20210280 안현태

Develop the first project.  
프로젝트 개관: 반복 댄스 및 그와 관련된 형태의 댄스의 녹화 및 분석, 그리고 모듈의 섞기

Define your module of dancing

여기에서 모듈의 정의는 같은 위치, 동작에서 시작해서 끝나는 하나의 단위를 의미한다.  
이를 선택한 이유는, 비록 대부분의 ‘춤’은 같은 ‘모션’에서 시작해서 끝나는 것이 적지만, 비디오게임 등에 사용되는 ‘모션’은 특정한 행동을 반복하는 형태로 형성된다. 대부분의 춤에서 이러한 형태의 모션이 드러나는 경우는 거의 없지만, 특정 형태의 반복, 그리고 그 형태의 변주를 엮어놓은 형태의 춤들이 존재하기는 한다. 비록 아이디어 자체는 다양한 형태의 춤에 적용될 수 있지만, 개인의 기술적 한계/능력적 한계로 인해 “모듈”의 시작과 끝이 어느 정도 일정하거나 수수한 형태의 댄스들을 주로 테스트하고자 한다.(에: 라인댄스)

Explain your technology

아마도 중요하게 이용해야 할 기술은 특정 모션 데이터의 시작과 끝을 섞어줄 수 있는, 그러니까 ‘보간’(Interpolation)을 해줄 수 있는 알고리즘/프로그램일 것이다. 또한, 앞에서의 모듈의 정의에 의해 비슷한 시작점과 종착점을 가지고 있는 모션 데이터를 가진다 하더라도, 사람은 완벽하지 않기에 이러한 모션 데이터의 자세가 완전히 같을 것이라 예상할 수 없으며, 이렇게 비슷한 모션 데이터를 혼합, 취사선택 함으로써 춤의 모션을 안정되게 하는 것이 주요 목표이다.

현재 이에 관련된 기술에 대해서는 더 조사가 필요하다. 우선 비디오에서 모션 데이터를 추출할 수 있는 Mediapipe와 같은 Data Generation, 두 모션 데이터의 특정 부분(시작과 끝)을 비교하고 이의 중간점을 찾을 수 있는 알고리즘, 그리고 이러한 새 모션 데이터를 보여줄 수 있게 다시 한번 적용시킬 수 있는 엔진으로 구성될 것이다.

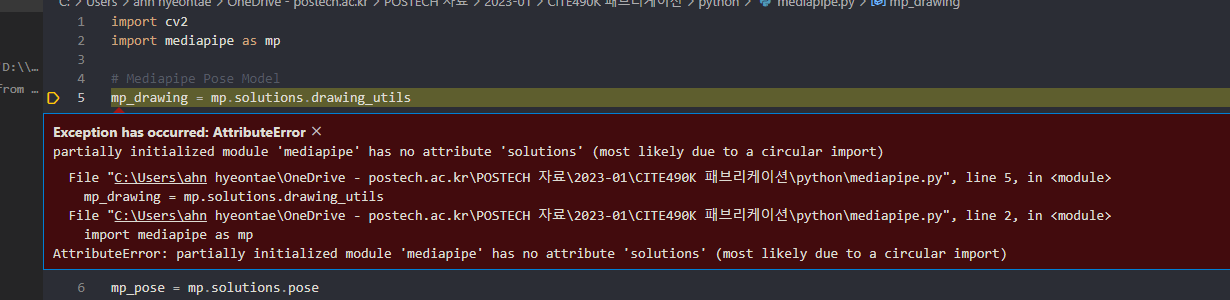
Implementation 순서:

