



TEMA – FESTEJANDO A CONEXÃO CAMPO CIDADE.

NOME DO PROJETO DE PROGRAMAÇÃO:

CIDADE
2025

IDENTIFICAÇÃO DO PROJETO DE PROGRAMAÇÃO

DADOS DA ESCOLA:

1. NOME DA INSTITUIÇÃO: COLÉGIO CÍVICO MILITAR NEWTON SAMPAIO
2. CÓDIGO (INEP):
3. E-MAIL INSTITUCIONAL:
4. TELEFONE:
5. NOME DO DIRETOR:
6. PROJETO SELECIONADO:
 - () SUBCATEGORIA 1
 - () SUBCATEGORIA 2
 - () SUBCATEGORIA 3

DADOS DO(A) PROFESSOR(A) ORIENTADOR:

1. NOME:
2. CPF:
3. RG:
4. E-MAIL:
5. TELEFONE CELULAR:
6. TAMANHO DA CAMISETA:

DADOS DO(A) ESTUDANTE:

1. NOME: HIGO INOCENCIO PRADO
2. CGM: 1008924747
3. CPF: 148.290.259-13
4. RG: 16.720.256-0
5. E-MAIL: HIGO INOCENCIO PRADO
6. TELEFONE CELULAR: 43996089284
7. TAMANHO DA CAMISETA: G

SUMÁRIO

1 INTRODUÇÃO	4
1.1 OBJETIVOS DO PROJETO.....	4
1.2 JUSTIFICATIVA DO PROJETO	4
1.3 LINK DO PROJETO	4
2 DESCRIÇÃO DO PROJETO	5
3 CONSIDERAÇÕES FINAIS	8

1 INTRODUÇÃO

1.1 OBJETIVOS DO PROJETO

Este projeto foi concebido para criar, em p5.js, um endless runner 2D que celebre a conexão entre o campo e a cidade, transformando-a em experiência interativa. Pretende-se:

- Demonstrar como a biblioteca p5.js pode ser usada para gerar elementos gráficos, animações e lógica de jogo de forma simples e didática.
- Engajar o usuário em uma jornada contínua de saltos e desvios, valorizando aspectos rurais (árvores, colinas) e urbanos (prédios, néons) conforme o personagem progride.
- Inspirar e instigar curiosidade acerca de técnicas de desenvolvimento em JavaScript para arte e jogos, incentivando o aprendizado de programação visual.

1.2 JUSTIFICATIVA DO PROJETO

A escolha de p5.js decorre de sua sintaxe intuitiva e do modelo de desenho orientado a “sketch”, que permite que iniciantes criem elementos visuais por meio de funções como `setup()` e `draw()`. Ao longo do desenvolvimento, constatou-se que bibliotecas como p5.js aceleram prototipagem de jogos simples, favorecendo experimentação de animações (squash & stretch), partículas e efeitos de parallax.

Como inspiração, tomou-se o famoso jogo do dinossauro offline do navegador Chrome (jogo T-Rex), em que um personagem salta obstáculos em um cenário monocromático. A mecânica básica de pular cactus foi adaptada: aqui, o stick-figure corre em cenários duplos (rural e urbano) e enfrenta uma variedade maior de obstáculos e power-ups. A partir dessa base, novas ideias foram incorporadas, tais como:

- Ciclo dia-noite alternando sol e lua a cada 1 000 pontos.
- Quatro power-ups distintos (invencibilidade, slow motion, duplo salto e pontuação dobrada).
- Animações de partículas para salto, colisão e coleta de itens.

1.3 LINK DO PROJETO

https://github.com/higoprado/Agriinho/blob/main/Endless_Runner_Projeto_Avancado_2025_06_04_13_27_59.zip

2 DESCRIÇÃO DO PROJETO

Esta seção detalha o fluxo de execução, a arquitetura de classes, as mecânicas de jogo (incluindo a inspiração no dinossauro offline) e as inovações implementadas em p5.js para criar uma experiência envolvente.

