

# Projeto PEG-4

## Especificação de Requisitos de Software

Versão 0.4

14/03/2022

Versão	Autor(es)	Data	Ação
0.1	Arthur de Sousa Costa, Arthur Medeiros Machado de Souza e Felipe Valentin Nascimento	22/11/2021	Produção da especificação de requisitos, implementação da modelagem e interface gráfica
0.2	Arthur de Sousa Costa, Arthur Medeiros Machado de Souza e Felipe Valentin Nascimento	01/02/2022	Atualização dos diagramas de classes e de casos de uso. Produção dos diagramas de atividades e sequência. Seção “Definições e Regras” adicionada no documento de Especificação de Requisitos de Software.
0.3	Arthur de Sousa Costa, Arthur Medeiros Machado de Souza e Felipe Valentin Nascimento	02/03/2022	Refinamento dos diagramas de classes, casos de uso, visão geral de interação, atividades e sequência. Elaboração de novos diagramas referentes ao Pygame. Produção de máquina de estado e modelagem de algoritmo.
0.4	Arthur de Sousa Costa, Arthur Medeiros Machado de Souza e Felipe Valentin Nascimento	14/03/2022	Refinamento e conclusão da modelagem UML. Implementação final do código e lançamento de Release Notes.

### Conteúdo:

1. Introdução
2. Visão Geral
3. Requisitos da Aplicação

# 1. Introdução

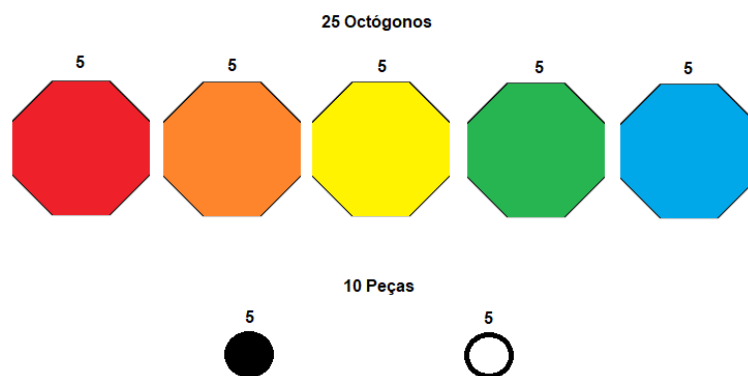
## 1.1. Objetivo

Desenvolvimento de um programa de entretenimento, o qual possui todas as ferramentas necessárias para que dois usuários sejam capazes de jogar PEG-4.

## 1.2. Definições e Regras

### Componentes

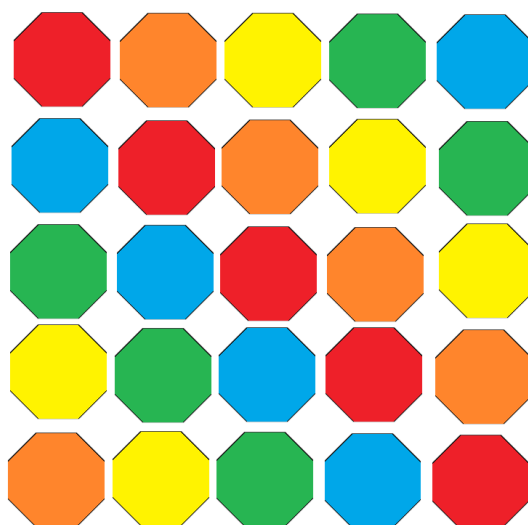
- 25 octógonos coloridos, também chamados de posições. Há cinco cores diferentes distribuídas igualmente, logo, têm-se 5 octógonos para cada cor.
- 10 peças, onde 5 são de cor branca e 5 de cor preta.



Componentes

### Tabuleiro

Para montar o tabuleiro, precisamos organizar os 25 octógonos em 5 linhas e 5 colunas (5x5). Dois octógonos de mesma cor não podem se tocar, ou seja, não podem ser adjacentes (vertical ou horizontalmente).



Exemplo de Tabuleiro

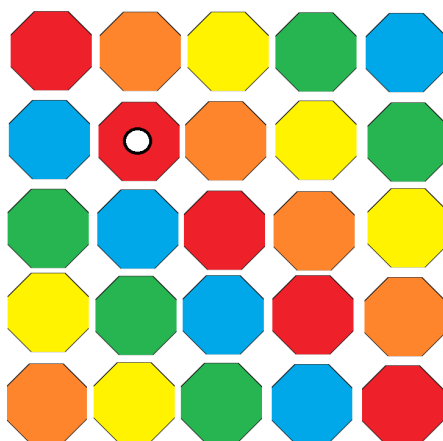
## Objetivo

O objetivo do jogo é pegar 4 octógonos da mesma cor. O jogador que assim o fizer, é considerado vencedor.

