Documento de Design de Jogo (DDJ): Code Fight - A Batalha dos Mestres

1. Resumo Executivo

Code Fight é um jogo de luta 2D, em estilo **pixel art**, criado para a web. O jogo é uma homenagem aos clássicos do gênero, com uma temática única onde professores de programação se enfrentam usando suas habilidades e conhecimentos. O projeto será desenvolvido com **HTML, CSS e JavaScript**.

---

2. Conceito do Jogo

- **Gênero:** Jogo de Luta 2D.
- **Plataforma:** Navegadores Web.
- **Modo de Jogo:** PvP (Jogador vs. Jogador).
- **Estilo Visual:** Pixel art.
- **Objetivo:** Derrotar o oponente zerando sua barra de vida antes que o tempo acabe.

---

3. Personagens

Cada personagem é baseado em um professor, com habilidades que refletem suas personalidades e disciplinas.

- **Professor Elison - O Adivinho**
  - **Estilo de Luta:** Ágil e imprevisível.
  - **Habilidade Especial: Teleporte Aleatório**. A cada 10 golpes bem-sucedidos em sequência, ele tem 50% de chance de se teletransportar para as costas do oponente, confundindo-o e abrindo uma nova janela de ataque.
  - **Cenário:** Uma sala de aula cheia de quadros de programação, com códigos complexos e equações matemáticas ao fundo.
- **Professor Laerte - O Cultivador**
  - **Estilo de Luta:** Lento, mas com golpes pesados e de alto impacto.
  - **Habilidade Especial: Golpe da Enxada**. Um ataque de área que causa dano em múltiplos pontos no oponente. Seu golpe final pode ser uma "colheita" de dano maciço.
  - **Cenário:** Uma horta pixelizada, simbolizando o "cultivo" do conhecimento, com vegetais e flores em blocos.
- **Professor Roberto - O Sargento do Código**
  - **Estilo de Luta:** Preciso e de longo alcance.
  - **Habilidade Especial: Código Travador**. Ele arremessa linhas de código que imobilizam o oponente por um breve período, deixando-o vulnerável a ataques.
  - **Cenário:** Um escritório com pilhas de livros, um computador antigo e uma cafeteira, representando a disciplina e a dedicação.

- **Professor Barba - O Filósofo**
    - **Estilo de Luta:** Calmo e focado em defesa e contra-ataques.
    - **Habilidade Especial: Meditação Imóvel**. Ele entra em um estado de meditação por 2 segundos, tornando-se invulnerável a qualquer dano. O próximo ataque desferido após a meditação causa o dobro de dano.
    - **Cenário:** Um campo florido e um lago tranquilo, refletindo a paz e o equilíbrio da filosofia.

---

4. Mecânicas de Jogo

- **Barras de Vida:** As barras de vida dos dois jogadores serão exibidas no topo da tela. A luta termina quando a vida de um dos jogadores chega a zero.
- **Cronômetro:** Um cronômetro de 90 segundos será exibido. Se o tempo acabar, o jogador com a maior porcentagem de vida vence.
- **Movimentação:** Os personagens poderão se mover para frente e para trás, pular e se agachar.
- **Ataques:** Cada personagem terá ataques básicos (soco e chute) e uma habilidade especial.

---

5. Controles (Teclado)

- **Jogador 1:**
    - **Mover:** `A` (para trás) e `D` (para frente).
    - **Pular:** `W`.
    - **Socar:** `T`.
    - **Chutar:** `Y`.
    - **Habilidade Especial:** `U`.
- **Jogador 2:**
    - **Mover:** Seta Esquerda (para trás) e Seta Direita (para frente).
    - **Pular:** Seta para cima.
    - **Socar:** Tecla numérica `1`.
    - **Chutar:** Tecla numérica `2`.
    - **Habilidade Especial:** Tecla numérica `3`.

---

6. Lógica de Programação Detalhada (JavaScript)

A espinha dorsal do jogo será um **loop de jogo contínuo** (`requestAnimationFrame`) que atualiza o estado do jogo e redesenha a tela a cada frame.

- **1. Estrutura de Classes:**

    - Uma classe base `Personagem` terá propriedades como `x`, `y`, `vida`, `velocidade`, `largura`, `altura` e métodos básicos como `mover()`, `pular()`, `atacar()`.
    - Classes específicas (`ProfessorElison`, `ProfessorLaerte`, etc.) herdarão de `Personagem`, sobrescrevendo o método `atacar()` ou adicionando um método para a habilidade especial.
- **2. O Loop de Jogo:**
    - **Atualização de Posição:** Um loop `for` irá iterar por todos os objetos (`personagens`, `projéteis`). A lógica de gravidade e as variáveis de movimento serão atualizadas a cada frame.
    - **Detecção de Colisão:** Usaremos uma função `colide(obj1, obj2)` que verifica se as caixas de colisão (hitboxes) dos objetos se sobrepõem. Se um ataque colidir, a vida do oponente será reduzida.
    - **Renderização:** O `canvas` será limpo e todos os elementos serão redesenhados com base nas suas novas posições e estados.
- **3. Gerenciamento de Inputs:**
    - Eventos de teclado (`keydown` e `keyup`) irão ativar variáveis booleanas (`isMovingRight`, `isJumping`, etc.). O loop de jogo usará essas variáveis para mover os personagens continuamente enquanto as teclas estiverem pressionadas.
- **4. Sistema de Áudio e Mistérios:**
    - **Falas (MP3):** A classe `Audio()` do JavaScript será usada para reproduzir falas gravadas antes de cada luta.
    - **O Mistério do Renê:** Uma função com um **`Math.random()`** será chamada no final de cada partida. Se o número gerado for menor que um valor baixo (por exemplo, `0.01`), a frase "Fui eu não" será exibida na tela.

---

7. Artes e Animações

- O jogo será feito em **pixel art**, com personagens e cenários desenhados usando CSS.
- As animações (andar, pular, atacar) serão criadas com **CSS keyframes** para cada estado do personagem, dando um estilo de movimento autêntico e retrô.