

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 *Catch* (<https://boardgamegeek.com/boardgame/15929/catch>), onde dois jogadores disputam quem será o vencedor. O jogo possui um tabuleiro 8x8 com 64 células, onde os jogadores alternam jogadas bloqueando duas células adjacentes. Um jogador bloqueia duas células sempre na horizontal, enquanto o seu oponente sempre na vertical. Se um jogador cercar uma área contendo de 1 a 3 células, seja por bloqueios ou pelas bordas do tabuleiro, então ele captura essa área para ele. O jogo termina se não houver jogada possível para o jogador atual. Vence o jogador que tiver mais células capturadas, ou empata se a quantidade for a mesma de seu oponente.

### 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-qt-installer>.

#### 3.1. Configuração

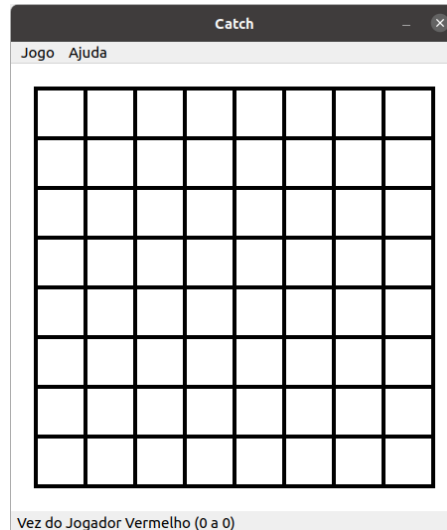
O programa deve conter uma única janela com três componentes:

- 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.
- 2) Um tabuleiro 8x8 com 64 células. Ao mover o mouse por uma célula, o tabuleiro deve indicar, com uma cor verde, se essa célula e sua célula adjacente (na horizontal ou na vertical) podem ser selecionadas por esse jogador. Caso sejam selecionadas (ao clicar na célula), essas devem ser marcadas, com uma cor cinza. Caso células sejam capturadas, deve-se marcar quem as capturou, através de peças da cor de seu jogador.
- 3) Uma barra de status com indicação de qual jogador é a vez e células foram capturadas até o momento pelo jogador atual e seu oponente.

## Laboratório de Linguagens de Programação

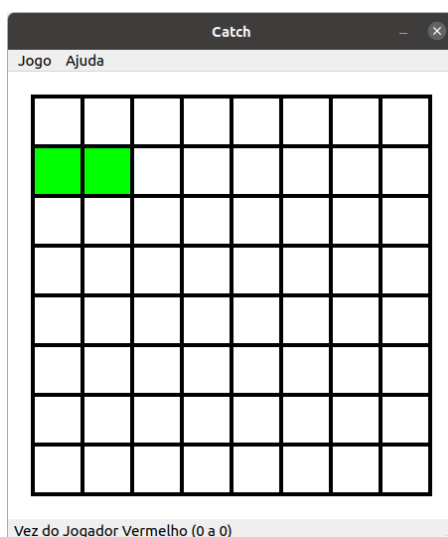
### Prof. Andrei Rimsa Álvares

A seguir é dado um exemplo de configuração inicial.

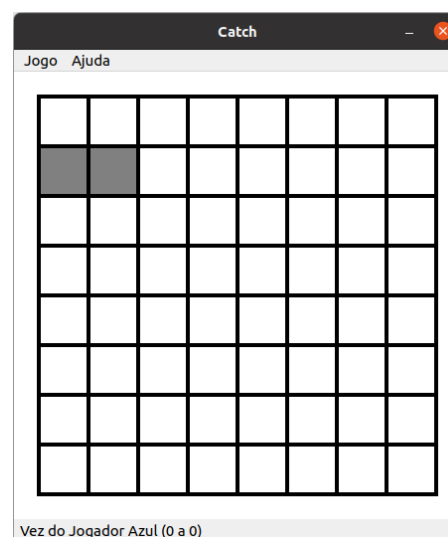


### 3.2. Jogabilidade

Após estabelecida a configuração inicial do tabuleiro, começa-se pelo Jogador Vermelho que deve selecionar duas células adjacentes sempre na horizontal. Por exemplo, ao passar o mouse pela célula na segunda linha e primeira coluna ( $C_{2 \times 1}$ ), essa célula e sua célula adjacente na horizontal ( $C_{2 \times 2}$ ) são marcadas indicando, com uma cor verde, que essa jogada pode ser feita (a). Note que nenhuma das células da última coluna são elegíveis para selecionar, já que não existem células adjacentes a elas na horizontal. Após clicar nessa célula, ela e sua célula adjacente são marcadas permanentemente, com uma cor cinza, indicando que não podem mais ser selecionadas (b). Nesse caso, nenhuma célula foi capturada. Passa-se a vez para o Jogador Azul.



(a) Antes da jogada.

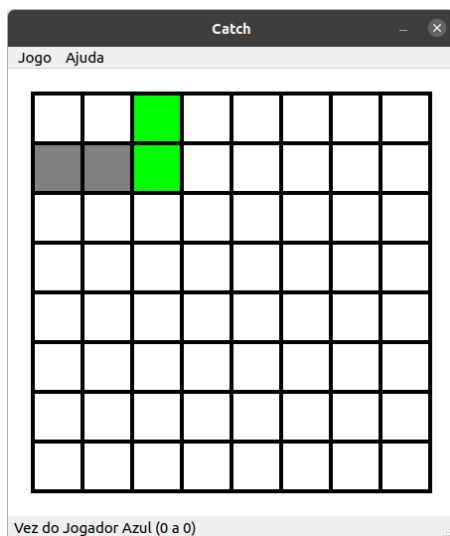


(b) Após a jogada.

