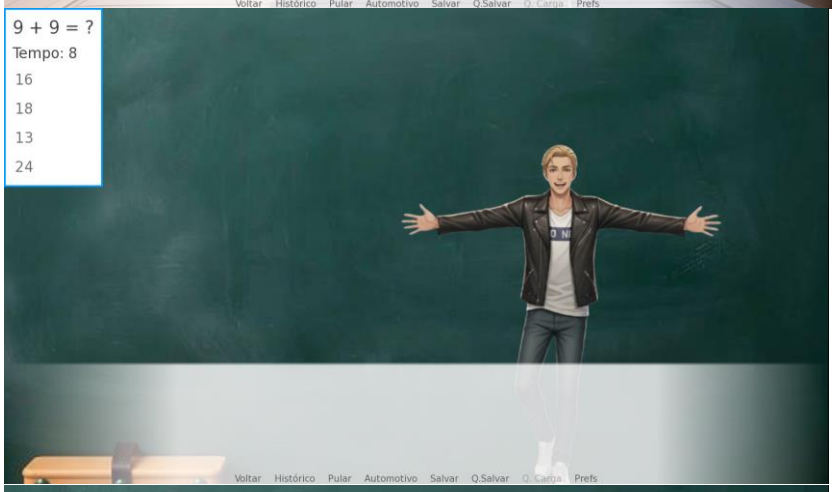
Ferramentas e Tecnologias: A equipe escolheu Ren'Py como engine por sua facilidade para visual novels e suporte a Python.. IA foi usada para ajudar no design dos personagens e algumas imagens de fundo e rascunho de roteiros.
Cronograma e Diário de Bordo: O cronograma foi dividido em 5 semanas: semana 1 – nos reunimos para pesquisar sobre o tema e criamos uma base para o roteiro; semana 2 – realizamos uma reunião para discutir o design de personagens e separar o cenário; semana 3 – começamos a estruturar o jogo em Ren'Py; semana 4 – integramos músicas e realizamos testes; semana 5 – nos reunimos para decidir se íamos adicionar algo e realizamos alguns ajustes.

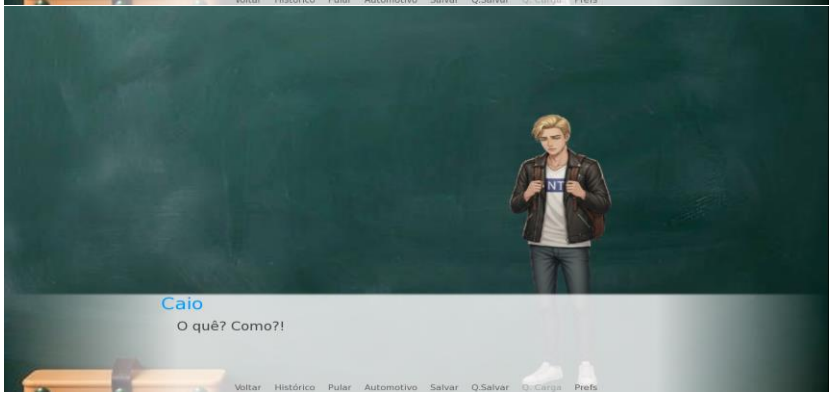
3. DESENVOLVIMENTO

3.1 Design e Arte (A 'Cara' do Jogo)

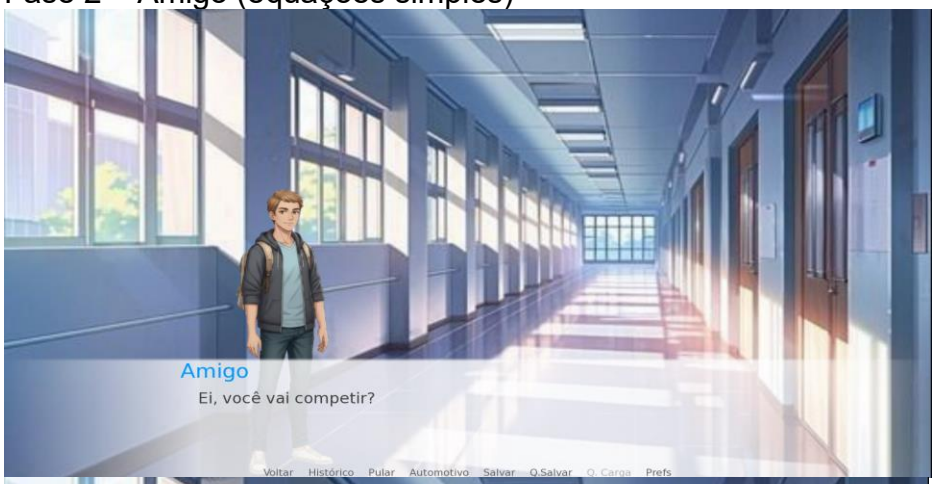
Nosso jogo é uma visual novel de matemática feita em Ren'Py (Python).
Por fora, parece só uma história com personagens conversando e respondendo perguntas.
Por dentro, ele funciona assim:
O jogador é um aluno que participa de uma competição de matemática na escola;
A história é dividida em fases, cada uma com um oponente.
Fase1–Caio(aritmetica)

