

# PSI2651 - EP1

Autor: Felipe Anez de Toledo Blassioli NUSP: 7991058

## Executável

Foi utilizado OpenCV 2.4.10. Para gerar o executável:

```
$ make ; ./ep1
```

**OBS:** Garantir que existe a imagem do template *default\_template.png*

## Estrutura

- **main.cpp:** Deveria parsear parametros. Possui o *int main()*.
- **app.cpp:** Loop principal da aplicação. Lê o arquivo de entrada, acessa a câmera mandando os frames capturados para processamento pelo *target\_detector*.
- **ui.cpp:** Painel de Control e botões e intersecção desses objetos com os targets.
- **helpers.cpp::** *somatorioUm* e *dcReject*.
- **detector/**
  - **detector.cpp:** Classe dos detectores de target. Possui a implementação básico de *process\_frame*. E faz a escala do template original.
  - **grayscale.cpp:** Herda de *TargetDetector*. Utiliza do *opencv.matchTemplate* sobre frames e templates em escalas de cinza.
  - **candidate.cpp:** Descreve um candidato a target. Possui o template correspondente e dados geometricos.

## Funcionamento

O processamento de frames básico pode ser visto em *src/detector/detector.cpp*. Em *TargetDetector::process\_frame* o frame é pré-processado e sobre o resultado aplica-se *templateMatching* sobre a regioao do painel (e nao do frame inteiro) gerando uma lista de candidatos, depois é escolhido o melhor candidato.

O arquivo principal é *src/detector/grayscale.cpp*. Esse detector de targets pré-processa os templates (versões escaladas de *default\_template.png*) que recebe fazendo *somatorioUm(dcReject(frame))* e depois transforma para escala de cinza.

O template-matching é feito em *GrayscaleTargetDetector::match* e utiliza da função do *opencv matchTemplate*. A metrica utilizada foi *CV\_TM\_CCOEFF\_NORMED* e encontrei um parâmetro aceitável de valor máximo maior que **0.58** para candidatos a targets.

Em *GrayscaleTargetDetector::find\_best\_candidate* seleciono o candidato de score de valor máximo. O score é o valor encontrado por *minMaxLoc* do *opencv* sobre o resultado de *matchTemplate*.

A aplicacao (*src/app.cpp*) recebe o melhor candidato do detector de targets e faz uma intersecção com o painel de controle (*src/ui.cpp*), que por sua vez faz a intersecção com seus botões.