**Especificação de Requisitos do Jogo Pega em 6**

**Projeto Pega em 6**

**Especificação de Requisitos de Software**

Versão 1.0

22/06/2021

|  |  |  |  |
| --- | --- | --- | --- |
| **Versão** | **Autor(es)** | **Data** | **Ação** |
| 1.0 | Eduardo Willwock Lussi  Igor Perazzoli | 22/06/2021 | Estabelecimento dos requisitos |

**Conteúdo:**

1. Introdução;
2. Visão geral;
3. Requisitos de software.

**1 Introdução**

**Objetivo:** desenvolvimento de um programa que suporte a disputa de partidas do jogo Pega em 6 entre vários jogadores em uma mesma máquina.

**Referências:**

[1] <<https://papergames.com.br/pega-em-6/>> (regras do jogo).

**2 Visão Geral**

**Arquitetura do programa:** programa orientado a objetos, *stand-alone*.

**Premissas de desenvolvimento:**

* O programa deve apresentar uma interface gráfica bidimensional;
* O programa deve ser implementado em Python, devendo executar em qualquer plataforma que disponha da linguagem interpretada Python.
* O programa deve dar suporte à interação local entre os usuários.

**3 Requisitos de Software**

**3.1 Requisitos Funcionais**

**Requisito funcional 1** **- Iniciar:** o programa deve apresentar a opção de menu “iniciar” para o início de uma nova partida, operação em que é definido a quantidade de jogadores que participarão da partida e o jogador que a inicia;

**Requisito funcional 2 - Reiniciar:** o programa deve apresentar a opção de menu “reiniciar” para o início de uma nova partida, considerando as seguintes restrições:

• A operação mantém a quantidade de jogadores já definida e estabelece o jogador que inicia;

• Caso haja partida em andamento, o programa deve solicitar autorização para sua interrupção – neste caso, apenas com a autorização do usuário que solicitou a operação pode prosseguir.

**Requisito funcional 3 - Escolha da quantidade de jogadores:** o programa deve dar a opção para escolha entre 2 e 10 jogadores, conforme as regras de jogo (referência 1).

**Requisito funcional 4 - Distribuição das cartas:** no início de cada rodada, o programa deve distribuir as cartas devidamente embaralhadas na seguinte ordem: primeiro, quatro cartas para a mesa e depois, dez cartas para cada jogador;

**Requisito funcional 5 - Procedimento de escolha da carta:** o programa deve suportar o procedimento de escolha da carta dos usuários por meio de click de botão esquerdo do mouse sobre a carta escolhida da sua mão;

**Requisito funcional 6 - Ocultação das cartas entre os jogadores:** o programa deve garantir que os jogadores não vejam as cartas uns dos outros;

**Requisito funcional 7 - Procedimento de escolha de fileira:** quando a carta escolhida pelo jogador no procedimento de escolha de carta não for maior que a maior carta de nenhuma fileira, ele deve escolher uma fileira para ser substituída pela carta escolhida, enquanto a soma dos pontos das cartas da fileira indicada será somada a sua pontuação atual;

**Requisito funcional 8 - Comunicação do estado da partida:** após todos os jogadores terem escolhido suas cartas, o novo estado da partida deve ser exibido na interface, atualizando conforme as regras a serem aplicadas;

**Requisito funcional 9 - Encerramento da rodada**: ao término de uma rodada (quando as cartas dos jogadores se esgotarem), o programa deve analisar o estado atual da partida para definir o que acontecerá a seguir, isso é feito com base nas duas possibilidades abaixo:

* Caso algum dos jogadores tenha alcançado pontuação igual ou superior a 66 pontos, é decretado o término da partida e os resultados devem ser exibidos na interface;
* Caso nenhum jogador alcance a marca dos 66 pontos, uma nova rodada é iniciada.

**3.2 Requisitos Não Funcionais**

**Requisito não funcional 1 - Especificação de projeto:** além do código Python, deve ser produzida especificação de projeto baseada em UML, segunda versão;

**Requisito não funcional 2 - Interface gráfica para usuário:** o programa deverá ter interface gráfica única, partilhada pelos usuários;

**Requisito não funcional 3 - Tecnologia de interface gráfica para usuário:** A interface gráfica deve ser baseada em Tkinter;