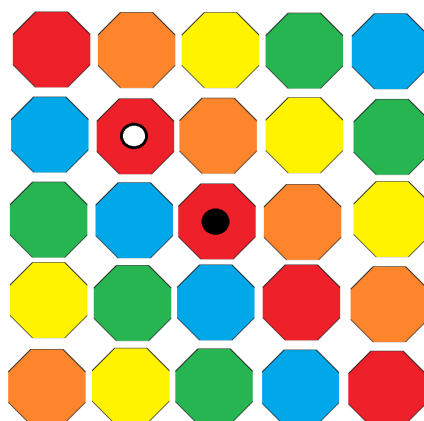
## Como Jogar

PEG-4 é separado em duas fases:

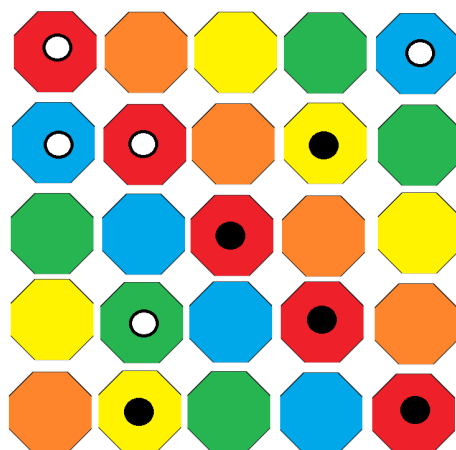
- Fase 1: Colocação das Peças - O jogador que possui as peças brancas começa. Ele escolhe uma de suas peças que não está no tabuleiro e a coloca em cima de qualquer octógono não ocupado. Em seguida, o jogador que possui as peças pretas faz o mesmo. Ambos colocam suas peças no tabuleiro de forma alternada, até que todas as dez peças estejam em cima de octógonos.



Jogador das peças brancas coloca

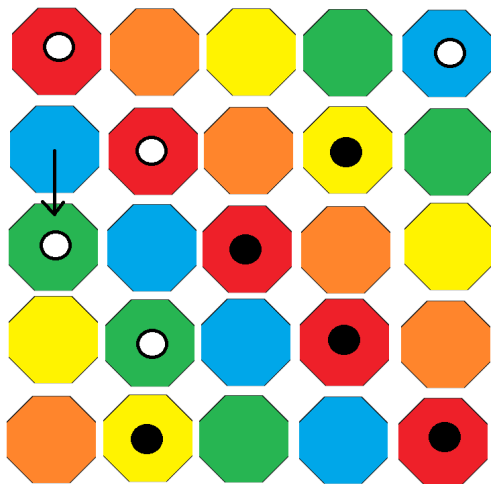


Veza do jogador das peças pretas

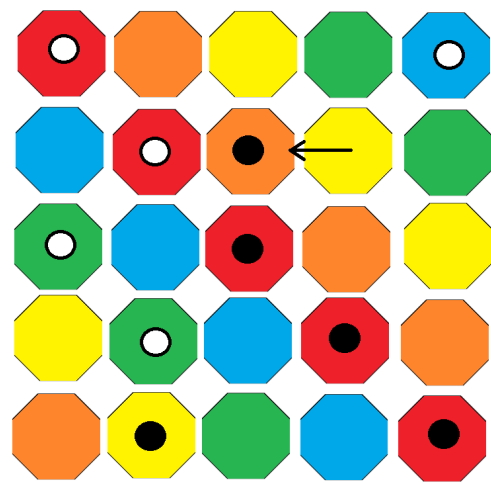


Término da Fase de Colocação

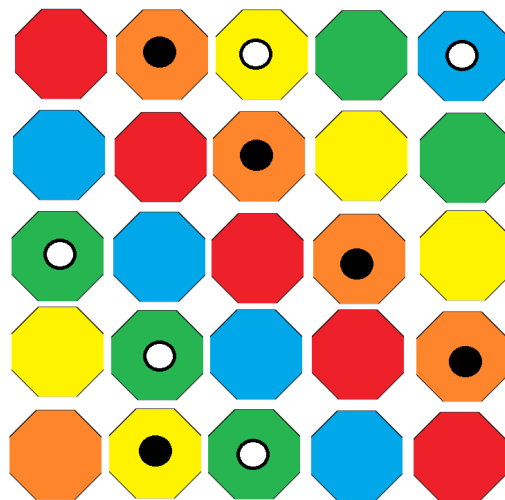
- Fase 2: Movimentação das Peças - Assim que o jogador das peças pretas colocou sua última peça no tabuleiro, começa a fase de movimentação. Agora, na vez do jogador das peças brancas, ele pode escolher qualquer uma de suas peças no tabuleiro e movimentá-la na horizontal ou vertical, desde que o octógono destino não esteja ocupado. Em seguida, o jogador das peças pretas faz o mesmo. Ambos movimentam suas peças no tabuleiro de forma alternada, até que um dos jogadores pegue 4 octógonos da mesma cor e torna-se o vencedor.