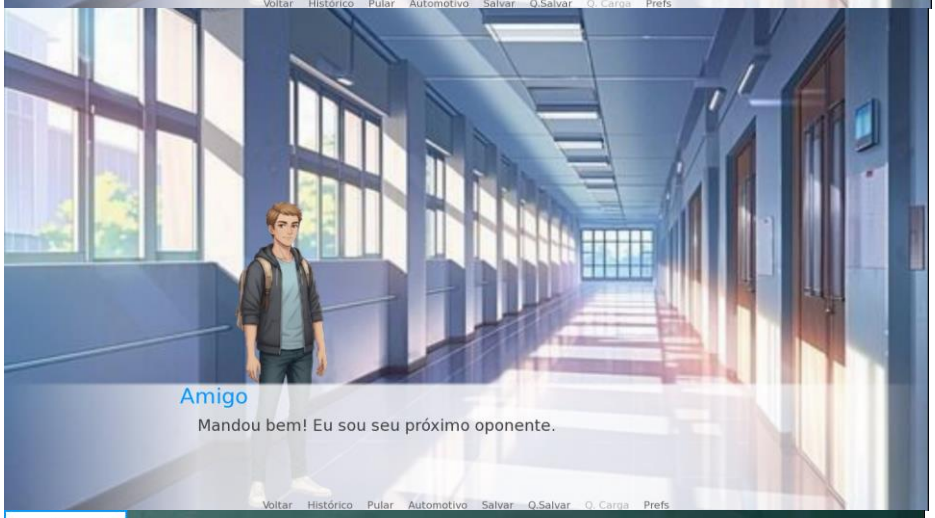




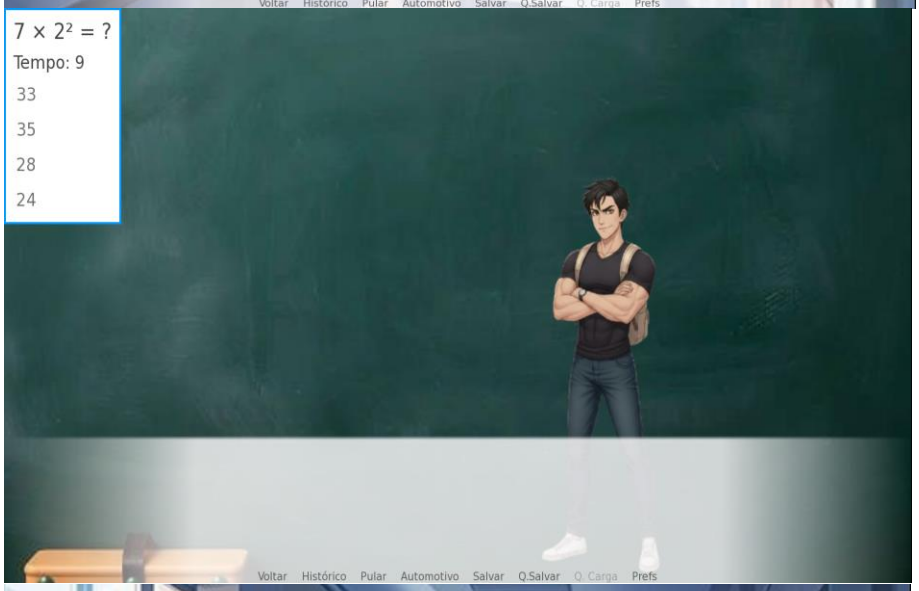


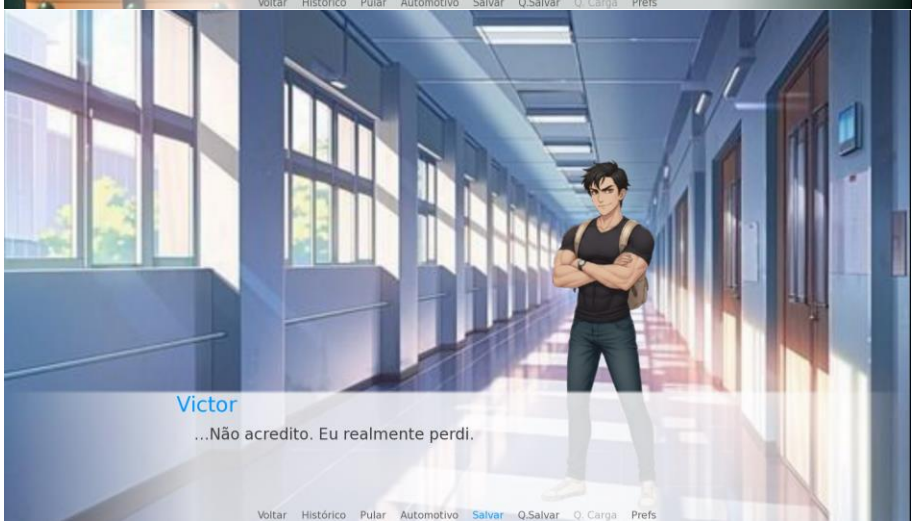