1. Mediapipe을 python을 통해 implement하기

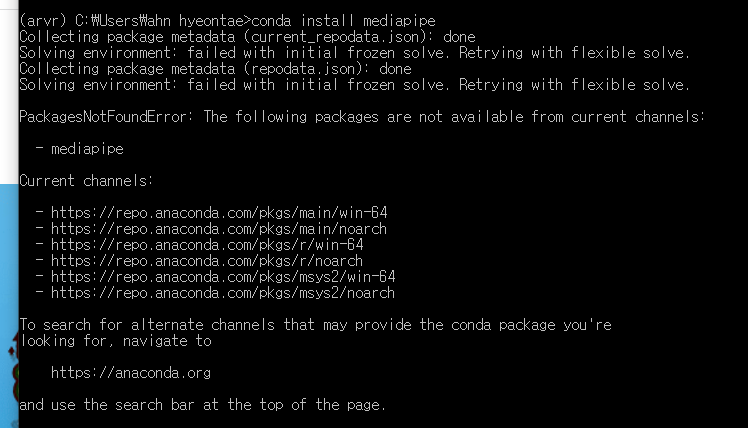
4/27 VSCode를 Python을 사용할 수 있게 수리. 이를 위해 아나콘다 재설치 및 미니콘다 삭제.