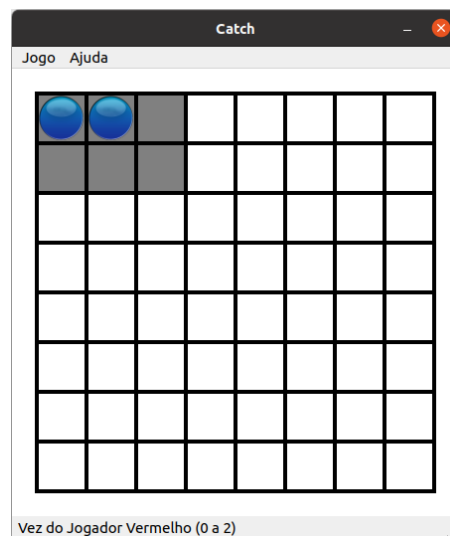
O Jogador Azul deve selecionar duas células adjacentes, sempre na vertical agora. Por exemplo, ele pode passar o mouse sobre a célula  $C_{1 \times 3}$  que

Laboratório de Linguagens de Programação  
Prof. Andrei Rimsa Álvares

deverá indicar que ela e a célula  $C_{2 \times 3}$  (célula adjacente abaixo) podem ser escolhidas (c). Repare que as células  $C_{1 \times 1}$  e  $C_{2 \times 2}$  não são elegíveis nessa jogada, já que não é possível selecionar duas células na vertical por existirem células marcadas na vizinhança imediata. Também não é possível escolher nenhuma das células da última linha, já que estão na borda do tabuleiro. Se for feita a jogada, então as células  $C_{1 \times 3}$  e  $C_{2 \times 3}$  são marcadas. Como duas células foram cercadas ( $C_{1 \times 1}$  e  $C_{2 \times 2}$ ), considerando as bordas e as células marcadas, então o Jogador Azul pode capturá-las, marcando com uma peça Azul de sua cor (d). Passa-se a vez para o Jogador Vermelho fazer sua próxima jogada. Repare que a barra de status indica a vez do Jogador Vermelho e que está perdendo de 0 a 2.



(c) Antes da jogada.



(d) Após a jogada.

Os jogadores vão alternando as jogadas até que não hajam mais jogadas possíveis, ou seja, não existem duas células que possam ser selecionadas. Nesse caso, verifica-se qual jogador possui mais células capturadas e o declara como vencedor (e) ou declara empate se for o caso (f).



(e) Jogador Vermelho venceu.

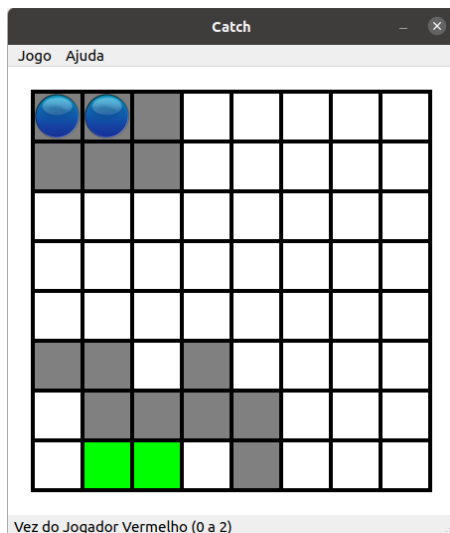


(f) Empate.

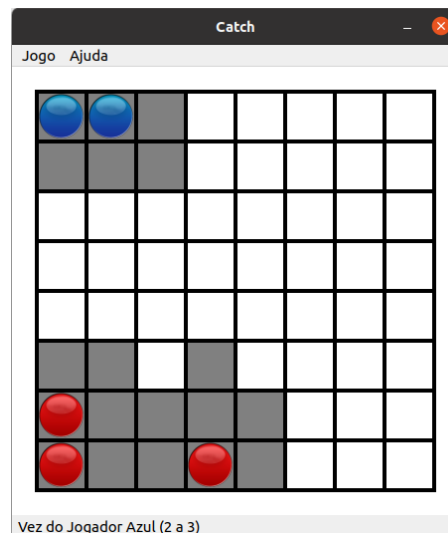
## Laboratório de Linguagens de Programação

### Prof. Andrei Rimsa Álvares

Note que durante o jogo, múltiplos grupos de 1 a 3 células podem ser capturadas simultaneamente. Considere a situação a seguir. Caso o jogador Vermelho escolha a célula  $C_{8 \times 2}$  (g), então dois grupos de 1 a 3 peças são formados. O grupo contendo as células  $C_{7 \times 1}$  e  $C_{8 \times 1}$  e o grupo contendo somente a célula  $C_{8 \times 4}$ . Essas células desses grupos são então capturadas (h).



(g) Antes da jogada.



(h) Após a jogada.

## 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 19/06/2023 (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.