Fase 2 – Amigo (equações simples)



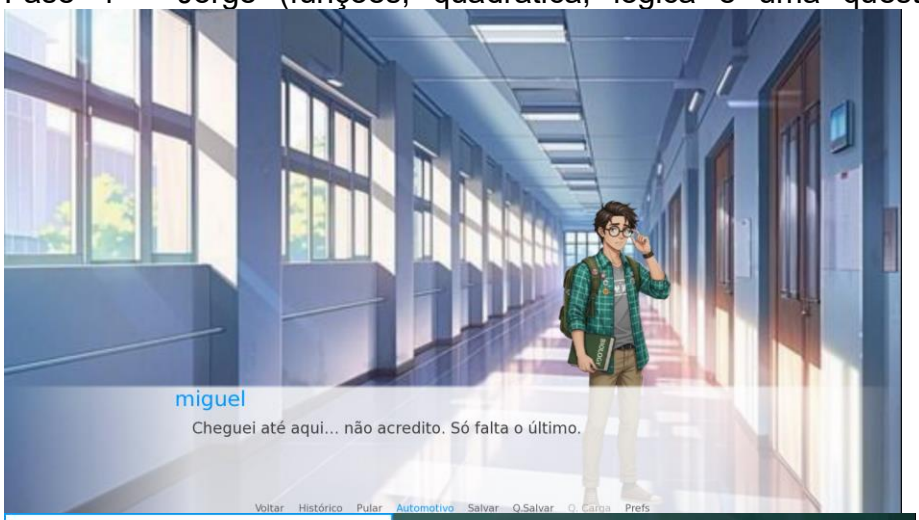


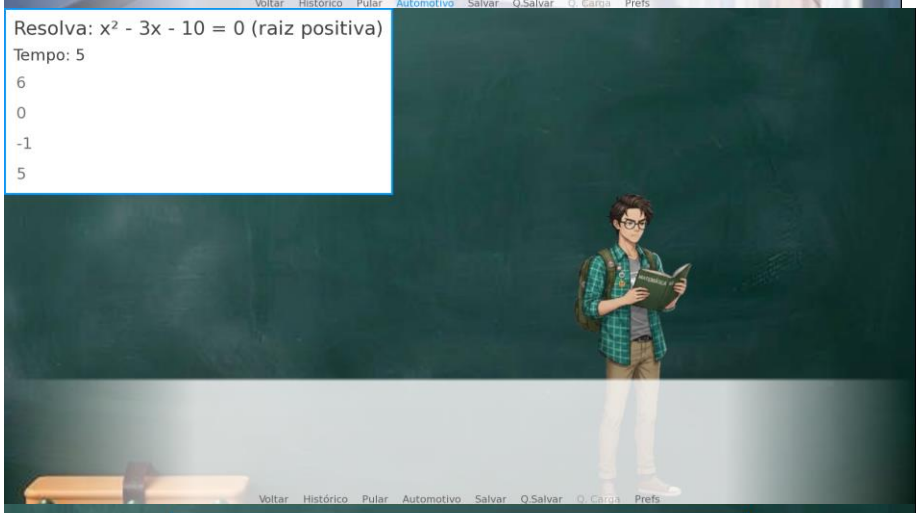
Fase 3 – Victor (raciocínio lógico)

