



Laboratório de Linguagens de Programação
Prof. Andrei Rimsa Álvares

Trabalho Prático II

1. Objetivo

O objetivo desse trabalho prático é permitir que os alunos pratiquem os conceitos de programação orientada a eventos em um sistema computacional. Para isso, será utilizado o framework Qt (<https://qt.io>) sobre C++ para criação de interfaces gráficas no desenvolvimento de uma aplicação.

2. Descrição

Desenvolver uma aplicação gráfica que simule o jogo *Pursuit* (<https://youtu.be/MScOazcEfH0>), onde dois jogadores disputam quem será o vencedor. O jogo possui um tabuleiro 7x7, ou seja, 49 células distribuídas em 7 linhas e 7 colunas. O tabuleiro é preenchido com 27 peças, sendo duas de cores diferentes para cada jogador e outras 25 peças imóveis que servem como bloqueios para as movimentações. Os jogadores alternam jogadas, onde cada uma consiste em duas etapas: uma de mover e outra de retirar uma peça de bloqueio. Na etapa de mover, um jogador deve conseguir movimentar sua peça em qualquer uma das 8 direções da rosa dos ventos (Norte, Nordeste, Leste, Sudeste, Sul, Sudoeste, Oeste, Noroeste), como se fosse uma rainha no jogo de xadrez, de forma a percorrer todo caminho em uma dessas direções até que não seja mais possível – ou seja, até antes de encontrar uma peça de bloqueio ou uma das bordas do tabuleiro. Após o movimento, ele deve fazer a etapa de retirar uma peça de bloqueio do tabuleiro que esteja imediatamente adjacente à sua peça. O jogo termina em duas situações: quando a peça do jogador consegue movimentar até a peça adversária sem encontrar nenhum bloqueio no caminho ou quando um jogador não consegue fazer uma jogada completa (mover e retirar).

3. Instruções

O trabalho deve ser feito obrigatoriamente em C++ utilizando o framework Qt. O download da API e da IDE podem ser encontradas no link: <https://www.qt.io/download-open-source>.

3.1. Configuração

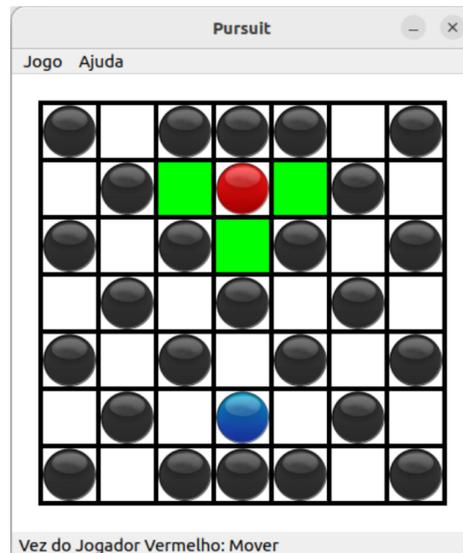
O programa deve conter uma única janela com três componentes:

- 1) Um menu com dois itens:
 - a. **Jogo**: selecionar ações relacionadas ao jogo.
 - i. **Novo**: reiniciar o tabuleiro para a posição inicial.
 - ii. **Sair**: fechar o programa.
 - b. **Ajuda**: obter informações sobre o jogo.
 - i. **Sobre**: abrir um pop-up (*dialog*) com informações sobre o(s) desenvolvedor(es) do trabalho com nome e e-mail.

