PLANO DE TESTES

1. INTRODUÇÃO

Este documento apresenta o plano de testes para o jogo Eco Runner, que será desenvolvido em Python com Pygame e integrado a um banco de dados PostgreSQL. Como o jogo ainda não foi implementado nesta primeira entrega, o plano tem caráter introdutório e descreve o que será testado (requisitos funcionais) e o que ficará fora do escopo (requisitos não funcionais) nesta fase inicial.

2. OBJETIVO DOS TESTES

Planejar os testes que deverão garantir o correto funcionamento das principais funcionalidades do jogo, de acordo com os requisitos funcionais definidos, delimitando também os pontos que não serão considerados nesta etapa.

3. ABORDAGENS DE TESTES UTILIZADOS

TIPOS DE TESTE	VERIFICAÇÃO
Teste Funcional	Validar login, coleta de itens, bloqueio de itens errados e salvamento de progresso.
Teste de Integração	Garantir comunicação correta entre o jogo e o banco de dados (login, salvamento e carregamento).
Teste de Usabilidade	Confirmar alertas adequados ao jogador em erros de classificação de itens.
Teste de Desempenho	Medir o tempo de resposta de comandos em execução.

4. ESCOPO DOS TESTES

4.1 Dentro do escopo

ID	Funcionalidades	Objetivo do teste	Método de teste

RF01	*Login* de Usuário	Validar autenticação de credenciais válidas, carregamento de progresso e início do jogo	Particionamento de Equivalência (credenciais válidas)
RF02	Cadastrar Jogador	Verificar criação de novo jogador, respeitando regras de tamanho de nickname e senha	Análise de Valor Limite (mín. e máx. de caracteres) + Particionamento de Equivalência
RF03	Coletar materiais recicláveis (Fase 1)	Garantir que o sistema registre corretamente os 20 itens e permita o avanço para a próxima fase apenas quando todos forem coletados	Grafo de Causa-Efeito (condição: coletar 20 itens, efeito: transição de fase)
RF04	Classificar Itens Recicláveis (Fase 2)	Validar que: ● O sistema aceite apenas quando o item for colocado na lixeira correta. ● O sistema rejeite e devolva o item em caso de escolha incorreta, exibindo o feedback visual adequado. ● O jogador avança para a fase final apenas após classificar corretamente os 20 itens.	Tabela de Decisão (para verificar todas as combinações entre item e lixeira) + Particionamento de Equivalência (casos válidos e inválidos de classificação)
RF05	Combater Yluh	Confirmar funcionamento do combate, redução de vidas e finalização da fase	Grafo de Causa-Efeito (eventos de ataque e mudança de estado de vida)

RF06	Gerenciar jogo: pausar, salvar e carregar progresso.	Garantir que o progresso (fase, vidas) é salvo corretamente e carregado na próxima sessão	Teste Combinacional (diferentes estados de jogo X ações: pausar, salvar, carregar)

4.2 Fora do Escopo

ID	**Funcionalidades**	**Objetivo do teste**	**Método de teste**
RNF01	Tempo de Resposta	Garantir que as respostas visuais ocorram em ≤ 200ms	Teste de Desempenho (não realizado nesta fase)
RNF02	Consistência e Legibilidade da Interface	Validar padrão visual e legibilidade em todas as telas	Inspeção de Usabilidade / Heurísticas de Nielsen (não realizado nesta fase)
RNF03	Compatibilidade Windows/Linux	Verificar execução correta em diferentes sistemas operacionais	Teste de Instalação e Execução em Ambientes Diferentes (não realizado nesta fase)
RNF04	Integridade e Resiliência dos Dados Salvos	Validar que os dados salvos são recuperados corretamente mesmo após falhas	Error Guessing + Teste de Falha Forçada (não realizado nesta fase)