Jogador das peças brancas movimenta



Vez do jogador das peças pretas



Vitória do jogador das peças pretas

### 1.3. Referências

<https://www.youtube.com/watch?v=ihmhF9lfZxg>

## 2. Visão Geral

### 2.1. Arquitetura do Software

Programa autossuficiente (*stand-alone*), com interface gráfica do usuário e que utiliza o paradigma de programação orientada a objetos (POO).

### 2.2. Premissas de Desenvolvimento

- O programa deve ser implementado em Python com a biblioteca Pygame
- O programa deve executar em qualquer plataforma capaz de interpretar Python e que possua a biblioteca Pygame
- Ambos os jogadores devem ser capazes de interagir na mesma interface gráfica
- O projeto deve ser produzido com Visual Paradigm

### **3. Requisitos da Aplicação**

#### **3.1. Requisitos Funcionais**

##### **[RF01] Iniciar Jogo**

O programa deve ser capaz de iniciar o jogo com um comando do usuário. Deste modo, o jogo pode ser conduzido de acordo com a vontade dos jogadores, que podem dar início a qualquer momento. O início do jogo deve ser acionado por meio de um clique ou botão, ou outro mecanismo gráfico, presente na janela inicial do programa. Assim que o jogo for iniciado, o tabuleiro é montado de forma aleatória e automática. Os jogadores devem decidir entre si quem tomará posse das peças brancas e pretas, sendo que as brancas sempre começam. Preferencialmente, deve haver uma sinalização das peças ou do nome do jogador que deve jogar em cada turno.

##### **[RF02] Colocar Peça**

Após o início da partida, ambos os jogadores devem colocar suas cinco peças no tabuleiro da interface, de forma alternada. Deste modo, o usuário pode montar sua estratégia e fazer sua jogada, desde que válida. Durante seu turno, o jogador pode colocar uma de suas peças em qualquer posição do tabuleiro que não esteja ocupada, clicando sobre o octógono desejado. Caso sua colocação seja um movimento inválido, é necessário repetir a jogada. Caso contrário, será a vez do próximo jogador. É possível ganhar o jogo na fase de colocação de peças, mesmo que muito improvável. Quando um jogador ganha a partida, uma mensagem é disposta na tela informando a conclusão do jogo, e os jogadores podem iniciar uma nova partida.

##### **[RF03] Mover Peça**

Assim que ambos os jogadores estiverem com suas cinco peças colocadas no tabuleiro, eles podem começar a movê-las. É desta maneira que o jogo é disputado a fim de determinar o vencedor. O jogador, durante seu turno, pode clicar sobre uma de suas peças no tabuleiro, indicando qual delas deseja mover. Em seguida, clica sobre uma posição válida (uma casa desocupada na horizontal ou vertical), revelando o movimento de sua peça. O programa deve ser capaz de detectar movimentos inválidos e alertar o jogador da vez para realizar outra jogada. Assim que movimentar uma de suas peças com sucesso, termina o turno daquele jogador e será a vez do próximo. Nesta etapa, o jogo pode ser finalizado a qualquer momento, assim que um dos jogadores tiver 4 peças em octógonos de mesma cor. Quando um jogador ganha a partida, uma mensagem é disposta na tela informando a conclusão do jogo, e os jogadores podem iniciar uma nova partida.

## **3.2. Requisitos Não Funcionais**

### **[RNF01] Usabilidade**

O programa deve ser de fácil operação, de modo que o usuário seja capaz de manusear o sistema de maneira rápida e eficiente.

### **[RNF02] Portabilidade**

O programa deve ser executado em qualquer plataforma que possua um interpretador Python da versão atual, com acesso a biblioteca de jogos Pygame.

### **[RNF03] Implementação**

O programa deve ser implementado na linguagem de programação Python, na versão atual, em conjunto com a biblioteca de jogos Pygame. O mesmo também deve utilizar o paradigma de programação orientada a objetos.

### **[RNF04] Interface**

O programa deve dispor de uma interface gráfica do usuário, contendo uma tela de menu principal e outra tela para a realização da partida, a qual deve conter todos os elementos necessários do jogo PEG-4, tais como tabuleiro de octógonos, peças de diferentes cores e botões, de modo que o jogo se torne atrativo e proveitoso. Os dois jogadores devem interagir nesta mesma interface.