**Introdução:**

Este documento apresenta o escopo do projeto de desenvolvimento do jogo "Ticket to Ride". Nosso objetivo é criar uma versão digital do clássico jogo de tabuleiro, respeitando suas regras originais e proporcionando uma experiência interativa, intuitiva e agradável para os jogadores. A aplicação deverá permitir partidas para múltiplos jogadores, gerenciar a distribuição de cartas, a construção de rotas e a pontuação de forma automática. Além disso, o projeto envolverá a definição de requisitos funcionais e não funcionais, a divisão de tarefas entre os integrantes do grupo e a definição das tecnologias que serão utilizadas.

**Requisitos funcionais:**

**RF1. Configurar partida:**

O sistema deve permitir que o usuário configure aspectos da partida como quantidade de jogadores, cores, número de IAs e tempo.

**RF2. Iniciar Partida:**

O sistema deve permitir iniciar uma nova partida com 2 a 5 jogadores (IA ou não).

**RF03. Distribuir Cartas:** Ao começar uma partida, o sistema deve distribuir as cartas de vagão e cartas de destino iniciais para cada jogador.

**RF04. Exibir Interface de Jogo:** O sistema deve apresentar um mapa de jogo com as cidades e as rotas possíveis para os jogadores, além de 2 a 5 indicadores de jogadores (contendo as respectivas cores, pontos e quantidade de cartas na mão).

**RF05. Jogar Turno:** O sistema deve permitir que cada jogador escolha uma de três ações em seu turno:

* Comprar cartas de vagão: O jogador deve poder comprar 2 cartas escolhidas entre abertas ou fechadas do baralho, exceto ao comprar uma carta coringa aberta.
* Construir uma rota usando cartas: O jogador deve poder usar suas cartas de vagão para conquistar rotas no tabuleiro.
* Comprar novas cartas de destino: O jogador deve poder comprar 3 cartas de destino adicionais, devendo obrigatoriamente ficar com pelo menos uma delas.

**RF06. Cartas Destino Responsivos:** O jogo deve destacar os vértices ao jogador, quando este passar o mouse em cima da carta destino.

**RF07. Marcar Pontuação:** O sistema deve conceder pontos ao jogador sempre que ele construir uma rota (com base na tabela de pontos).

**RF08. Verificar Estado das Cartas:**

* Caso as cartas do baralho de compra acabem, o sistema deve embaralhar o descarte criando um novo baralho de compra.
* Caso exista 3 coringas abertos, o sistema deve recolher todas as cartas abertas e gerar 5 novas.
* Após o jogador terminar seu turno, se ele comprou alguma carta aberta, o sistema deve repor as cartas abertas de tal forma que 5 cartas sempre fiquem exibidas.

**RF09. Finalizar Partida:** O sistema deve finalizar a partida quando um jogador ficar com 2 trens ou menos e todos os jogadores subsequentes finalizarem seus turnos.

**RF10. Declarar Vencedor:** Ao final do jogo o sistema deve calcular a pontuação final, considerando:

* Rotas construídas ao longo do jogo.
* Cartas de destino completadas
* Cartas de destino não completadas (penalização).
* A maior rota construída.

**RF11. Salvamento de Partidas:**  
O sistema deve permitir que o usuário salve o estado corrente da partida a qualquer momento.

**RF12. Carregamento de Partidas:**

O sistema deve permitir que o usuário carregue uma partida previamente salva.

**RF13. Interface Responsiva:**  
 O sistema deve permitir que o usuário escolha diferentes tamanhos de tela (16:9 1920x1080p, 5:4 1280x1024p, 4:3 1024x768p).

**Requisitos não funcionais:**

**RNF01. Usabilidade:**O sistema deve ser intuitivo e fácil de aprender, com feedback visual claro para as ações do jogador.

**RNF02. Desempenho:**O tempo de resposta entre uma ação do jogador e a atualização da interface deve ser inferior a 1 segundo.

**RNF03. Confiabilidade:** O sistema não deve travar ou perder dados no meio do jogo.

**RNF04. Manutenibilidade:**O código-fonte deve ser modular, organizado e documentado para facilitar futuras melhorias ou correções.

**RNF05. Portabilidade:**O sistema deve rodar em diferentes sistemas operacionais (Windows, MacOS, Linux).

**Ambiente de Desenvolvimento:**

OS: Windows 10 Professional

Linguagem de programação: Python 3.11

Lista de bibliotecas:

* Pygame v2.6
* Biblioteca genérica grafos v
* Random v