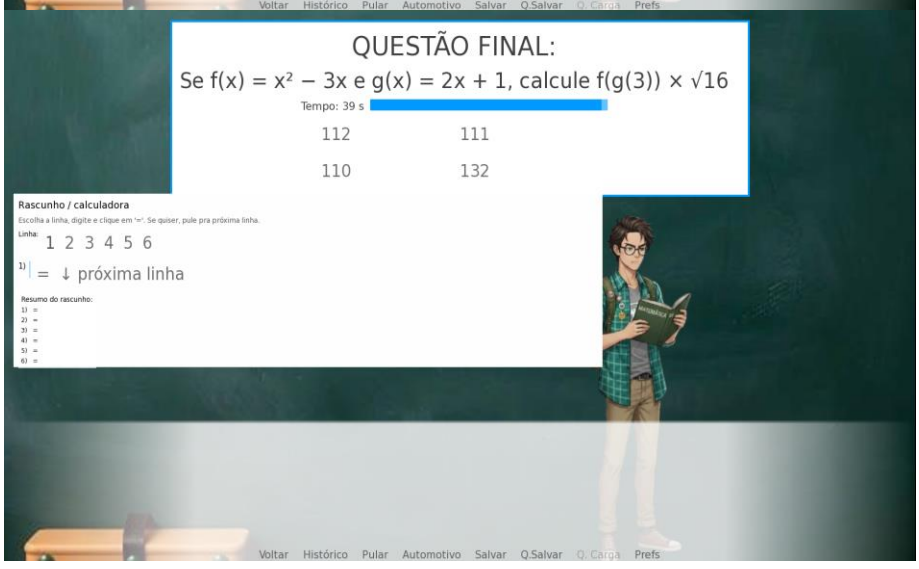




Fase 4 – Jorge (funções, quadrática, lógica e uma questão final épica).







Cada fase chama um minigame de perguntas e respostas:
o código gera perguntas aleatórias
mostra a pergunta em uma tela (screen) com alternativas,

espera a resposta,

confere se está certa e soma pontos.

No final de cada minigame, o jogo decide com if se o jogador passa para o próximo oponente ou se perde e volta a tentar.

Na última fase, ainda existe uma tela especial de “rascunho/calculadora”, onde o jogador pode digitar contas em até 6 linhas diferentes para resolver a questão final.

3.2 Arquitetura e Código (O 'Cérebro' do Jogo)

Lógica Principal: como usamos condicionais (if/else) e laços (for/while)
A lógica principal do jogo é baseada em:
laços for para repetir perguntas,
condicionais if/else para decidir vitória/derrota.
Exemplo 1 – Minigame da Fase 1 (aritmética)
Esse trecho é ótimo pra print porque mostra for + if + função que gera a questão.

```
label mg_fase1_aritmetica:
    $ score = 0
    $ rounds_total = 5
    $ vitoria_necessaria = 3
    $ tempo_por_questao = 10
    python:
        for i in range(rounds_total):
            q, correta, alts = gerar_aritmetica()
            resp = renpy.call_screen("quiz_mc_screen", q, alts, tempo_por_questao)
            if resp == correta:
                score += 1
    if score >= vitoria_necessaria:
        return True
    else:
        return False
```

O que está acontecendo:

for i in range(rounds_total) → faz um laço para repetir 5 perguntas.

Em cada rodada:

gerar_aritmetica() cria a pergunta, a resposta certa e as alternativas.

renpy.call_screen(...) mostra a tela do quiz e pega a resposta do jogador.

if resp == correta: → se o jogador acertar, soma 1 ponto.

No final:

if score >= vitoria_necessaria: → se acertar pelo menos 3, retorna True (venceu a fase).

senão, retorna False (perdeu a fase).
Exemplo 2 – Decisão de vitória/derrota na história

```
call mg_fase1_aritmetica
if _return:
    jump vitoria_caio
else:
    jump derrota_caio_menu
```

O minigame retorna True ou False.

if _return: → se for True, vai para a cena de vitória.

else: → se for False, vai para o menu de derrota (tentar de novo ou desistir).

