

שם: נועה וזריאס הדס
תז: 209169424

פרויקט HAND SKELETON

תיאור האפלקציה

הפרויקט הוא מערכת לזיהוי והציגה בזמן אמת של שלד יד בתלת-ממד מתווך וידאו. המערכת משלבת בין Python ל-C++, כאשר רכיב ה-*Python* משתמש ב-*MediaPipe* של Google לזיהוי 21 נקודות שלד (Landmarks) בכל יד, ורכיב ה-*C++* משתמש ב-*OpenGL* להציג השلد בתלת-ממד. שני הרכיבים פועלים במקביל ומתקשרים ביניהם באמצעות קובץ JSON שמתעדכן בכל פריטים. ממשק גרפי (GUI) ב-*Tkinter* מאפשר למשתמש לבחור וידאו, להפעיל ולעוזר את המערכת, ולהשווות את העיבוד בזמן אמת. המערכת תומכת לזיהוי של עד שתי ידיים במקביל, ומציגת את המפרקים וה"עוצמות" בצורה ויזואלית בתלת-ממד.

טכנולוגיות:

- **MediaPipe** - זיהוי של 21 נקודות שלד (Landmarks) בכל יד.
- **OpenCV (opencv-python)** - קריית וידאו, עיבוד פרימיטים, והציגת התמונה המקורית.
- Tkinter - ממשק גרפי להפעלת המערכת, בחירת וידאו, שליטה בהשניה/עיצוב. NumPy - חישובים ופעולות עזר (תלות עיקפה של OpenGL ו-OpenCV). MediaPipe - רינדור גרפי בתלת-ממד של שלד היד.
- **GLU / GLUT (FreeGLUT)** - ניהול חלון, אירועים ותצוגה.
- **lohmann/json** - ספרייה לקרוא קובץ JSON עם נתונים הנקודות.
- **C++17** - שפת התכנות לרכיב ההציגה. מערכת כלים מלאים.
- **Makefile** - בניית קומpileציה של רכיב ה-*C++*.
- **קובצי (pause.flag / done.flag)** - שליטה וסynchronization בין הרכיבים.
- **מבנה JSON** - קובץ *json* *current.json* לשיתוף המידע בין *Python* ל-*C++*.

מבנה

```
3D-Hand-Skeleton/
|
+-- assets/
|   +-- hand_video.mp4
|   +-- current.json
|   +-- pause.flag
|   +-- done.flag
|   +-- mano_models/
|       +-- mano_models.tangos
|       +-- mano_models.tangos.manob
|
+-- python/
|   +-- gui_launcher.py
|   +-- track_video.py
|
+-- src/
|   +-- main.cpp
|   +-- hand_loader.cpp
|
+-- include/
|   +-- hand_loader.h
|
+-- build/
|   +-- hand_skeleton
|
+-- Makefile
+-- README.md
```

תיקיות מסאים וקובץ עבודה #
קובץ הויזאו הנבחר לעיבוד #
עם נקודות היד (מתזען כל פריט) JSON קובץ #
וגם להשהיית העיבוד #
(Pause) (End of video) וגלי לסיום העיבוד #
(לא בשימוש טריגז) MANO קובץ מודול #
קובץ ויזואו/תמונות להזמנה

קוד Python
ממשק גרפי להפעלה, השניה ועיצוב #
JSON מפקב אחר הידיים וסימנת נקודות ל- #
C++
קוד מקור #
של שלד היד OpenGL רינדור #
והתרחן לבנה ב- JSON קראת קובץ #
פונקציית סעימת הנקודות Landmark גנרטת מכנה #
של C++ קובץ Header קובץ #
של Landmark גנרטת מכנה #
תיקיות פלט של הקומפליזיה #
קובץ ההרצה של רכיב ה- #
הוראות בניה של רכיב ה- #
תיעוד הוראות שימוש

פירוט קבצים:

gui_launcher.py •

תפקיד: משחק גרפי (GUI) ב-Tkinter שמנהל את הפעלת המערכת.

פעולות עיקריות:

בחירה קובץ וידאו והעתקתו ל-4.assets/hand_video.mp4

הרצה במקביל של: MediaPipe track_video.py (מעקב track_video.py) ו-MediaPipe hand_skeleton build (מציג build/hand_skeleton++)

השניה/המשך באמצעות יירה ומחיקה של pause.flag

יעירה וסירה של שני התהליכים יחד.

ניתור מצב התהליכים - אם אחד נסגר, סגור גם את השני.

track_video.py •

תפקיד: מודול מעקב אחר נקודות הידיים בוידאו.

פעולות עיקריות:

פתיחת וידאו מ-4.assets/hand_video.mp4 עם OpenCV. שימוש ב-Hands API ליזיהוי עד שתי ידיים (21 נקודות לכל יד).

כתיבת הנתונים (z, y, x) ל-current.json assets/current.json בפורמט JSON, בעדכון אוטומטי (temp file → החלפה).

בדיקה קובץ pause.flag - אם קיים, עוצר זמנית את העבודה.

הציג הוידאו בחלון OpenCV בזמן ריצה.

בסוף הסרטון: סגור את הוידאו, מוחק חלונות, וכותב done.flag.

main.cpp •

תפקיד: מציג גרפי ב-OpenGL שמציר את שלד היד בתלת-ממד על בסיס קובץ JSON.

פעולות עיקריות:

קריאה ל-.assets/current.json(load_hand_landmarks) כדי לטען את שמלת הידיים. שימירת הנקודות במבנה <<std::vector<std::vector<Landmark>>.

וקטור של ידיים, כל יד עם 21 נקודות.

ציור: נקודות המפרקים בירוק (GL_POINTS).

קווים לבנים בין נקודות לפי מבנה העצמות של MediaPipe (GL_LINES).

עדכון התצוגה כל 33 (~30FPS) מילס.

בדיקה אם done.flag קיים - במידה וכן, סגור את התוכנית.

hand_loader.cpp •

תפקיד: קריאת קובץ JSON והמרתתו למבנה נתונים C++.

פעולות עיקריות:

פתיחת הקובץ באמצעות ifstream.

שימוש בספריית json/lohmann כדי לפרש את המידע.

מעבר על כל הידיים והנקודות, בניית Landmark לכל נקודה (z, y, x).

החזרת מבנה std::vector<std::vector<Landmark>> לשימוש במציג.

ספריות שיש להתקין:

להורדת mediapipe, opencv, opengl דרוש להריץ:

sudo apt update

sudo apt install build-essential libgl1-mesa-dev libglu1-mesa-dev
\\ freeglut3-dev

nlohmann-json3-dev python3-pip python3-tk ffmpeg

pip3 install mediapipe opencv-python

הרצה:

לקמפל בעזרת ה makefile את הקוד:

make clean && make

להריץ את הפקודה:
python3 python/gui_launcher.py

לבחור סרטון שורצים מהGUI
ללחוץ על כפתור START