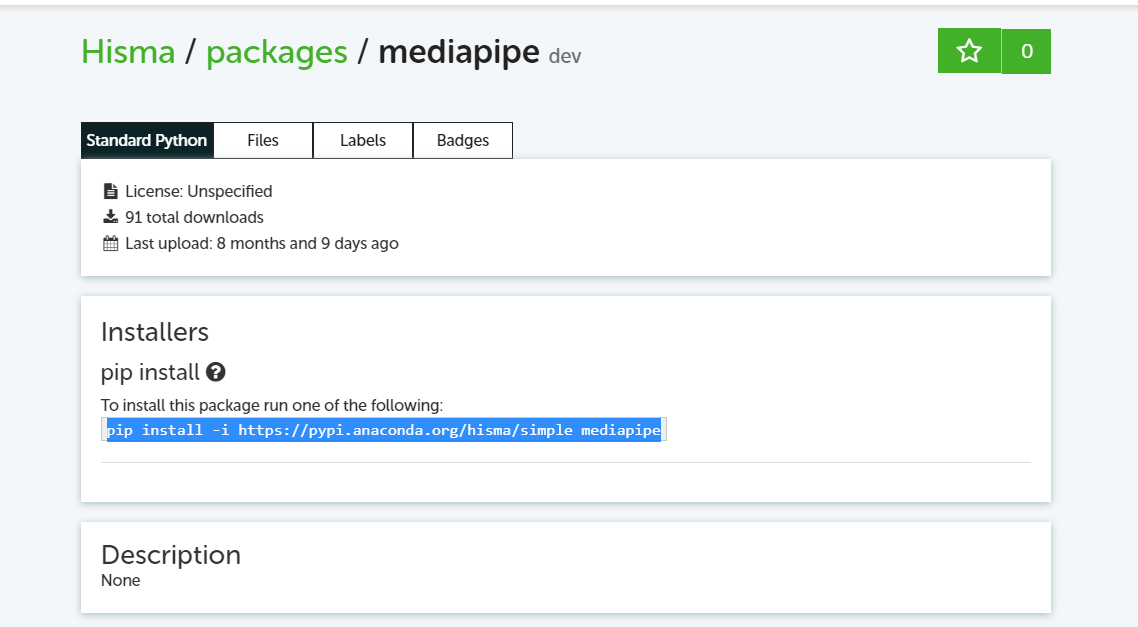
4/29 mediapipe 사용을 위한 기본 과정 재-수강

4/30 현재 mediapipe에 대해서 python 코드 실험중



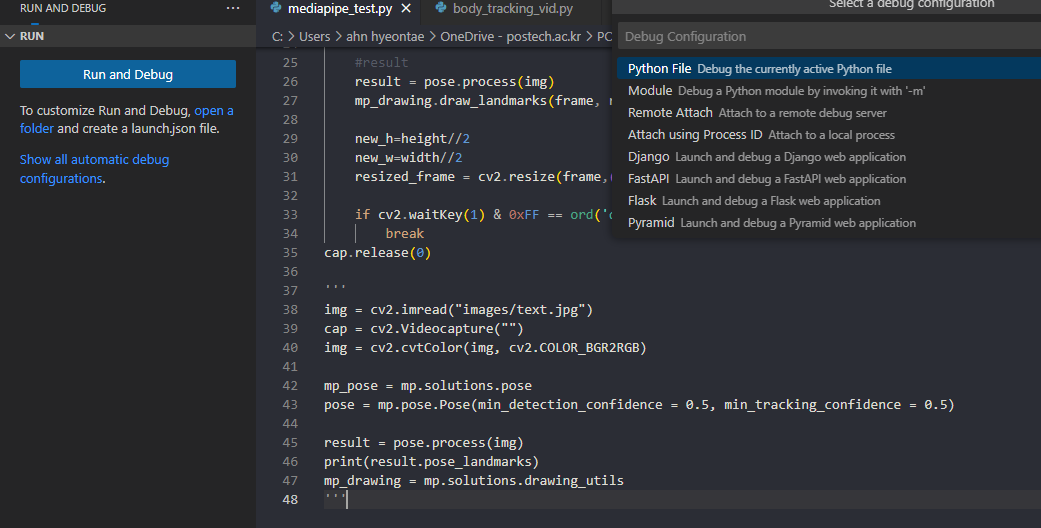
이름을 mediapipe.py로 정했기에 난 오류를 뒤로 하고 새롭게 conda env에 다운로드 하려 하였으나,

와 같이 오류가 발생함으로써 이를 고치고자 함.



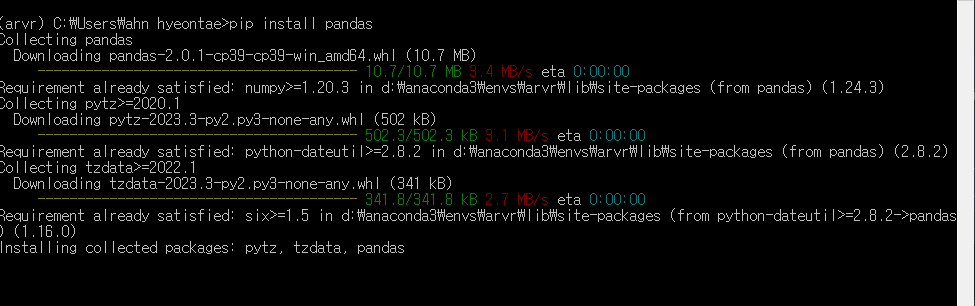
현재 이를 고치고자 하였으나 오류가 계속 나서 Rona 선생님께 질문하고자 월요일로 미룸.

5/1: 근로자의 날로 인해 Rona 선생님 만나지 못함. 또한, 문제를 해결하지 못함. Mediapipe가 아닌, 다른 Motion Detection 모델에 대해서 알아보는 중.

5/2: python -m pip install mediapipe 을 통해 mediapipe 문제 해결. 해당 코드의 동작 확인. 해당 코드의 기록을 위해, 우선적으로 이를 display할 수 있게 수정하는데 몰두. 