Esse padrão se repete nas outras fases (álgebra, lógica e final).

Funções e Parâmetros: exemplos importantes

Vou citar duas funções importantes pra mostrar que o projeto tem lógica real e estrutura de programação:

(a) Função que gera questões de aritmética

```
init python:
    def gerar_aritmetica(nivel=1):
        ops = ["+", "-", "×"] + (["÷"] if nivel >= 2 else [])
        amax = 9 if nivel == 1 else 12
        a = random.randint(1, amax)
        b = random.randint(1, amax)
        op = random.choice(ops)
        if op == "+":
            val = a + b
        elif op == "-":
            if a < b:
                a, b = b, a
            val = a - b
        elif op == "×":
            val = a * b
        else:
```

```
            divisor = random.randint(2, max(2, amax // 2))
            val = random.randint(2, max(2, amax // 2))
            a = val * divisor
            b = divisor
        questao = f"{a} {op} {b} = ?"
        alternativas = {val}
        while len(alternativas) < 4:
            alternativas.add(val + random.randint(-7, 7))
        alts = list(alternativas)
        random.shuffle(alts)
        return questao, val, alts
```

Explicação:

Essa função gera automaticamente uma questão de aritmética.
Recebe um parâmetro nivel (por padrão 1) que aumenta a dificuldade.
Sorteia dois números (a e b) e uma operação (+, -, × ou ÷).

Calcula o resultado correto (val).
Monta o enunciado ("a op b = ?").
Cria 4 alternativas (uma certa e três erradas próximas) e embaralha.
Retorna:
o texto da questão, a resposta correta, a lista de alternativas.

(b) Função da calculadora/rascunho da fase final

Essa parte mostra que o jogo tem uma ferramenta de apoio ao jogador — não é só um quiz simples.

```
init python:
    def calc_expr(txt):
        txt = txt.strip()
        if not txt:
            return ""
        txt = txt.replace("x", "").replace("X", "")
        txt = txt.replace(",", ".")
        try:
            return str(eval(txt, {"_builtins_": None}, {}))
        except Exception:
            return "erro"
```

Explicação:

A função recebe o texto digitado pelo jogador.

Limpa os espaços, troca x por * e vírgula por ponto.

Usa eval (de forma segura) para calcular a expressão matemática.

Se o jogador digitar algo inválido, retorna "erro".

Essa função é usada nas variáveis calc_l1, calc_l2, etc., para calcular até 6 linhas diferentes de rascunho na fase final.

Unity na prática: como usamos a engine (Ren'Py + Python)

No lugar do Unity, usamos o Ren'Py, que combina visual novel com Python.
Com ele, controlamos os cenários, os personagens, o quiz com tempo e até a tela da calculadora.

Exemplo de partes da engine:

Personagens:

```
define p = Character("[nome]", dynamic=True)
define f = Character("Amigo")
define c = Character("Caio")
define t = Character("Professor")
define v = Character("Victor")
define l = Character("Jorge")
```

Cenários (backgrounds):

```
image bg sala_classe = bg_cover("images/bg/sala (1).jpeg")
image bg corredor    = bg_cover("images/bg/corredor2.jpeg")
image bg lousa       = bg_cover("images/bg/lousa.jpeg")
```

Tela do quiz com tempo (screen):

```
screen quiz_mc_screen(questao, alternativas, tempo=10):
    default tempo_restante = tempo
    frame:
        vbox:
            text "[questao]" size 40
            text "Tempo: [tempo_restante]"
            for alt in alternativas:
                textbutton str(alt) action Return(alt)
    timer 1.0 repeat True action SetScreenVariable("tempo_restante",
tempo_restante - 1 if tempo_restante > 0 else 0)
    timer tempo action Return(None)
```

Tela final (calculadora):

```
screen quiz_final_screen(questao, alternativas, tempo=40):
    frame:
        vbox:
            text "QUESTÃO FINAL:" size 52
            text "[questao]" size 42
    frame:
        vbox:
            text "Rascunho / calculadora"
            hbox:
                text "1)"
                input value VariableInputValue("calc_linha1") length 28
                textbutton "=" action Function(calc_l1)
                text "[calc_res1]"
```

3.3 O Site de Divulgação (HTML/CSS)

O objetivo principal do site é atuar como o centro oficial de divulgação do jogo "Math Championship". A sua função primária é ser a "casa" do projeto na internet, apresentando a história, o visual e, o mais importante, fornecendo o link direto para o jogador baixar o jogo.