Laboratório de Linguagens de Programação
 Prof. Andrei Rimsa Álvares

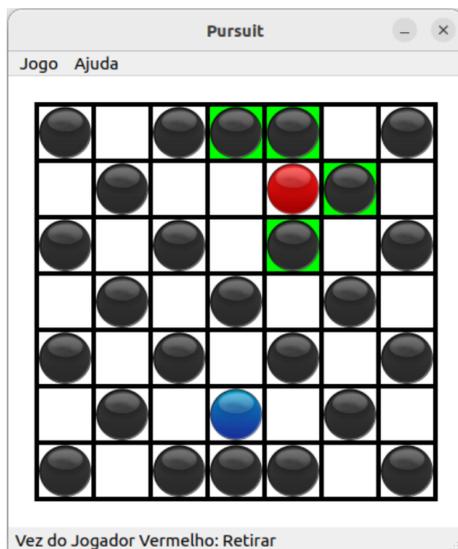
- 2) Um tabuleiro com 49 células distribuídas em 7 linhas e 7 colunas. A distribuição das peças pode ser observada na imagem do exemplo de configuração inicial. As células desenhadas de verde indicam as possíveis jogadas naquele momento.
- 3) Uma barra de status com indicação de qual é a vez do jogador e em que etapa ele está na jogada.

A seguir é dado um exemplo de configuração inicial.

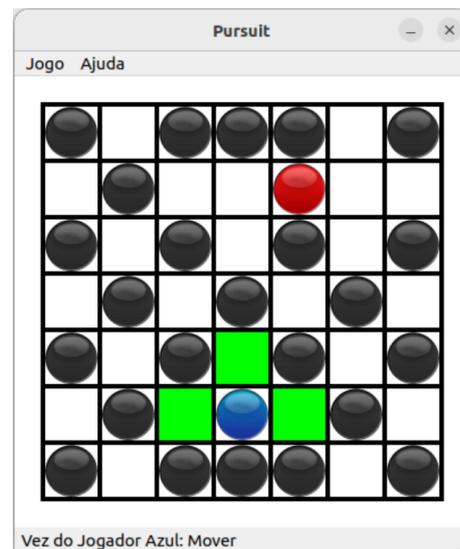


3.2. Jogabilidade

O jogo sempre começa com o Jogador Vermelho que deve primeiro movimentar sua peça e depois retirar uma peça de bloqueio adjacente. Essa mecânica é ilustrada a seguir. Depois, passa-se a vez para o Jogador Azul.



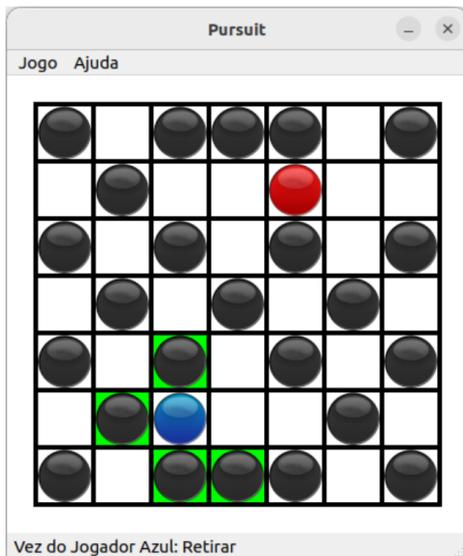
(a) Após mover para a direita.



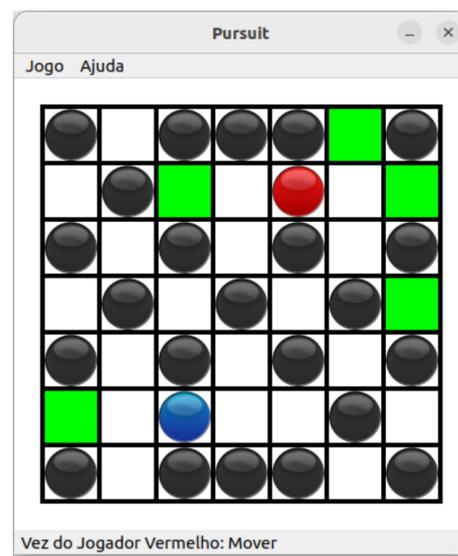
(b) Após retirar a peça à direita.

Laboratório de Linguagens de Programação
 Prof. Andrei Rimsa Álvares

Agora é a vez do Jogador Azul que deve seguir as mesmas etapas, conforme exemplificado a seguir. Note que embora ele tenha movido para a esquerda, ele pode retirar qualquer peça marcada em verde (a). Coincidentemente, ele retirou a peça mais à esquerda (b). Depois de passar a vez para o Jogador Vermelho, observe todas as células em verde para qual ele pode movimentar sua peça.

