

## TP5

# Um jogo simples de basquetebol, usando computação visual

Nuno Carmo Vivas – 74799 – a74799@ualg.pt

Relatório Técnico – LESTI – Computação Visual

### Resumo

O objetivo deste trabalho é apresentar, em forma de jogo, as várias tecnologias aprendidas no decorrer da cadeira de Computação Visual. Estabeleceram-se metas iniciais, em forma de documento-proposta, as quais foram enviadas ao Professor e posteriormente, aprovadas. De forma incremental, foram-se completando os desafios no código e o resultado final é um jogo funcional. No final é perceptível que o modelo de computação visual *MediaPipe* está bastante avançado e consegue, de forma mais ou menos estável, apresentar uma *precision* bastante alta no que se refere a identificação de *features* do corpo humano. Usaram-se dois processamentos diferentes, um para processar apenas mãos e outro para processar cara, pois foi estabelecido como objetivo pelo Professor fazer identificação da cara também. É a melhor solução para este tipo de trabalho, não obstante *Python* não suportar nativamente “*event driven development*”

### 1. Introdução

O problema apresentado resume-se em conseguir demonstrar, de forma mais ou menos simples, o uso de técnicas aprendidas no decorrer da disciplina de Computação Visual. Estas técnicas focam-se, como o próprio nome da disciplina indica, no processamento de imagens usando, sobretudo, técnicas de Inteligência Artificial, mas também outras *libraries*, como o *openCV*, de modo a fazer um número grande de operações e transformações sobre as imagens para conseguir determinado objectivo. A introdução fornece o contexto para o estudo. Define a questão de investigação, a sua relevância e como o estudo pretende abordá-la. Deve atrair o

interesse do leitor, estabelecendo o cenário para o que virá a seguir.

### 2. Revisão da Literatura

Durante o semestre, procurou-se percorrer um pouco a história da computação visual, começando-se por usar modelos mais antigos, para terminar em modelos já recentes e com muito maior capacidade. Esta evolução é notória neste trabalho o qual seria impossível de realizar, pelo menos de forma prática, usando modelos com 10 ou mais anos. Em termos práticos, se fosse opção teria optado pelo uso de apenas um modelo (e assim, um único processamento) para realizar este trabalho. Deixando de fora reconhecimentos de faces e outro tipo de processamentos. Permitiu-me perceber estas “*frameworks*” de uma forma, pelo menos, introdutória o que facilitará não só o planeamento como eventual trabalho futuro na área, caso surja.

### 3. Materiais e Métodos

Esta secção descreve os métodos e técnicas utilizados para recolher, analisar e interpretar os dados. Deve ser detalhada o suficiente para que outros possam replicar o estudo. Deve incluir informações sobre o design do estudo, a seleção de participantes, os instrumentos e procedimentos utilizados, os métodos de análise dos dados, etc.

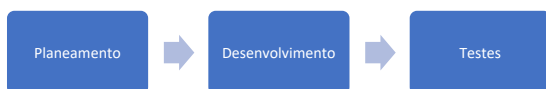
#### 3.1. Metodologia

A metodologia usada em programação foi do tipo incremental, pretendendo dividir os problemas em problemas mais pequenos e resolvendo-os um a um, de forma intercalada. Inicialmente foi definido

um roteiro (na tabela em baixo), discutido com o Professor e este serviu como base para a execução do trabalho.