Além de divulgar, o site também tem um forte objetivo instrucional. Ele serve como um manual que prepara o jogador antes mesmo da instalação, detalhando os requisitos mínimos do sistema, explicando os controles e apresentando os personagens principais.

Por fim, o site funciona como um portfólio para a equipe de desenvolvimento. Ele oferece uma visão "por trás das câmeras" do projeto, exibindo o storyboard, a galeria de assets (imagens e músicas), a linguagem de programação utilizada e até trechos de código. Esta transparência celebra o trabalho da equipe e dá os devidos créditos aos criadores no rodapé.

4. PROTÓTIPO E TESTES

Protótipo e Testes (Jogo rodando)

Protótipo inicial

O primeiro protótipo do jogo era simples:

Apenas as primeiras fases de quiz (aritmética e álgebra).

Menos falas e menos variação nas perguntas.

Sem a calculadora/rascunho na fase final.

A ideia era testar primeiro a mecânica básica:

gerar pergunta → mostrar na tela → receber resposta → somar pontos → decidir se passa ou perde.

Depois disso, evoluímos o projeto:

Adicionamos a Fase 3 (lógica);

Criamos a Fase 4 com funções e lógica mais difícil;

Implementamos a questão épica final e a tela de rascunho/calculadora.

Testes e bugs encontrados

Durante os testes, encontramos e corrigimos vários problemas:

Calculadora só funcionava na primeira linha:

Criamos variáveis separadas (calc_linha1, calc_linha2, etc.) e um seletor de linha (calc_linha_atual) para alternar.

Erros de indentação no Ren'Py:

Corrigimos blocos vbox, hbox e frame com recuos errados até o script compilar sem erros.

Fluxo de vitória/derrota incorreto:

Alguns retornos (return True/False) estavam faltando; revisamos todos os if _return: das labels.

Tempo das questões:

Ajustamos o tempo para 10–12 segundos nas fases e 40 segundos na questão final.

Após os ajustes, testamos todas as rotas do jogo:

vencer e perder cada fase,

usar a calculadora com expressões diferentes,

garantir que o jogo nunca travasse nem ficasse sem saída.

5. CONCLUSÃO

Durante o desenvolvimento do Math Championship, aprendemos muito sobre lógica de programação, estrutura de funções, condicionais (if/else), laços de repetição (for/while) e também sobre como o Ren'Py combina Python com narrativa visual.
Percebemos na prática como um jogo precisa ter tanto uma história envolvente quanto um código bem estruturado para funcionar corretamente.
O resultado final ficou ainda melhor do que o planejado no início.
A ideia inicial era apenas um quiz simples de matemática, mas conseguimos evoluir o projeto para uma visual novel completa, com múltiplas fases, personagens, sistema de pontuação e até uma calculadora interativa na fase final, o que deixou o jogo mais criativo e dinâmico.
As maiores dificuldades foram:

Corrigir erros de indentação e blocos do Ren'Py que davam falha na compilação;
Fazer a calculadora funcionar em todas as linhas e não apenas na primeira;
Ajustar o tempo das perguntas e o fluxo de vitória e derrota, para o jogo rodar sem travar;
O que mais gostamos de fazer foi criar as fases com desafios diferentes, programar as perguntas aleatórias e ver o jogo realmente "ganhar vida" com os personagens, músicas e transições.
Se tivéssemos mais um mês de trabalho, gostaríamos de:
Criar novas fases com outros tipos de matemática (como geometria ou probabilidade);
Adicionar sistema de ranking e pontuação final;
Melhorar os efeitos visuais e sonorous;
Criar uma versão jogável online, para que outras pessoas também possam testar o jogo.

REFERÊNCIAS

Algumas imagens de fundo (sem direitos autorais): https://potat0master.itch.io/free-background-music-for-visual-novels-bgm-pack-1?

Outras foram geradas pelo Gemini.

Sons utilizados no jogo (sem direitos autorais):just-relax-11157.mp3

clarity-of-the-sea-calm-piano-232597.mp3

cinematic-epic-warrior-fantasy-or-gaming-soundtrack-417322 - Atalho.lnk

lofi-study-music-361055.mp3

Tutorial do ren'py:
https://www.youtube.com/watch?v=iYE5Ojd96cg&list=PL8hh5X1mSR2CPnQfVZdSOVkgbIfL8wWX2