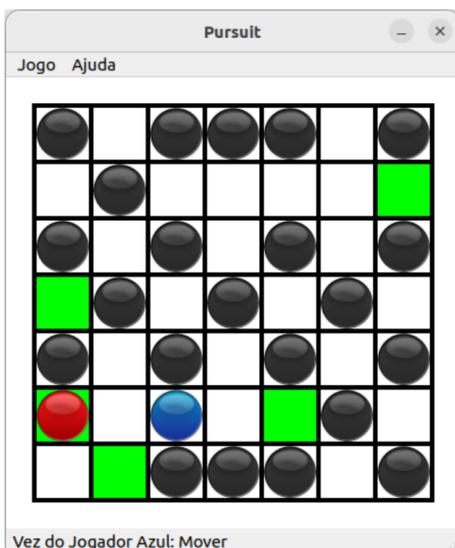


(a) Após mover para a esquerda.



(b) Após retirar a peça à esquerda.

O jogo termina em duas situações. Na primeira situação, um jogador consegue alcançar a peça adversária sem encontrar nenhum bloqueio, conforme pode ser visto a seguir (a) se o Jogador Azul movimentar para a esquerda. Caso ele movimente no sentido vencedor, deve-se mostrar uma janela pop-up indicando que ele ganhou (b).



(a) Antes do Jogador Azul mover.

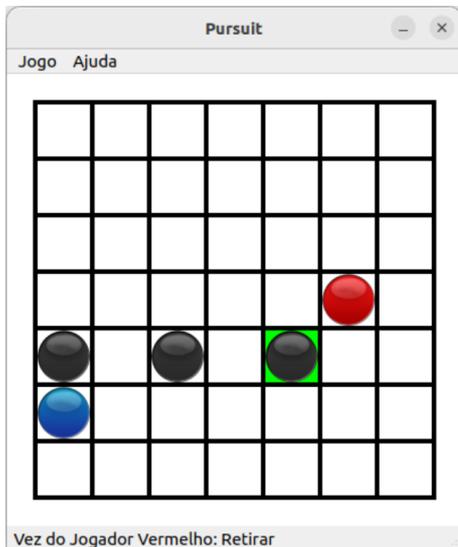


(b) Após mover e vencer o Jogo.



Laboratório de Linguagens de Programação
Prof. Andrei Rimsa Álvares

A outra situação é quando um jogador não pode fazer mais uma jogada válida. Nesse caso, a vitória é dada ao jogador adversário, conforme pode ser observado a seguir.



(a) Antes do Jogador Vermelho retirar. (b) Jogador Azul não possui jogada válida.

4. Avaliação

O trabalho deve ser feito em grupo de até dois alunos, sendo esse limite superior estrito. O trabalho será avaliado em 15 pontos, onde essa nota será multiplicada por um fator entre 0.0 e 1.0 para compor a nota de cada aluno individualmente. Esse fator poderá estar condicionado a apresentações presenciais a critério do professor.

Trabalhos copiados, parcialmente ou integralmente, serão avaliados com nota **ZERO**, sem direito a contestação. Você é responsável pela segurança de seu código, não podendo alegar que outro grupo o utilizou sem o seu consentimento.

5. Submissão

O trabalho deverá ser submetido até as 23:59 do dia 17/06/2024 (segunda-feira) exclusivamente via sistema acadêmico em pasta específica. **Não serão aceitos, em hipótese alguma, trabalhos enviados por e-mail ou por quaisquer outras fontes.** A submissão deverá incluir todo o código-fonte do programa desenvolvido e arquivos de apoio necessários em um arquivo compactado (zip ou rar). Nesse caso, deve ser enviado todo o diretório de trabalho que inclui o arquivo de projeto (*.pro), códigos-fonte (*.h e *.cpp), arquivos de design (.ui), arquivos de recursos (*.qrc) e outros arquivos pertinentes. **Não serão consideradas submissões com links para hospedagens externas.** Para trabalhos feitos em dupla, deve-se incluir um arquivo README na raiz do projeto com os nomes dos integrantes da dupla. Nesse caso, a submissão deverá ser feita por apenas um deles.