Contexto	Evento	Resposta	Algoritmo	Prioridade
Reconhecimento de gestos	Pinch em cima da bola	"Apanhar" objeto	Hand Landmarks	Alta
Movimento Corporal	Movimento da mão enquanto pinch existe	"Transportar objeto"	Hand Tracking	Alta
Movimento dos dedos	Release	"Largar" objeto	Hand Landmarks	Alta
Colocação Virtual de objetos/ Detecção de corpos	Spawn do cesto, após x segundos e depois de ponto ou falha	Cria um cesto/arco no ecrã, em posição aleatória, fora dos limites de corpo detetados	Circle function / Body detection	Média
Definição de posição	Score – Sobreposição de elementos gráficos ou texto	Verificar se bola ou objeto está na mesma posição que o cesto	Hand Tracking	Média
Display de texto	Display Score	Mostrar o Score no ecrã	Opencv.putText	Média
Dificuldade	Swipe Left/Right	Diâmetro do cesto	Circle function / frame width	Média
Face	Jogo entra em pausa	Se não houver deteção de face, o jogo pausa	Face detection	baixa
Confettis	Quando encesta	Sempre que é feito um cesto é mostrada uma animação (além de somar o score)	Frame insertion	baixa
Duas mãos levantadas	Pausa	Jogo entra em pausa	Hand Tracking	baixa

Sempre que possível, foram usadas funções para aplicar as mesmas soluções em várias zonas do código. O projeto teve 3 fases, a saber:



### 3.2. Equipamento e ferramentas utilizadas

Este projeto foi inteiramente desenvolvido utilizando um Apple M1 como computador de trabalho. Como ferramentas, foram utilizadas o *Blender* para criação de imagens em três dimensões, *Visual Studio Code* como IDE, *Gimp* para processamento de imagem, *chatGPT* (versão 3.5) para produção de código, Microsoft Word como processador de texto e Github como repositório. Este repositório é público e pode ser consultado por qualquer pessoa.

Como livrarias, usaram-se *opencv*, *os*, *time*, *mediapipe*, *statistics*, *random*, *math* e *numpy*. Nem todas chegaram ao formato final.

## 4. Resultados e discussão

Os objetivos foram conseguidos e o projeto implementa um jogo de "basquetebol virtual" onde o jogador apanha numa bola, colocada ao acaso no ecrã, com qualquer uma das mãos e a transporta para um cesto, este também, colocado ao acaso, no ecrã. De notar que durante o projeto, inicialmente começou-se por usar apenas o modelo das mãos do *mediapipe* e depois foi feito um refactor para que o programa usa-se o modelo holístico.

### 4.1. Resultados experimentais

Por aqui a tabela proposta e depois falar dos obstáculos encontrados e de as escolhas tomadas. Por também os problemas anotados no código.

O primeiro grande obstáculo foi a definição de *pinch*. Tentou-se primeiro fazer de forma ótima, isto é, utilizando apenas o dedo indicador e o dedo gordo, mas provou-se não ser suficiente. Depois de analisado e discutido com o Professor, chegou-se à conclusão que o melhor seria fazer com todos os dedos. Depois deste código implementado, o *pinch* ficou significativamente mais robusto. Depois foi o movimento da bola. É fundamental que o deslocar da bola seja natural para o jogador. Aqui, a opção tomada foi de ter como referência apenas o dedo indicador (pela simples razão de ser aquele que tem a maior probabilidade de ser identificado). Esta opção, mostrou-se bastante eficaz e tornou o jogo bastante fluido. A bola a girar, foi um desafio superado logo na definição. Têm-se uma série de imagens em sequência, pré-carregadas numa lista, as quais são mostradas na imagem consoante o índice de frame atual (baseado na média por segundo). De notar que tanto o movimento como o pinch apenas são permitidos à mão esquerda do jogador. O resultado é um movimento fluido, mesmo em computadores como o Mac M1 que são notoriamente mais lentos (por enquanto) a usar livrarias como o *openCV*. Foi implementada uma função que, de forma aleatória faz *spawn* de uma caixa. Esta caixa pode ser usada

para mostrar tanto a bola de basquetebol, como o cesto. O aparecer da bola em sítios ao acaso, torna o jogo mais interessante. São feitos alguns displays no ecrã. É mostrado o resultado e dificuldade no canto inferior direito. No canto superior direito a aferição dos *pinches* e na mão, os pontos atualmente detetados. Esta última opção foi tomada porque se verificou que havia alturas em que o algoritmo não deteta ponto nenhum e assim proporciona-se ao jogador uma pista visual do resultado dos seus movimentos. Por fim, como medida para evitar medidas inválidas no programa, foi definido que sempre que a bola sai do ecrã, é reiniciada para um ponto ao acaso no ecrã.

