Relatório do trabalho da disciplina de Introdução à Visão por Computador

Trabalho Prático

Gonçalo Oliveira Araújo - 27928

João Manuel Freitas Ribeiro – 27926

Tiago Miguel Dias Miranda - 27937

Engenharia em Desenvolvimento de Jogos Digitais

Outubro de 2024

|  |
| --- |
| Afirmo por minha honra que não recebi qualquer apoio não autorizado na realização deste trabalho prático. Afirmo igualmente que não copiei qualquer material de livro, artigo, documento web ou de qualquer outra fonte exceto onde a origem estiver expressamente citada. |

Gonçalo Oliveira Araújo - 27928

João Manuel Freitas Ribeiro – 27926

Tiago Miguel Dias Miranda - 27937

**Índice**

[Objetivo do Trabalho 1](#_Toc181109610)

[Jogo Base 1](#_Toc181109611)

[Algoritmo de Segmentação 1](#_Toc181109612)

[Obstáculo Iniciais 2](#_Toc181109613)

[Desenvolvimento do Projeto 2](#_Toc181109614)

[Proximo psssssa 2](#_Toc181109615)

[<Nome do Capítulo II> 3](#_Toc181109616)

# Objetivo do Trabalho

Este trabalho tem como objetivo colocar em prática as técnicas e os conhecimentos sobre Visão por Computador, adquiridos durante as aulas da disciplina, transformando um jogo simples feito em Python num jogo que pode ser controlado através da imagem captada pela câmara baseado em algoritmos de segmentação.

## Jogo Base

O jogo que decidimos usamos como base foi o “Pong”, para ser mais específico foi usada uma cópia do jogo feita em Python usando a biblioteca “PyGame”, por um usuário chamado de “Baris Bayrak” e publicado no site oficial do PyGame. ([link](https://www.pygame.org/project-Very+simple+Pong+game-816-.html))

Uma imagem com captura de ecrã, Objeto astronómico, natureza, astronomia

Descrição gerada automaticamente

## Algoritmo de Segmentação

## Desenvolvimento do Projeto

**1:** Capturamos a imagem da câmara e convertemos de BGR para HSV (usando a função *cv2.COLOR\_BGR2HSV* da biblioteca *OpenCV*) para posteriormente facilitar a segmentação.

**2:** Através das *TrackBars* implementadas o código define as máscaras usando os valores máximos e mínimos dos parâmetros relativos às cores, no caso, como usamos HSV temos de definir para a Gama (*Hue*), para a Saturação (*Saturation*) e para a Luminosidade (*Value*). A implementação das *TrackBars* traz muitas vantagens na captação das cores, nos auxiliando a especificar precisamente os parâmetros da cor que desejamos captar, pois com a mudança de ambiente alguns aspetos das cores podem ser visivelmente diferentes.

**3:** Com a imagem captada, usamos as máscaras para filtras apenas as cores que desejamos, neste projeto usamos as cores Verde e Azul pois são cores mais incomuns, dificultando assim que elementos do cenário interfiram.

4: (O código dá display ao resultado das filtragens com as máscaras)

**5:** Usando as cores já filtradas determinamos (como o código faz o retângulo na cor)

6: (elimina os ruídos)

7: (determina o ponto no meio do retângulo)

8: (faz uma escala com o height da tela de jogo)

9: (move os paddles)

## Problemas Encontrados

(aquilo que o prof explicou na aula sobre os limites da height)

(endender o código do jogo e implementar o movimento dos paddles, pelo facto de python ser uma linguagem nova e blablablaba)

(seila …)

# <Nome do Capítulo II>

A formatação deve ser consistente ao longo de todo o documento. Uma das formas de manter a formatação consistente é recorrer a estilos para formatar os principais formatos utilizados, como se apresenta de seguida.

Os itens podem ser numerados de uma forma (estilo itens01):

1. Primeiro ponto
2. Segundo ponto
3. …

Ou de outra (estilo itens02):

Primeiro ponto

Segundo ponto

…

Podem também não ter qualquer numeração (estilo itens03), mas uma marca:

* Primeiro ponto
* Segundo ponto
* …

Ou outra (estilo04)

* Primeiro ponto
* Segundo ponto
* …

Não esqueça de incluir um capítulo final para a Bibliografia que deverá seguir o formato APA.