

Plano de Testes

Desenvolvimento de um Jogo em Unity abordando a Sustentabilidade

Alunos:

Bryan Souza da Silva – 113341
Lucca Antonio Menegatti – 113145

Orientador:

Prof. Me. Diego Henrique Negretto

Maio/2025

Histórico de Revisões

Data	Versão	Descrição	Autores
<30/maio/25>	<1.0>	Release inicial	Bryan/Lucca

Índice

1. Estratégia de Testes	4
1.1 Testes Estruturais (Testes de Caixa Branca)	4
1.2 Testes Funcionais (Testes de Caixa Preta)	4
2. Estratégias de Testes Específicos	5
2.1 Testes de Integração	5
2.2 Testes de Sistema	5
2.3 Testes de Aceitação	5
2.4 Testes de Regressão	6
3. Plano de Testes para os Requisitos Funcionais Especificados	7
4. Plano de Testes e Critérios de Aceitação para os Requisitos Não-Funcionais	8
Considerações Finais	8

1. Estratégia de Testes

1.1 Testes Estruturais (Testes de Caixa Branca)

- **Objetivo:** Garantir que o código-fonte implementado funciona conforme esperado, validando a lógica das funções, fluxos de decisão, laços e estrutura de dados.
- **Aplicação:**
 - Testar scripts C# desenvolvidos no Unity;
 - Validar funções responsáveis por:
 - Controle das escolhas e consequências do jogador;
 - Cálculo do feedback imediato;
 - Navegação entre cenas/fases.
- **Ferramentas:** Unity Test Framework (Test Runner), NUnit.

1.2 Testes Funcionais (Testes de Caixa Preta)

- **Objetivo:** Validar se as funcionalidades implementadas no jogo atendem aos requisitos especificados, sem considerar a estrutura interna do código.
- **Aplicação:**
 - Testar se o jogador consegue:
 - Tomar decisões e ver os efeitos imediatos;
 - Cumprir missões baseadas nos ODS (2, 11 e 12);
 - Receber feedback instantâneo após cada decisão.
 - Serão conduzidos sem conhecimento do código, simulando a interação de um usuário com o jogo.

- **Técnica:** Testes exploratórios baseados em casos de uso.

2. Estratégias de Testes Específicos

2.1 Testes de Integração

- **Objetivo:** Garantir que os diferentes módulos do jogo funcionem corretamente em conjunto.
- **Aplicação:**
 - Integração entre:
 - Scripts de decisão e sistema de feedback;
 - Banco de dados e sistema de missões;
 - Sistema de navegação entre fases e progressão do jogo.
- **Ferramentas:** Unity Test Framework, depuradores integrados da IDE.

2.2 Testes de Sistema

- **Objetivo:** Verificar o comportamento do jogo como um todo, avaliando se todos os requisitos foram implementados corretamente.
- **Aplicação:**
 - Testar o fluxo completo: iniciar jogo → jogar → pontuar → completar fase → feedback;
 - Validar todas as possíveis trajetórias de decisões dos jogadores.
- **Cenários:** Usuário concluindo todas as fases com decisões corretas ou incorretas.

2.3 Testes de Aceitação

- **Objetivo:** Confirmar que o produto atende aos requisitos e expectativas dos usuários finais.

- **Aplicação:**
 - Realizar sessões de teste com potenciais usuários (alunos, professores);
 - Observar se:
 - O jogo é educativo e divertido;
 - As mensagens sobre sustentabilidade são compreendidas;
 - O design da interface é acessível.
- **Técnica:** Entrevistas, questionários, testes observacionais.

2.4 Testes de Regressão

- **Necessário?** Sim
- **Objetivo:** Garantir que alterações ou correções não afetem funcionalidades previamente implementadas.
- **Aplicação:**
 - Sempre que uma nova fase ou mecânica for implementada;
 - Atualização de gráficos ou sistema de feedback;
 - Importantes em fases finais, durante ajustes finos e correções de bugs.
- **Ferramentas:** Automação possível via Unity Test Runner.

3. Plano de Testes para os Requisitos Funcionais Especificados

Requisito Funcional	Tipo de Teste	Caso de Teste	Resultado Esperado
Iniciar fase	Funcional	Verificar se ao clicar em "Iniciar", a fase carrega corretamente	A fase é carregada e o jogo inicia
Pontuar	Funcional	Realizar ações sustentáveis no jogo e verificar se pontos são atribuídos	Pontuação é incrementada corretamente
Dar feedback	Funcional	Completar uma fase e verificar a tela de feedback educativo	Feedback claro e relacionado à sustentabilidade
Salvar progresso	Funcional	Salvar o progresso e reiniciar o jogo	Progresso anterior é mantido

4. Plano de Testes e Critérios de Aceitação para os Requisitos Não-Funcionais

Requisito Não-Funcional	Tipo de Teste	Caso de Teste	Critério de Aceitação
Tempo de carregamento	Desempenho	Medir tempo de carregamento de uma fase	Carregamento inferior a 5 segundos
Usabilidade	Aceitação/Heurístico	Realizar testes com usuários e observar navegação	Jogadores entendem como jogar sem instruções complexas
Compatibilidade	Sistema	Testar em diferentes dispositivos	Funcional em todas as plataformas definidas
Estabilidade do jogo	Estresse/Regressão	Jogar por longos períodos (ex: 30 minutos)	Jogo não apresenta travamentos ou crashes
Clareza da proposta educativa	Aceitação	Avaliação por usuários e/ou especialistas	Jogador compreende a mensagem de sustentabilidade

Considerações Finais

Este planejamento contempla todas as etapas de testes necessárias para assegurar a qualidade do jogo, desde o desenvolvimento até a validação com usuários finais, garantindo que ele cumpra seu objetivo: **explorar a gamificação como estratégia para abordar a sustentabilidade**.