Posteriormente, tratou-se a alteração de dificuldade via movimentos como thumbs up e thumbs down que foram aferidos via função comparando a base da mão com a ponta do dedo. Esta acontece apenas de 60 em 60 frames de modo a evitar mudanças sequenciais de dificuldade. Apenas é permitido fazer mudança de dificuldade com a mão direita do jogador.

#### 4.2. Análise dos resultados

A ideia inicial – o requerimento/requirement - de ter *swipes* para alterar a dificuldade provou criar muitas dificuldades na execução já que afeta diretamente o transporte da bola de basquetebol. Depois de várias iterações, concluiu-se que a melhor hipótese seria ter reconhecimento de gestos, ou solução semelhante. Atualmente, o MediaPipe já oferece este tipo de reconhecimento, não utilizando diretamente a livreria, mas através de *tasks* e de modelos pré-treinados próprios. Este tipo de tecnologia fica fora do scope deste projeto, que visa cimentar a utilização básica das livrerias estudadas nas aulas. Se houver continuidade neste projeto esta será certamente uma opção a seguir até porque é computacionalmente mais eficiente.

O programa consegue com sucesso e bastante precisão detetar *pinches* e transportar a bola. Consegue detetar poses específicas, como thumbs up e down, bem como caras. Elemento este que tem que estar sempre presente senão o programa entra em

pausa. Os objetivos foram assim todos alcançados e conseguiu-se, através de um simples jogo, demonstrar uma série de tecnologias e funcionalidades que a computação visual, atualmente, nos disponibiliza.

#### 5. Conclusão

Uma das conclusões a que chego ao fim deste trabalho é que o *mediapipe* é uma excelente livreria para nos iniciarmos na computação visual. Tem vários modelos pré-trabalhados, que permitem reconhecimento facial, mãos e de pose. É bastante versátil e eficiente. Não obstante, quando queremos ter mais controlo sobre o processamento e ao mesmo tempo aumentar precisão, é necessário adotar estratégias que impliquem usar modelos específicos para as tarefas que estamos a tentar executar.

O trabalho pode ser melhorado em muitos aspetos. Talvez o principal, tenha que ser um esforço suplementar para conseguir grupo de trabalho. Este trabalho foi realizado por uma única pessoa. Nestes casos, é frequente perder-se a perspetiva e estou seguro que isso influenciou algumas das decisões que foram tomadas.

Julgo que todos os objetivos propostos foram alcançados. Não obstante ainda haver caminho a percorrer, julgo que evolui bastante na linguagem de programação Python e que os conceitos de computação visual ficaram bastante bem cimentados como resultado do trabalho posto neste projeto.

#### Agradecimentos

Gostaria de agradecer à minha Mulher, o apoio constante durante esta fase, diferente, da minha vida. Ao Professor Jaime, a dedicação e alta disponibilidades mostradas para com os alunos desta disciplina.

#### Referências

Índices MediaPipe para dedos, (Outubro de 2023), Google, [https://developers.google.com/mediapipe/solutions/vision/hand\\_landmarker](https://developers.google.com/mediapipe/solutions/vision/hand_landmarker)

Configurações de modelo de mãos, (Março de 2023), Google, <https://github.com/google/mediapipe/blob/master/docs/solutions/hands.md>

Operações aritméticas em imagens, Equipa OpenCV, [https://docs.opencv.org/3.4/d0/d86/tutorial\\_py\\_image\\_arithmetics.html](https://docs.opencv.org/3.4/d0/d86/tutorial_py_image_arithmetics.html)

Modelo holístico do mediapipe, (Junho de 2023), Google, <https://github.com/google/mediapipe/blob/master/docs/solutions/holistic.md>