

Plano de teste - Zombie Bug Hunter

1. Introdução

Zombie Bug Hunter é um jogo que tem como principal objetivo sobreviver a um apocalipse zumbi, matando os monstros e passando de fases. Os controles do jogo são WASD, ou as setas. Para mirar e atirar, o jogador deverá usar o mouse, rotacionando o eixo do personagem e clicando para atirar as balas que deverão ser vermelhas.

- a. O ambiente de teste para esta aplicação é o:
<https://jogajunto.itch.io/zombiebughunter>

2. Arquitetura

O jogo é desenvolvido na plataforma Unity com a linguagem C#, sem necessidade de comunicação externa por meio de API's

3. Ferramentas de Teste:

Serão utilizadas as seguintes ferramentas para testes:

Tipo de teste	Ferramenta
Unitários	xUnit
API	-
Automatização	Pyautogui

4. Estratégia de Teste:

- Escopo de testes:

Todas as funcionalidades descritas no campo de Cenário de Testes serão executadas pela equipe de QA.

- Testes unitários: Serão executados pelos desenvolvedores adotando a estratégia do TDD, com uma cobertura de código maior ou igual a 75%;
- Testes Manuais: Serão executados pela equipe de QA's e deverão ser executados todos os casos de testes descritos no cenário de testes;
- Testes Automatizados: Serão executados pela equipe de QA's e deverá ser feito para testar os tiros.

Deverá ser lançada uma versão de Beta para 20 usuários testes.

5. Classificação dos Bugs:

- Critical/Blocker:

Bugs que não permitem o pleno funcionamento de requisitos funcionais. Sendo impraticável, ir para produção.

- Grave

Bugs em funcionalidades que não impliquem em seu não funcionamento pleno. Em que a funcionalidade não funciona 100% como deveria

- Moderada

Bugs que não atrapalham a funcionalidade do serviço.

- Leve

Pequenos erros visuais ou ortográficos

6. Cenários de Testes:

ID	Funcionalidade	Caso de teste	verficiações
1			
2			
3			

7. Metricas de qualidade

Metricas para lançamento do jogo:

a. Taxa de cobertura do código:

Deverão ser cobertos 75% dos componentes da aplicação

b. Taxa de aceitação

Não serão aceitos bugs classificados como Blocker/critical ou moderados.

c. Taxa de aprovação dos usuários:

Para lançamento, todos os usuários da versão beta deverão marcar o jogo como satisfatório.

d. Tempo de resposta para correção dos bugs