이를 통해 작동되는 것을 확인. 하지만, 예시 동영상에 따른 다양한 문제점들을 확인. 이를 해결하기 위해 다양한 방식들을 사용해 볼 예정

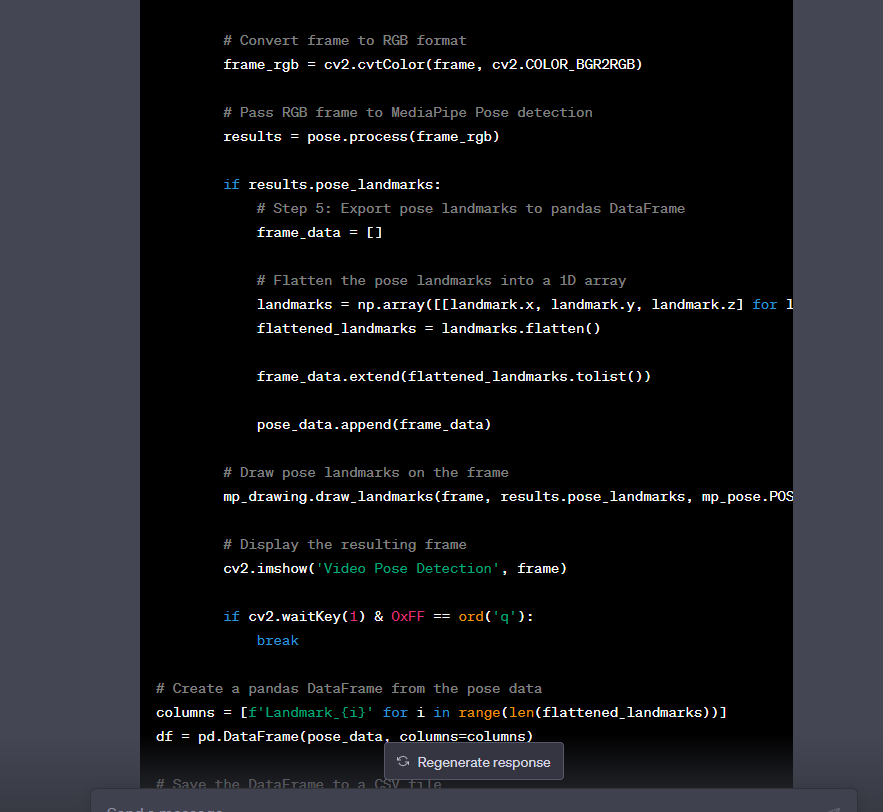
5/3: 우선적으로 관절들의 데이터를 csv 파일로 출력할 수 있도록 해야 한다. 이를 위해 예제로 주어진 ZED Project의 예시를 통해 모델을 csv 형태로 저장할 수 있도록 해본다.  
이를 위해서 pandas libarary를 다운로드한다.

  
5/8: 다운로드된 pandas library를 통해 어떻게 하면 Motion Data를 기록할 수 있는지 알아본다.

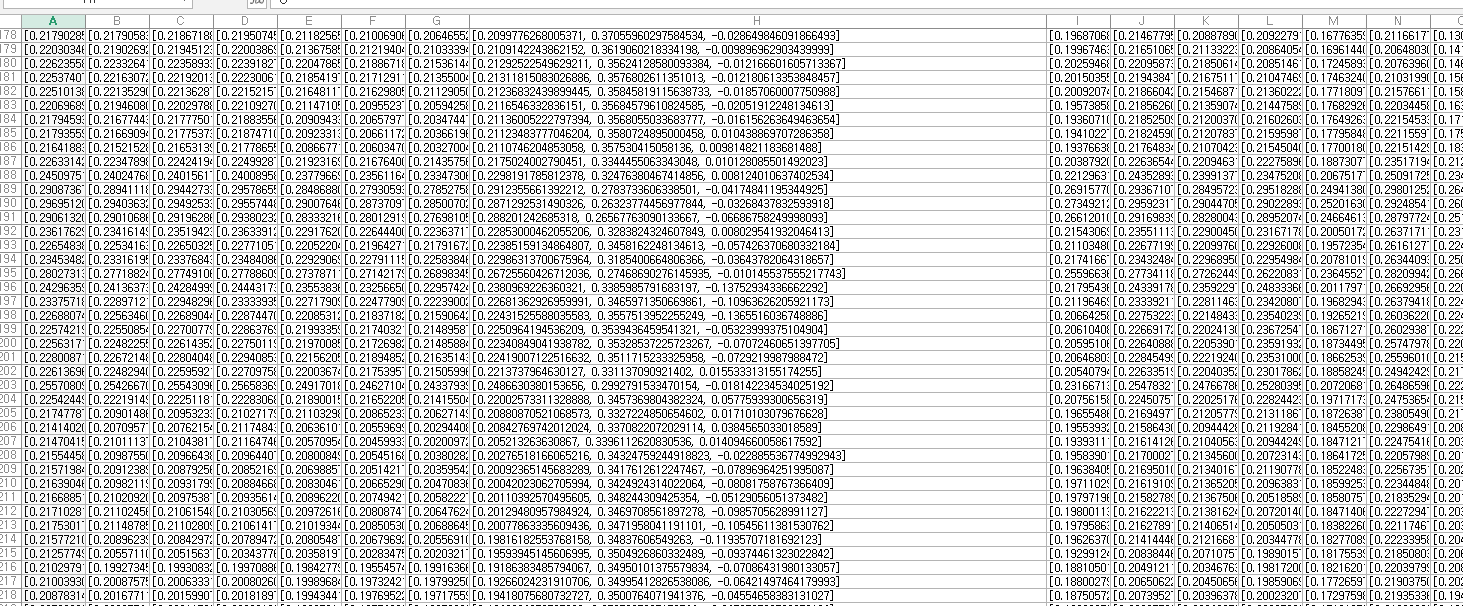
5/9: Mediapipe에서 주어지는 다른 형태의 Data Structure, Holistic에 대해 알 수 있다.

