

Rozdział o wyborze modeli:

Przeprowadzano testy, mające na celu wybór najlepszego oprogramowania. Kryteriami, którymi się kierowano, były przede wszystkim: Dokładność, możliwość działania na słabszym/starszym sprzęcie, wykrywania stawów i łatwego posługiwania się ich współrzędnymi, w celu zastosowania kinematyki prostej i odwrotnej, do weryfikacji poprawności wykonywanych zadań.

Kryteriami, które również zostały wzięte pod uwagę, ale nie są aż tak ważne, to ilość FPS, podczas wykrywania obrazu (nie takie ważne, bo ruchy raczej będą powolne). Nie musi być to aż tak szybkie, bo wykrywamy ruchy osoby, o ograniczonych możliwościach ruchowych, a nie np poprawność wykonywanych ruchów tanecznych.

Po dokonaniu szeroko zakrojonego researchu, zdecydowano się na wybór modelu "Mediapipe Pose". Wstępna selekcja została przeprowadzona, poprzez sprawdzenie, w jaki sposób model pozycję człowieka. Czy nie wykrywa zbyt dużej ilości zbędnych punktów, które mogą utrudniać pracę, czy wykrywa dłonie (czy ich nie pomija, niektóre modele wykrywają tylko ramię i przedramię) Wymienić

1. AlphaPose
2. MediaPipe Pose
3. BlazePose

Dokładność została sprawdzona przez użycie każdej z w/w bibliotek, poprzez testowanie jej na dokładnie tym samym video (CO JEST NA VIDEO)

Sprzęt na którym zostało sprawdzane jak działa, to (JAKIE KOMPUTERY I PARAMETRY ICH DOKŁADNE, NAJWAŻNIEJSZE)

Wykrywanie stawów została sprawdzona, w dokumentacji (albo po prostu że samemu veni vidi moze vici), a możliwość łatwego posługiwania się nimi, to indywidualne odczucia.

Pierwsza selekcja, odbyła się, poprzez sprawdzeni każdej z bibliotek, pod względem tego, jakie punkty są wykrywane. Zależało na wykrywaniu dłoni (najlepiej z podziałem na palce) i żeby było w 3D. Tutaj zdecydowano się na rozważanie następujących bibliotek

Drugim kryterium był łatwy dostęp do współrzędnych punktów

Kolejnym kryterium, na podstawie którego wybrano biblioteki, była dokładność. Zdecydowano się zostawić 3(?) najlepiej działającego modelu

Ostatecznego wyboru dokonano na podstawie możliwości i szybkości działania, na poszczególnych sprzętach.

Przeprowadzono testy, używając dwóch metod odczytu z kamery. Nie wykorzystując wątkowosci, oraz przy jej pomocy. Operacja I/O z kamery jest bardzo wymagająca dla procesora, co przekłada

się bezpośrednio na wydajność programu. Używając wielowątkowości, podajemy do programu odczytaną wcześniej klatkę z kamery. Dzięki temu, właściwa część programu, nie musi się zajmować odczytem i dekodowaniem, co zwiększa wydajność.