

MODELO PARA O DOCUMENTO DE REQUISITOS DO PROJETO

RF 01		
Nome:	Carregar fase	
Descrição:	O sistema deve carregar um nível contendo grade (largura × altura), matriz de peças, posições de fonte(s) e casa(s), e metas (todas as casas abastecidas).	
Atores:	Sistema,Jogador	
Prioridade:	Essencial	
Entradas e pré-condições:	O jogador inicia uma fase; sistema deve ter configuração da fase disponível.	
Saídas e pós-condições:	Mapa renderizado em ≤ 1s; contador de casas faltantes exibido.	
Fluxos de eventos		
Fluxo principal:	<div><div>1. Jogador Inicia a fase.</div><div>2. O sistema carrega a grade, peças, fontes e casas.</div><div>3. Sistema renderiza a fase em até 1 segundo.</div><div>4. Sistema mostra contador de casas a abastecer.</div></div>	
Fluxo secundário 1:		

RF 02		
Nome:	Iniciar a Fase	
Descrição:	O sistema deve iniciar o nível	
Atores:	Sistema,Jogador	
Prioridade:	Essencial	
Entradas e pré-condições:	O jogador seleciona uma fase disponível; sistema tem a configuração carregada.	
Saídas e pós-condições:	Fase iniciada; timer e contador de objetivos começam em 0.	
Fluxos de eventos		
Fluxo principal:	<div>1. O jogador seleciona iniciar a fase.</div> <div>2. Sistema inicializa variáveis de controle (tempo, cliques, status).</div> <div>3. Sistema renderiza a fase em até 1s.</div>	
Fluxo secundário 1:	Fase não encontrada ou não disponível → Sistema exibe mensagem de erro.	

RF 03		
Nome:	Menu de Pausar a Fase	
Descrição:	O jogador pode pausar a fase em andamento.	
Atores:	Jogador	
Prioridade:	Importante	
Entradas e pré-condições:	Fase em andamento.	
Saídas e pós-condições:	Jogo suspenso, menu de pausa exibido.	
Fluxos de eventos		
Fluxo principal:	<div>1. Jogador solicita pausa.</div> <div>2. Sistema congela temporizadores e interações.</div> <div>3. Sistema exibe menu de pausa com opções.</div>	
Fluxo secundário 1:	Fase não encontrada ou não disponível → Sistema exibe mensagem de erro.	

RF 04	
Nome:	Mover peças
Descrição:	O jogador pode arrastar e posicionar peças no grid.
Atores:	Jogador, Sistema
Prioridade:	Essencial
Entradas e pré-condições:	Fase carregada.
Saídas e pós-condições:	Peças posicionadas no grid.
Fluxos de eventos	
Fluxo principal:	<ol style="list-style-type: none"> 1. Jogador arrasta peça. 2. Sistema atualiza posição no grid em tempo real. 3. O sistema realiza a validação da conexão.
Fluxo secundário 1:	Se a validação da conexão for válida, finaliza a fase.

RF 05		
Nome:	Resetar a Fase	
Descrição:	O jogador pode reiniciar a fase a qualquer momento.	
Atores:	Jogador	
Prioridade:	Importante	
Entradas e pré-condições:	Fase em andamento.	
Saídas e pós-condições:	Fase reiniciada do zero.	
Fluxos de eventos		
Fluxo principal:	<div>1. Jogador seleciona “Resetar”.</div> <div>2. Sistema reinicia variáveis e recarrega layout inicial.</div>	
Fluxo secundário 1:	Fase não encontrada ou não disponível → Sistema exibe mensagem de erro.	

RF 06		
Nome:	Validação de conexões	
Descrição:	Duas peças vizinhas se conectam se seus lados adjacentes possuírem bocas compatíveis.	
Atores:	Sistema	
Prioridade:	Essencial	
Entradas e pré-condições:	Grade carregada com peças rotacionadas.	
Saídas e pós-condições:	O sistema registra conexões válidas conforme rotação atual.	
Fluxos de eventos		
Fluxo principal:	<div><div>1. O sistema analisa vizinhos (N, S, L, O).</div><div>2. Verifica se os lados adjacentes possuem bocas compatíveis.</div><div>3. Marca conexões válidas para simulação de fluxo.</div></div>	
Fluxo secundário 1:		

RF 07		
Nome:	Finalização da fase	
Descrição:	Se todas as casas foram abastecidas, exibir tela de sucesso com tempo, cliques, vazamentos e pontuação.	
Atores:	Jogador, Sistema	
Prioridade:	Essencial	
Entradas e pré-condições:		Todas as casas abastecidas.
Saídas e pós-condições:		Tela de sucesso exibida; opções de próxima fase e reinício.
Fluxos de eventos		
Fluxo principal:	<ol style="list-style-type: none"> 1. O sistema detecta todas as casas abastecidas. 2. Exibe tela de sucesso com estatísticas. 3. Bloqueia interações. 4. Oferece opções de avançar ou reiniciar. 	
Fluxo secundário 1:		

RF 08		
Nome:	Progressão de fases	
Descrição:	Completar fase N desbloqueia automaticamente a fase N+1.	
Atores:	Jogador, Sistema	
Prioridade:	Essencial	
Entradas e pré-condições:		Jogador conclui a fase atual.
Saídas e pós-condições:		Próxima fase desbloqueada; progresso gravado no LocalStorage (ou banco de dados, se logado).
Fluxos de eventos		
Fluxo principal:	<ol style="list-style-type: none"> 1. Jogador conclui fase N 2. Sistema desbloqueia fase N+1 3. Sistema grava o estado no LocalStorage ou BD 4. Jogador pode iniciar a fase N+1 	
Fluxo secundário 1:		