5/13: 다른 작업에 일정이 밀렸지만, Data Structure csv export 구현하기

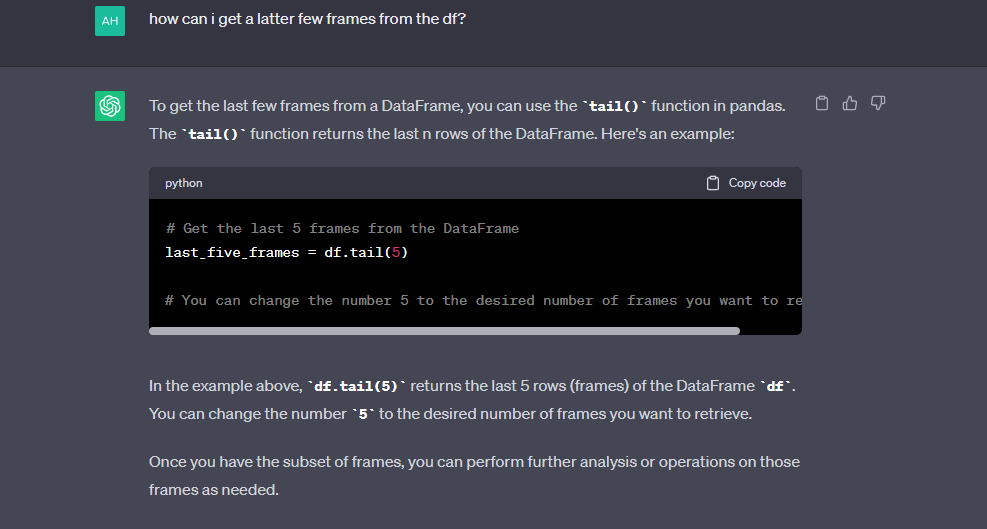
5/14: <https://www.kaggle.com/code/venkatkumar001/pose-prediction-generate-csv-keypoints-mediapipe/notebook> 참조를 통해 csv 구현, <https://github.com/google/mediapipe/issues/1803> <https://pastebin.com/BA4E1fAF> 이 더욱 도움이 됨



