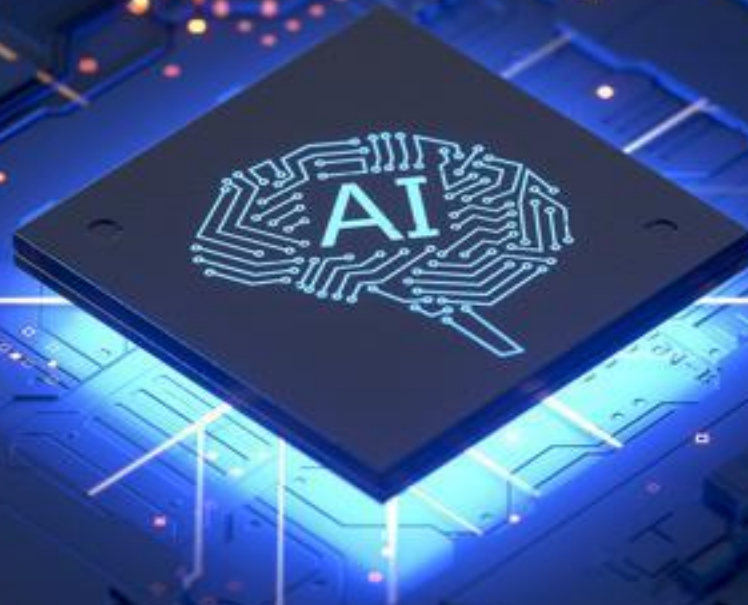


# Real-time Pose Estimation

Landmark detection with python for Unity Engine



# Project description

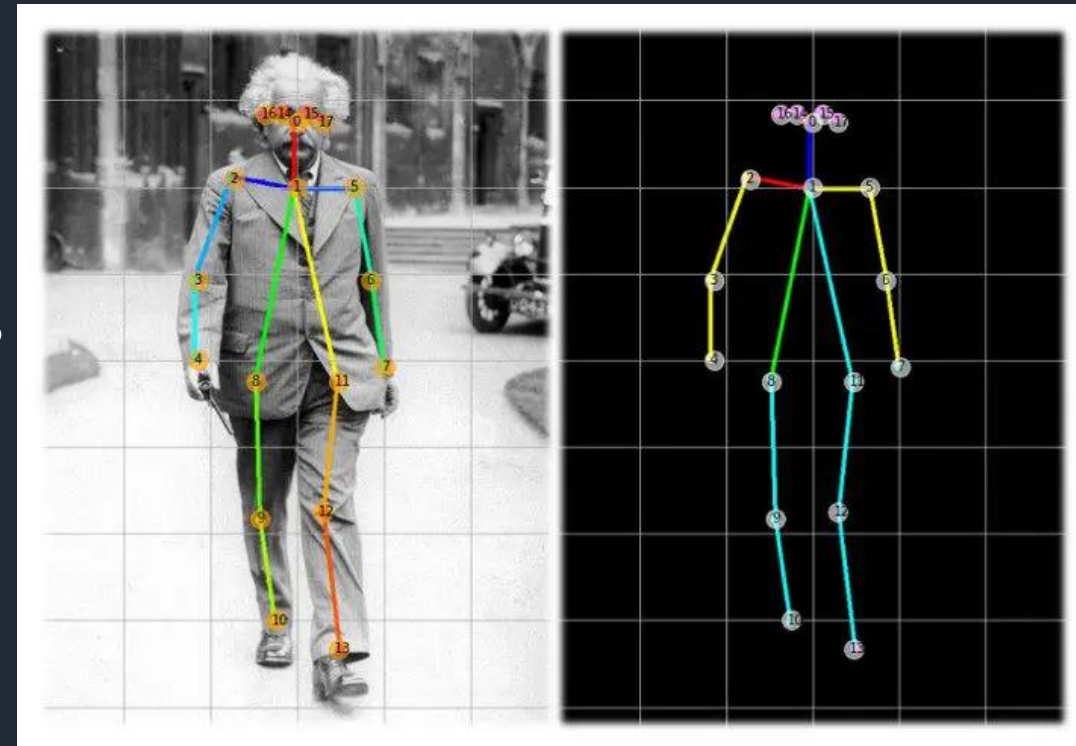
What?

Real time pose estimation for Untiy Engine with AI (ML)

Why?

Record Animations

Control Games with movements



# MediaPipe

- Flexible Framework developed by Google for multimedia processing Pipelines
- Consts of:
  - Graph-based Pipeline -> define Data flow yourself
  - Preprocessing Components -> pre-built components (modular units)
    - Feature-extractors, video decoders, machine learning components
  - Custom Components -> Dev Components yourself
  - Connect different Sources to MediaPipe
  - Synchronization and Threading
  - Optimizing and Hardware Acceleration

# MediaPipe

## YOLOv7 vs MediaPipe Pose Features

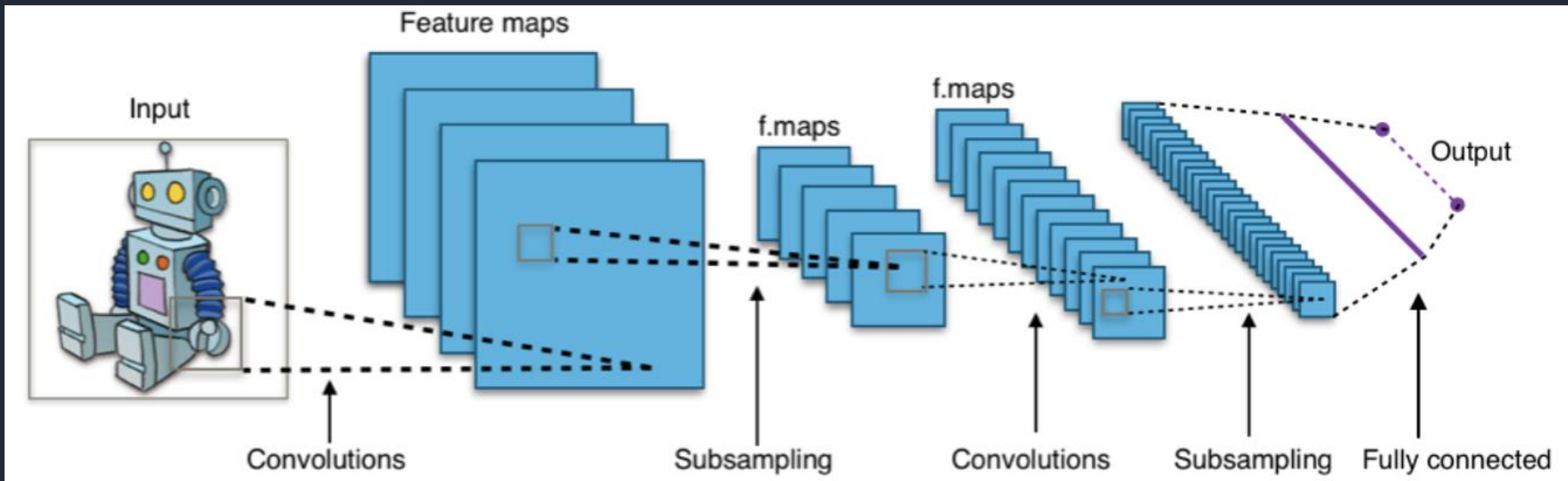
Features	YOLOv7 Pose	MediaPipe Pose
Topology	17 Keypoints <b>COCO</b>	33 Keypoints <b>COCO + Blaze Palm + Blaze Face</b>
Workflow	Detection runs for all frames	Detection runs once followed by tracker until occlusion occurs
GPU support	Support for both CPU and GPU	Only CPU
Segmentation	Segmentation not integrated to pose directly	Segmentation integrated
Number of persons	Multi-person	Single person