RF 09		
Nome:	Menu e seleção de fase	
Descrição:	Exibir um menu principal com identificação da fase atual, lista de fases (bloqueadas/desbloqueadas), indicador de progresso (ex.: ícones, estrelas ou %), e ação de “Iniciar/Continuar”.	
Atores:	Jogador	
Prioridade:	Importante	
Entradas e pré-condições:		Jogo iniciado; estado de progresso disponível.
Saídas e pós-condições:		Fase selecionada carregada (integrada com RF-01); visão de progresso atualizada.
Fluxos de eventos		
Fluxo principal:	<ol style="list-style-type: none"> 1. Jogador abre o menu principal 2. Sistema exibe fase atual, lista e status(bloqueada/desbloqueada). 3. Jogador seleciona “Iniciar/Continuar” ou escolhe uma fase desbloqueada 4. Sistema carrega a fase escolhida (RF-01). 	
Fluxo secundário 1:	<ol style="list-style-type: none"> 1. Jogador tenta abrir fase bloqueada 2. Sistema informa requisito e permanece no menu 	

RF 10		
Nome:	Sistema de pontuação	
Descrição:	A pontuação base = $1000 - (\text{tempo}5) - (\text{cliques}2) - (\text{vazamentos} \times 50)$, mínimo 0.	
Atores:	Sistema	
Prioridade:	Importante	
Entradas e pré-condições:		Simulação concluída.
Saídas e pós-condições:		Pontuação calculada e exibida.
Fluxos de eventos		
Fluxo principal:	<ol style="list-style-type: none"> 1. Sistema coleta dados (tempo, cliques, vazamentos). 2. Calcula pontuação. 3. Exibe breakdown na tela de resultado. 	
Fluxo secundário 1:		

RNF 01	
Nome:	Performance e Responsividade
Descrição:	O sistema deve responder instantaneamente às ações do jogador (arrastar e soltar), com operações de grid executadas em tempo real sem atraso perceptível. O jogo deve manter uma taxa de 60 quadros por segundo (FPS) durante todas as interações.
Atores:	
Prioridade:	Alta
Entradas e pré-condições:	O jogador está interagindo com os elementos do jogo.
Saídas e pós-condições:	A experiência de jogo é fluida e sem interrupções ou lentidão.
Fluxos de eventos	
Fluxo principal:	1. ...
Fluxo secundário 1:	

RNF 02	
Nome:	Usabilidade e Interface
Descrição:	A interface deve ser simples e intuitiva, permitindo que o jogador entenda as mecânicas sem a necessidade de um tutorial. O feedback visual (como o posicionamento no grid) deve ser claro e preciso. A área de clique das peças (64x64 pixels) deve facilitar a seleção.
Atores:	Jogador
Prioridade:	Alta
Entradas e pré-condições:	O jogador está na tela principal do jogo.
Saídas e pós-condições:	O jogador consegue interagir com o jogo de forma eficiente e sem frustração.
Fluxos de eventos	
Fluxo principal:	1. ...
Fluxo secundário 1:	

RNF 03		
Nome:	Compatibilidade e Portabilidade	
Descrição:	O projeto deve ser compatível com a versão 2024.13.1.193 ou superior do GameMaker Studio e ser projetado para funcionar em múltiplas plataformas (Windows, HTML5, Android, iOS). Os recursos gráficos devem utilizar formatos padrão (ex: PNG) para garantir a portabilidade.	
Atores:	Desenvolvedor	
Prioridade:	Média	
Entradas e pré-condições:		O projeto é compilado para uma plataforma de destino.
Saídas e pós-condições:		O jogo executa corretamente na plataforma de destino sem a necessidade de grandes alterações no código-fonte.
Fluxos de eventos		
Fluxo principal:	1. ...	
Fluxo secundário 1:		

RNF 04	
Nome:	Manutenibilidade e Extensibilidade
Descrição:	O código-fonte deve ser modularizado, com objetos separados para as principais funcionalidades (grid, peças, etc.). O sistema deve permitir que características como o tamanho do grid (atualmente 8x8) e a quantidade de obstáculos sejam facilmente configuráveis através de variáveis.
Atores:	Desenvolvedor
Prioridade:	Média
Entradas e pré-condições:	Um desenvolvedor precisa alterar ou adicionar uma funcionalidade.
Saídas e pós-condições:	A modificação pode ser implementada com baixo esforço e sem causar efeitos colaterais em outras partes do sistema.
Fluxos de eventos	
Fluxo principal:	1. ...
Fluxo secundário 1:	

RNF 05	
Nome:	Confiabilidade e Robustez
Descrição:	O sistema deve ser robusto, tratando possíveis erros (ex: um objeto controlador não encontrado) sem travar. Deve sempre validar os limites do grid para evitar acessos a posições inválidas e garantir que o estado do grid se mantenha sempre consistente.
Atores:	Sistema
Prioridade:	Alta
Entradas e pré-condições:	Ocorrem operações no grid ou eventos inesperados durante a execução do jogo.
Saídas e pós-condições:	O jogo continua a funcionar de forma estável e previsível.
Fluxos de eventos	
Fluxo principal:	1. ...
Fluxo secundário 1:	