O jogo utiliza p5.js para gerenciar canvas, desenho e lógica em três etapas: 1. **preload()** – reserva espaço para futuros assets (imagens, sons); 2. **setup()** – configura `createCanvas(windowWidth, windowHeight)`, `pixelDensity(2)` e `frameRate(60)`; chama `initializeGameElements()` para instanciar player, obstáculos, power-ups, partículas e elementos de fundo; 3. **draw()** – roteia renderização e atualização conforme o estado (`gameState`), separando responsabilidade em helpers como `drawBackdrop()`, `updateGame()`, `drawGame()` e `drawMenu()`.

Cada entidade é uma classe com métodos `update()`, `draw()` e condições de remoção (`isOffscreen()`, `isDead()`):

- **Player**: gerencia física (gravidade, pulo, duplo pulo), animações (squash & stretch) e poder-ups com timers;
- **Obstacle & PowerUp**: deslocam-se com `scrollSpeed`, desenhando-se conforme tipo e despoentilham ao sair da tela;
- **Particle & CollectEffect**: realizam efeitos visuais de poeira, explosão e anel expansivo;
- **RuralBackgroundElement & UrbanBackgroundElement**: suportam parallax scrolling e janelas urbanas que piscam dinamicamente.

Essa modularização facilita inclusão de novos tipos de obstáculos, power-ups ou efeitos sem alterar o loop principal.

A lógica básica de pular objetos em um cenário lateral fixo foi inspirada no jogo T-Rex do Chrome. Assim como o dinossauro desvia cactos, o stick-figure desvia árvores e prédios conforme progride. A inspiração se reflete em: - **Loops infinitos** de geração de obstáculos em intervalos regulares. - **Aceleração progressiva** da velocidade conforme a pontuação aumenta. - **Feedback visual** de colisão (flash vermelho e partículas) para reforçar o sentido de falha.

A partir desta base, foram criadas variações e inovações, como o ciclo dia–noite, power-ups e mudanças no design de obstáculos.

Além da mecânica clássica de pulo, foram adicionados:

- **Power-ups** (invencibilidade, slow motion, duplo salto, pontuação dobrada) com ícones animados e timers exibidos na tela.
- **Linha de chegada** dourada aparecendo aos 10 000 pontos, encerrando o desafio com tela de vitória.
- **Sistema de partículas** para salto (`spawnDust()`), coleta (`spawnPowerUpParticles()`) e colisão (`spawnCollisionParticles()`).
- **Squash & stretch** aplicado ao player para dar peso e impacto ao pulo.
- **Menu, Pausa e Tela Final** construídos em p5.js, com botões interativos via `mousePressed()` e checagem de retângulo (`pauseButtonRect`).

São suportados:

- **Teclado**: setas para mover/pular; M para reiniciar; E para pausar/resumir.
- **Mouse/Toque**: botões virtuais desenhados em `drawControls()` e lidos por `getControlInput()`, permitindo compatibilidade desktop/mobile.
- **Redimensionamento**: `windowResized()` recria canvas e elementos, mantendo proporção e definindo `groundY`.

Para garantir 60 FPS estáveis:

- Remoção em loop reverso de arrays;
- Dessinhos complexos minimizados dentro de laços críticos;
- Pixel density ajustado uma única vez;
- Estrutura clara de estados evita execuções desnecessárias fora de `playing`.

O código segue convenções BEM nos nomes de variáveis e classes, e conta com comentários explicativos, facilitando extensões futuras, como inclusão de áudio em `preload()`.

3 CONSIDERAÇÕES FINAIS

O site Agrinho alcançou seus objetivos de oferecer uma experiência interativa, inclusiva e informativa, celebrando a conexão entre o campo e a cidade. Usuários de diferentes perfis podem navegar, acessar conteúdo diversificado e interagir por meio do quiz e do minijogo.

Como próximos passos, recomenda-se aprimorar ainda mais a acessibilidade com ARIA avançado, adicionar animações e transições via JavaScript e explorar internacionalização do conteúdo. Em termos de design, seria interessante implementar um sistema de dark mode automático e disponibilizar versão multilíngue.

O desenvolvimento deste projeto possibilitou a aquisição de novas competências em HTML e CSS avançado, reforçando a capacidade de criar plataformas interativas e acessíveis. Essa experiência servirá de base para iniciativas futuras de maior complexidade.

Observação: Capa, margens, espaçamento e fonte, conforme ABNT.
FORMATO DO DOCUMENTO – PDF.