# MediaPipe

## BlazePose

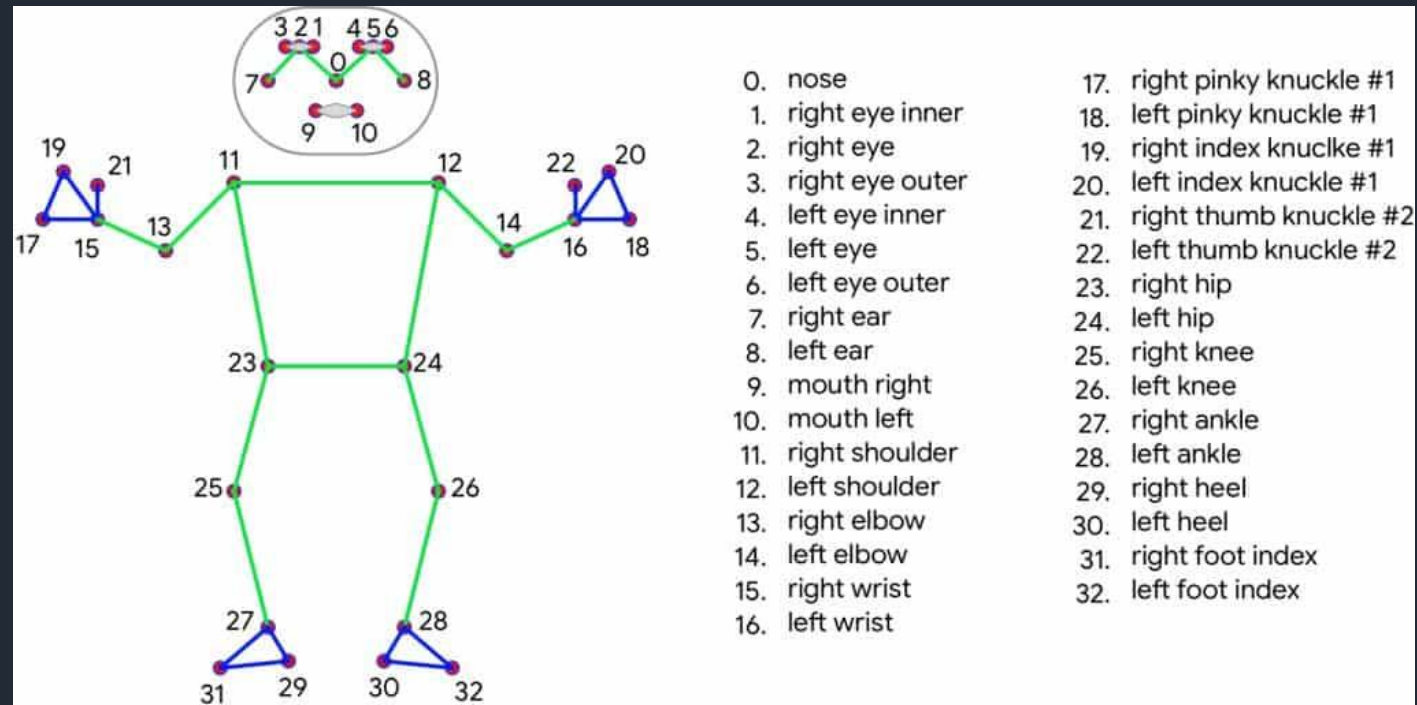
- Convolutional Neural Network (CNN)-based model
  - Convolutional Layer (Matrices with illumination values)
  - Pooling Layer (clean data)
  - Full-connected Layer (classification)



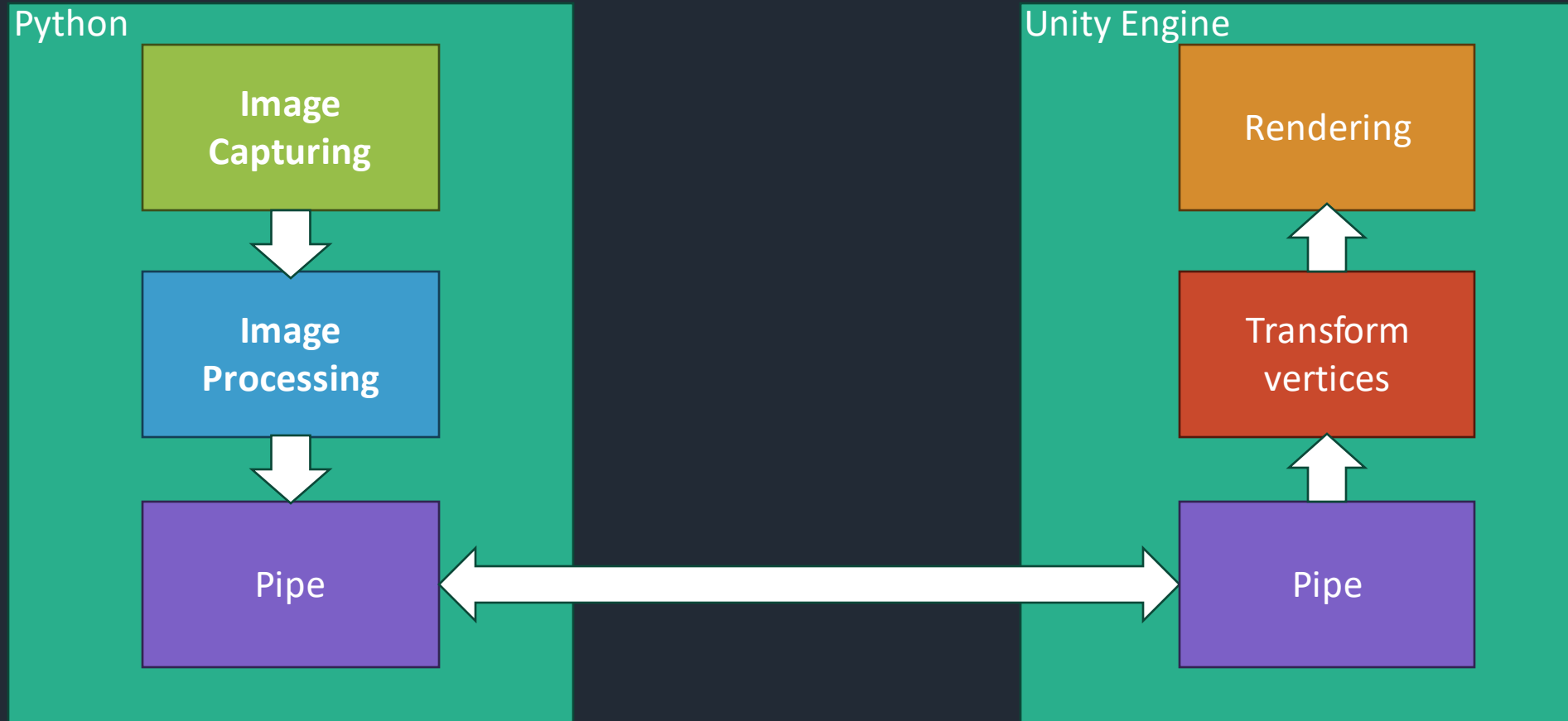
# MediaPipe

## BlazePose

- extracts 33 2D landmarks on the human body
- BlazePose is a lightweight machine learning architecture that achieves real-time performance on mobile phones and PCs with CPU

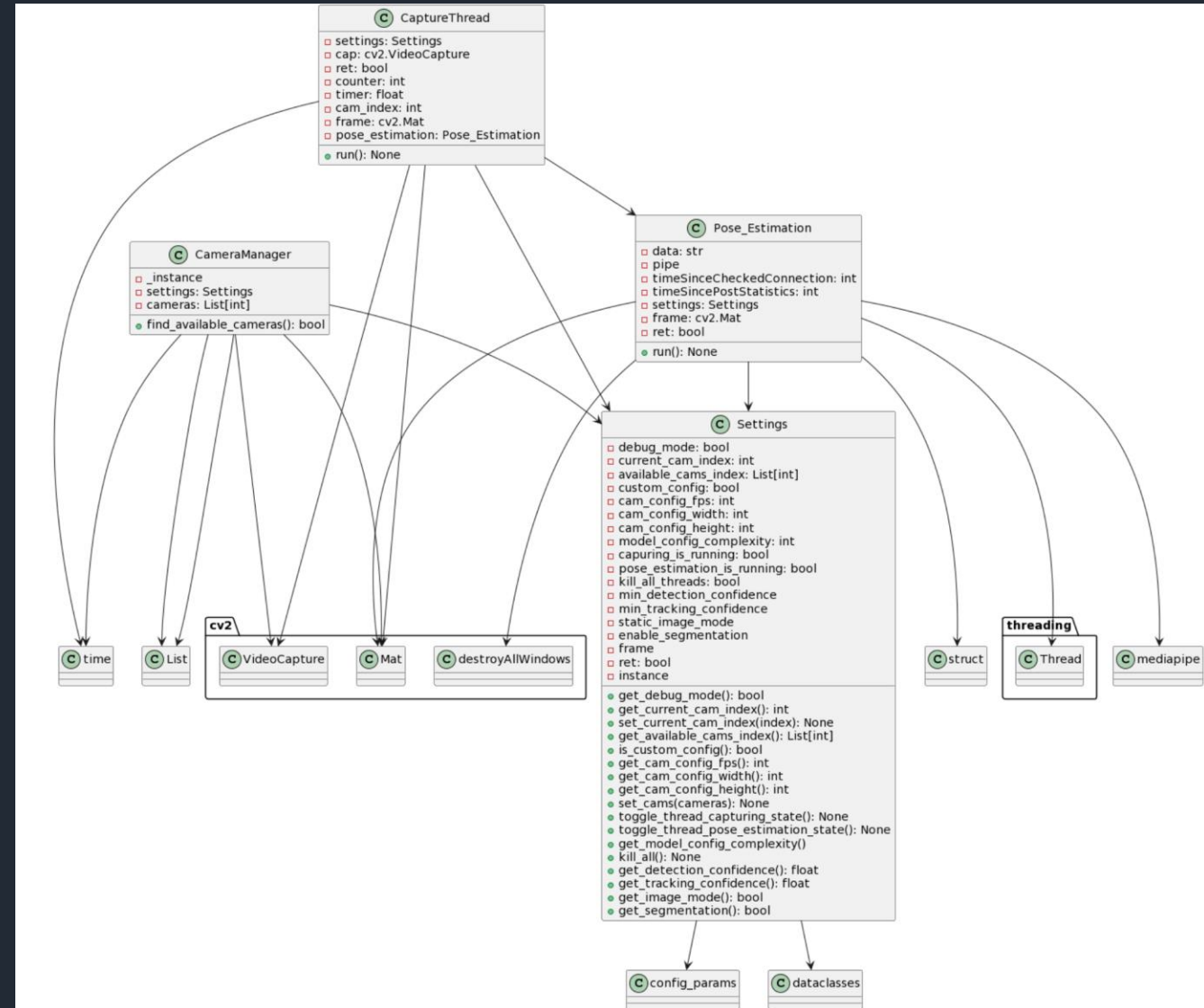


# System Overview

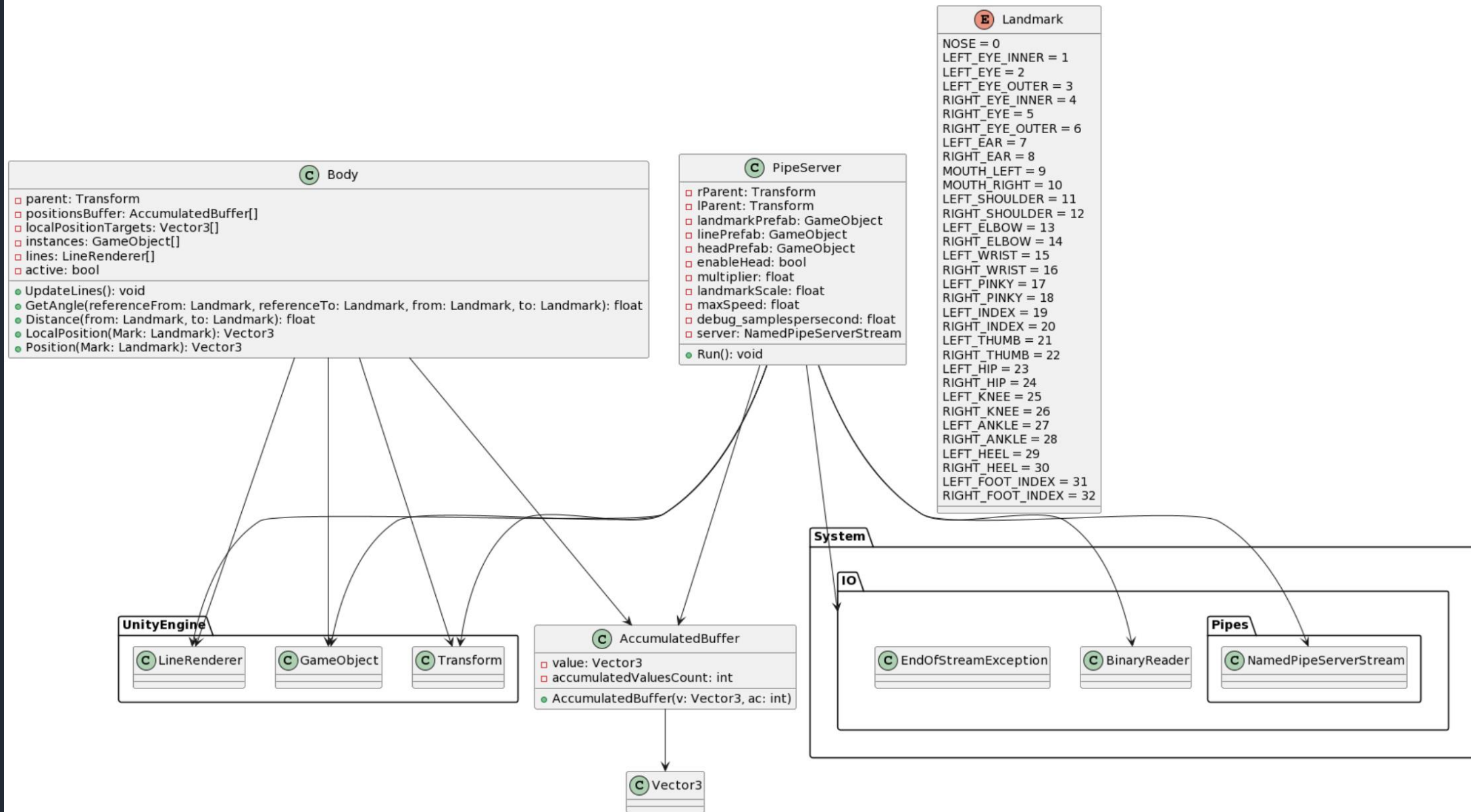


# Architecture (py)

- Libs: opencv, mediapipe, time ...
- Classes
  - CameraManager
    - Find and manage Cams
  - CaptureThread
    - Get frames from cam with opencv
  - Settings
    - Dataclass
  - Pose\_Estimation(Thead)
    - Predict Pose







The image displays a software development environment with three main components:

- Left Panel (Video Feed):** A window titled "Results pose\_estimation" showing a real-time video of a person with a skeletal pose estimation overlay. The person is wearing an orange shirt and blue shorts, standing with arms outstretched.
- Center Panel (Code Editor):** A code editor showing Python code. The code includes configuration parameters and a loop for processing data. The terminal output shows a series of numerical values representing pose data.
- Right Panel (Unity Game Engine):** A window titled "aai - SampleScene - Windows, Mac, Linux - Unity 2022.3.4f1 <DX11>" showing a 3D model of a stick figure with a pink head and red joints, mimicking the pose of the person in the video feed.

The bottom status bar indicates the current file is at "Ln 14, Col 16" and the Python version is "3.11.4 ('aai\_test': venv)".

# References

- [https://developers.google.com/mediapipe/solutions/vision/pose\\_landmarker](https://developers.google.com/mediapipe/solutions/vision/pose_landmarker)
- <https://learnopencv.com/yolov7-pose-vs-mediapipe-in-human-pose-estimation/>
- Kim, J.-W.; Choi, J.-Y.; Ha, E.-J.; Choi, J.-H. Human Pose Estimation Using MediaPipe Pose and Optimization Method Based on a Humanoid Model. *Appl. Sci.* **2023**, *13*, 2700.  
<https://doi.org/10.3390/app13042700>
- [https://de.wikipedia.org/wiki/Convolutional\\_Neural\\_Network](https://de.wikipedia.org/wiki/Convolutional_Neural_Network)
- <https://mobidev.biz/blog/human-pose-estimation-technology-guide>