ChatGPT를 사용, 이를 통해 사용하는 방법 알아내는 중

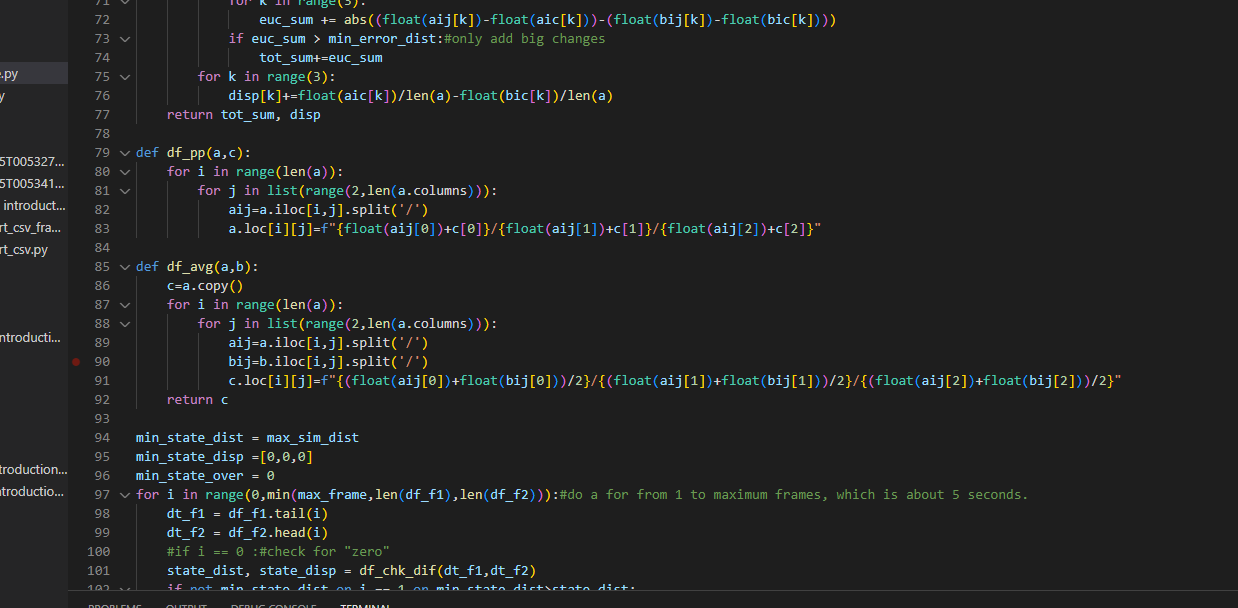


csv 형태로 추출 가능  
5/15: 해당 코드를 특정 대상에 대해서 뿐만 아니라 특정 파일 디렉토리에 있는 파일들 전체를 처리할 수 있도록 전환  
5/16: <https://developers.google.com/mediapipe/solutions/vision/pose_landmarker>   
랜드마크 마킹하기  
매 프레임 timeframe 마킹하기

5/22: CSV 모션 예제 추출, 모션 데이터 비교에 집중하기 시작  


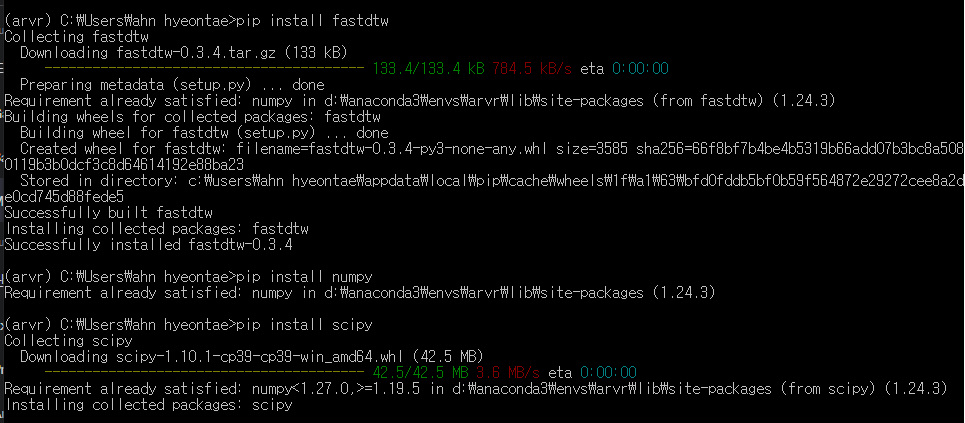
1. 추출된 모션 데이터를 다른 모션 데이터와 비교할 수 있는 방법 마련, 및 이의 중간점을 계산할 수 있도록 하기

5/24: 우선적으로 단순 중점을 찾을 수 있는 방법을 마련

5/31: 기존 방식 테스트; 

현재 CSV 파일 읽을 때 오류 발생. 이를 고치기 위해 ChatGPT와 상담 중. 이와 별개로, 최종 발표 도중에 알게 된 fastDTW(Dynamic Time Warping)의 적용을 검토

6/1: Scipy, FastDTW Install



1. A + A 후반부+B 시작부 + B 형태의 모션 데이터 만들어보기
2. 해당 모션 데이터를 Animation으로써 Unity Engine에 도입해보기

목적: