Dwight Look College of
ENGINEERING
TEXAS A&M UNIVERSITY

ECEN 404 Final Presentation
Team 72: Hand Gesture Recognition

Samuel Oncken, Steven Claypool
Sponsor: Prof. Stavros Kalafatis
TA: Pranav Dhulipala

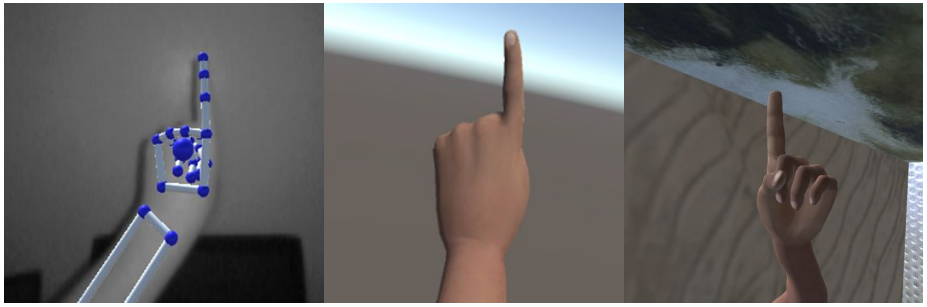
Agenda

1. Dataset Generation System Overview
2. View Replicated Synthetic Datasets
3. ML Gesture Classification Analysis/Progress - Pranav
4. ML Gesture Detection Analysis/Progress
5. Unity Robotic Arm Virtual Training Environment
6. Virtual Robotic Gripper Development

Problem Overview

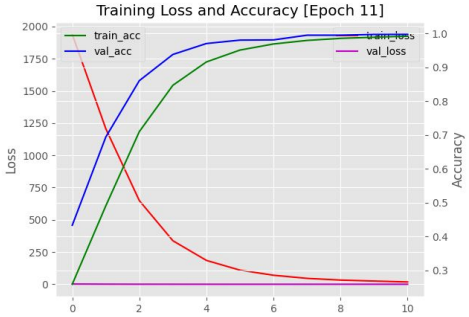
Problem Statement

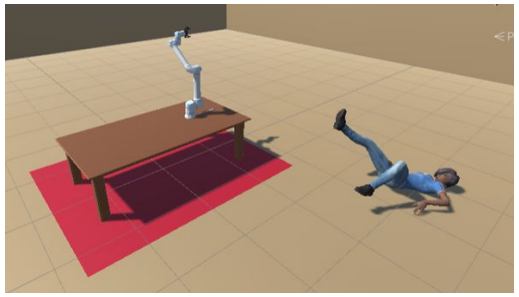
- Collecting the large amounts of data required to properly train a gesture recognition NN is **time consuming**, **resource intensive**, and **difficult to scale**



Our Objectives

- Create scalable, easy to use system that is capable of building large, **diverse** virtual datasets
- Test viability of using **virtual data** to train a hand gesture recognition neural network
  - Achieve **similar recognition accuracy** when tested against real, benchmark datasets
- Provide a **proof of concept** to extend virtual data usage to robotic arm training application

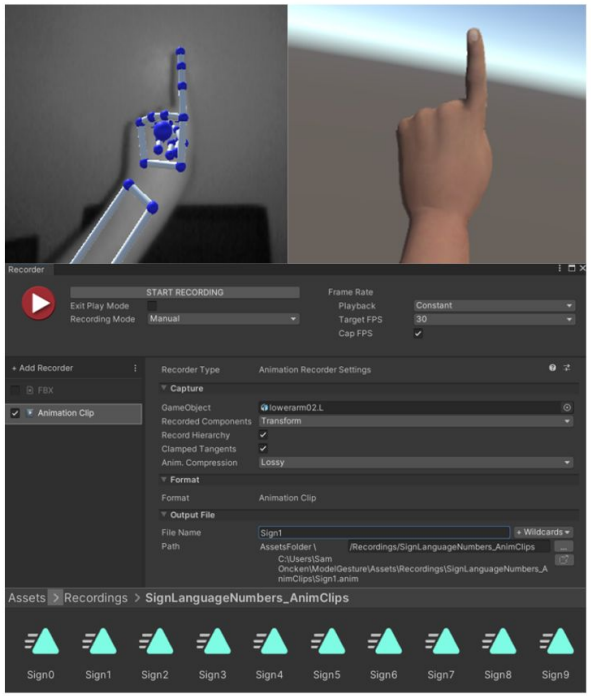




Integrated Project Diagram

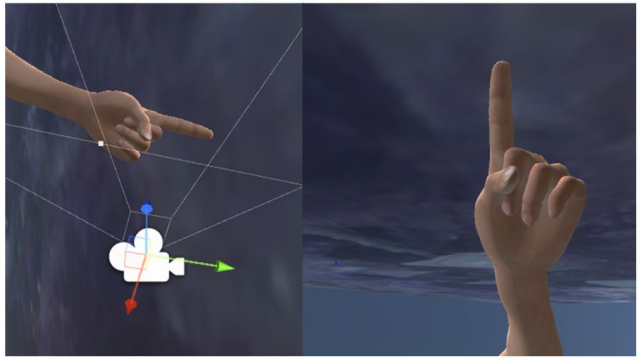
Gesture Dataset Generation Overview

Gesture Data Collection

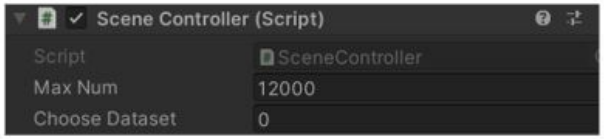


Human Model Generation & Model Animation

Data Capturing/Collection



User Control and Data Output Example



Virtual Hand Gesture Dataset Generation System

- All subsystems depicted in the previous slide have been merged and are working together in a singular environment

- Full system has been used to create 3 virtual hand gesture datasets
  - Created replicated dataset for Sign Language for Numbers - 12,000 images per gesture (132,000 images total)
  - Created replicated dataset for American Sign Language - 12,000 images per gesture (336,000 images total)

In 404:

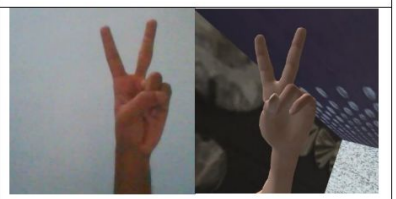
- Created HANDS dataset - 12,000 images per gesture (144,000 images total)
- Will be working to create a full body version of the HANDS dataset today for use in Steven's research (future portion of presentation)

Full System Results

Gesture "Space"

Gesture T

Gesture U



Gesture V

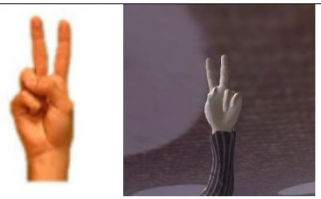
Gesture 2

Gesture 3

Gesture 4

Gesture 5

Gesture 1



Gesture 2

Gesture 3

Gesture 4

ML Classification Analysis Design Accomplishments

Accomplished:

- Extensive classification validation of real data training/testing and synthetic data training/testing for the alphabet and numbers datasets

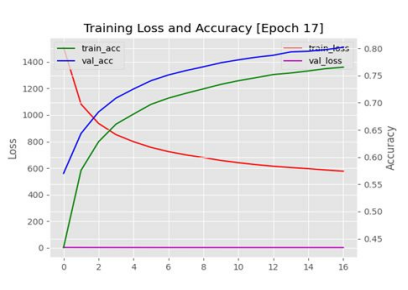
In progress:

- Validating models trained with synthetic data tested on real data

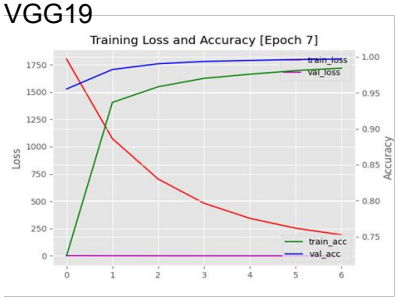
Ideal Models for Testing (successful training with real and synthetic data)

Alphabet Dataset for real (left) and synthetic (right) training

Inception

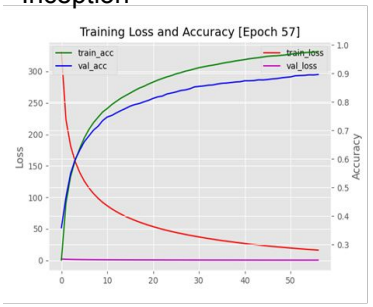


VGG19



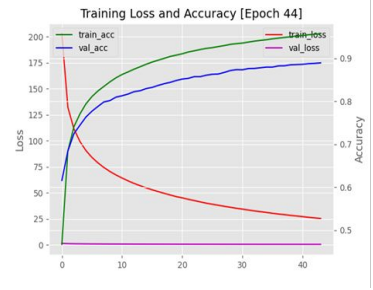
Numbers Dataset for real (left) and synthetic (right) training

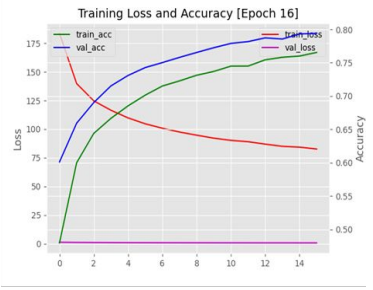
Inception



VGG16

Xception





Classification Validation

Validation Point:	Result
Train and test real alphabet and numbers dataset with acceptable accuracy metrics	Pass
Train and test synthetic alphabet and numbers dataset with acceptable accuracy metrics	Pass
Validate of models trained on synthetic data can be used to evaluate real data with similar accuracy metrics to real training and testing evaluations	In Progress

ML Object Detection Analysis Design Accomplishments

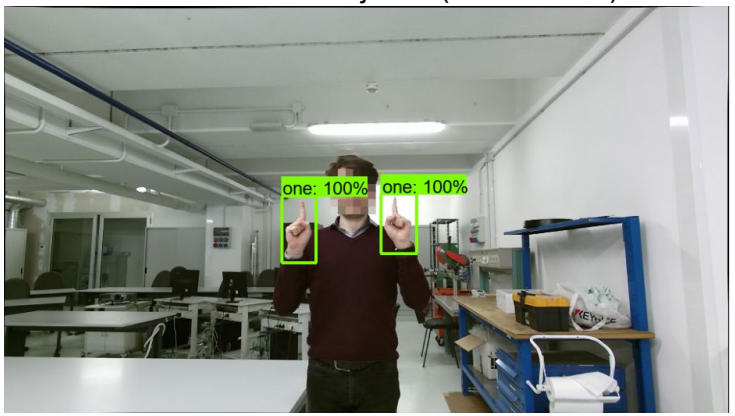
Accomplished:

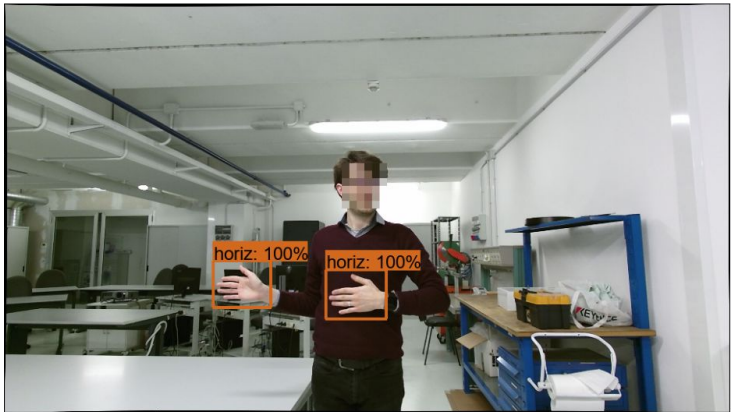
- Created a Faster RCNN for HANDS dataset using Tensorflow Object Detection API
- Real data training and testing completed and validated

In progress:

- Waiting for new synthetic HANDS dataset for training (match perspective of real data, add ground truth labels)
- Training faster RCNN with synthetic data, testing on synthetic and real data

Trained and tested on subject 1 (real HANDS)

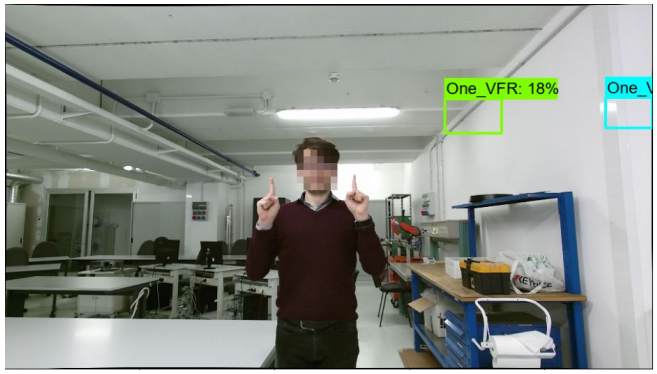


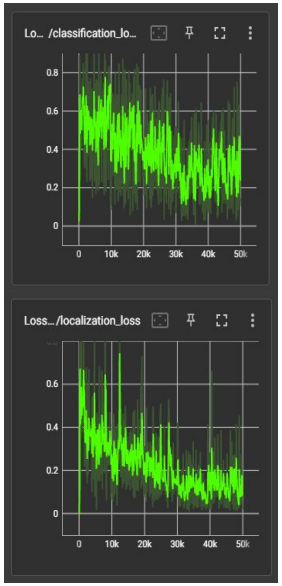


Faster RCNN Challenges

- Results with HANDS dataset showed failing localization and classification
  - attempted adjusting the model, reducing the number of classes and ground truths

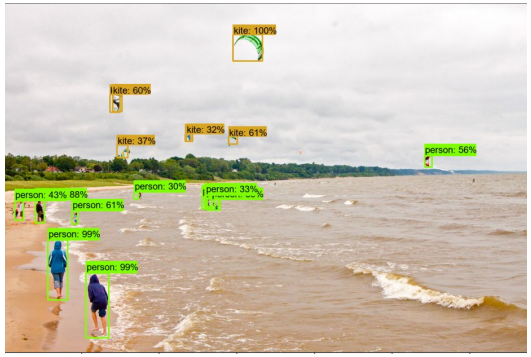
Trained/tested on real HANDS subject 1



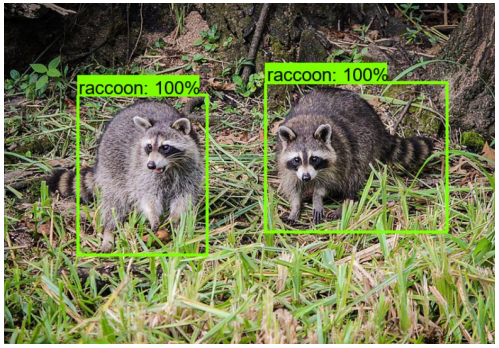


- Model showed good results with pretrained models and with training/testing on a raccoon dataset
- Raccoon dataset used different csv layout and TFRecord generation script

Tested using pre-trained RPN (COCO)



Trained/tested on raccoon dataset

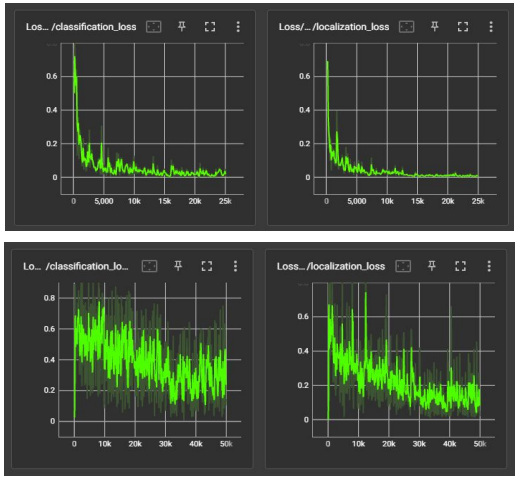


Faster RCNN Solution

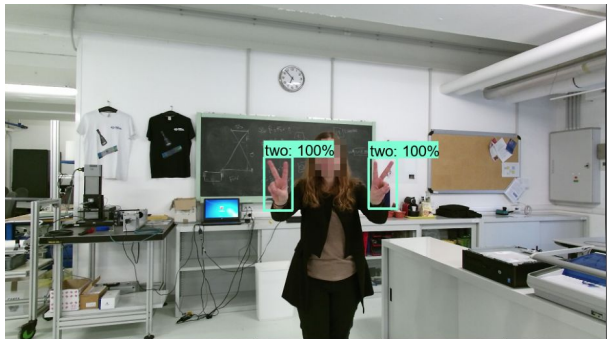
- Changed HANDS csv layout and used raccoon dataset TFRecord generator

- Model now trains to great accuracy

Training classification/localization loss comparison (new on top, old on bottom)



Trained on subject 1, tested on subject 2 (real HANDS)



Results from model trained using subject 1 real HANDS:

Tested Subject	mAP
Subject1	1
Subject2	0.65

Object Detection Validation

Validation Point:	Result
Train and test real HANDS dataset on Faster RCNN with acceptable accuracy metrics	Pass
Train and test synthetic HANDS dataset on Faster RCNN with acceptable accuracy metrics	In Progress
Validate of models trained on synthetic data can be used to evaluate real data with similar accuracy metrics to real training and testing evaluations	Untested

ML Analysis Conclusion

Classification

Changes:

- No changes to image classification analysis

Status:

- Finalizing testing of real images using model trained on synthetic data

Object Detection

Changes:

- Changed to Faster RCNN

Status:

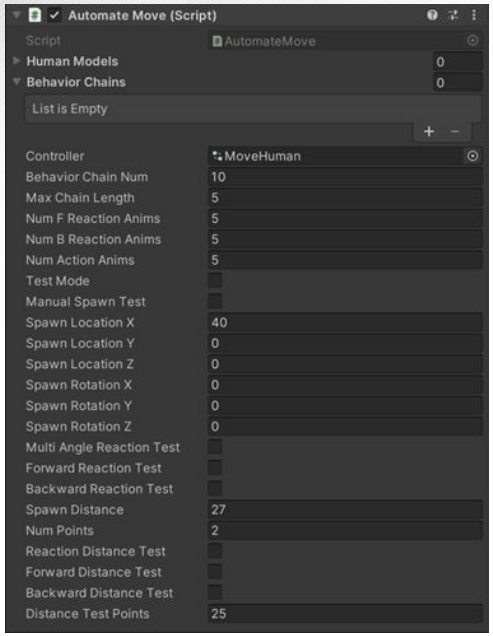
- Waiting for full body version of synthetic HANDS data for training, then testing on synthetic and real HANDS data

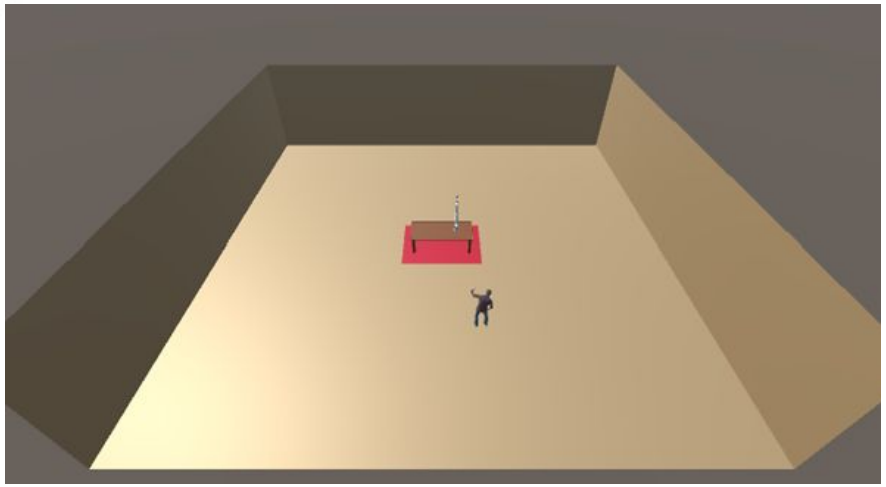
Robotic Arm Training Environment

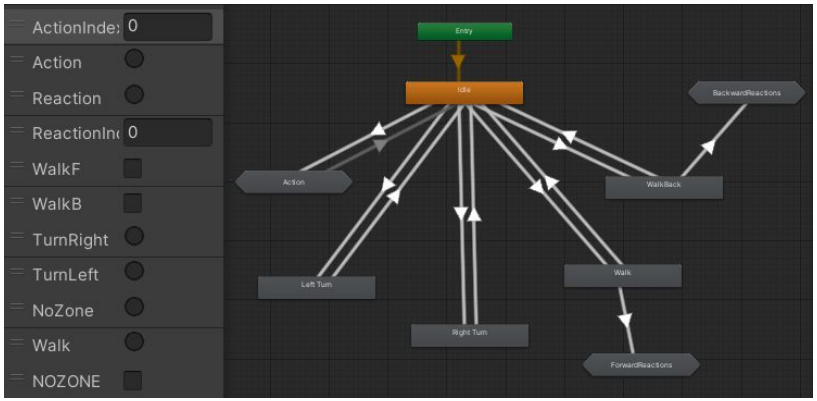
Created a fully functional system in Unity to be used in Pranav's research involving the virtual training of a robotic arm

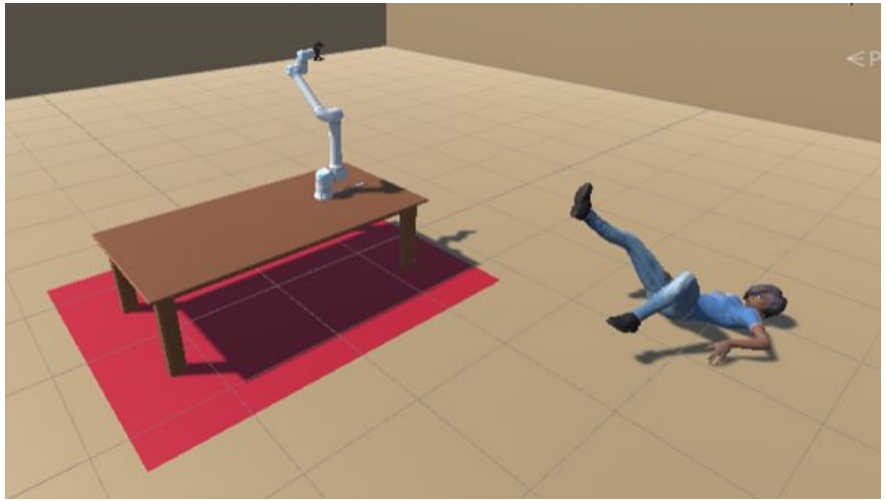
Environment Specifications:	Met?
Use of "behavior" system, where a given behavior has randomized depictions (animations)	Yes
Ability to chain behavior modules to form testing scenarios	Yes
Ability for chains of behavior modules to play automatically for future training process	Yes
Ability for characters to spawn/animate automatically at random in the testing field	Yes
If a character makes contact with robotic arm structure, halt any animation and perform instead a "reaction" behavior	Yes
Easy to adjust/customize user controls and tests	Yes
Robotic arm can be controlled as behavior chains are being performed	Yes

Robotic Arm Training Environment









Robotic Arm Training Environment Validation/Conclusion

Validation Test:	Result
All choices of action/reaction animations play at random	Pass
Behaviors in behavior chain play consecutively without overlap	Pass
All generated behavior chains play consecutively and after all are complete, Unity environment stops running	Pass
Unique MakeHuman characters are spawned at random in any location in the testing area other than the location of the robotic arm	Pass
Random forward and backward reaction animations are performed at the instant a character makes contact with the robotic arm region at any angle of approach.	Pass
User invalid input checks: Cannot have negative behavior chain number, behavior chain length, etc.	Pass
User input confirmation: Correct number of generated behavior chains, max length, etc.	Pass
Robotic arm can be manually maneuvered during runtime of training environment	Pass

System is confirmed to be fully functional for Pranav's research!

Robotic Gripper Development

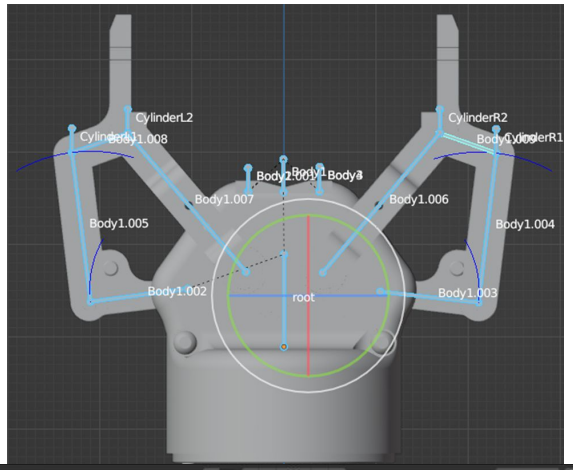
What was wrong?

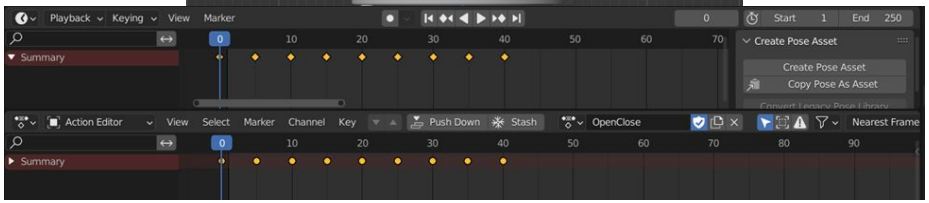
- Gripper would fly away from robotic arm
- Gripper components would separate over time
- Entire arm would rotate by itself
- Articulation joint system was used - complex

Robotic Gripper Development

My Solution:

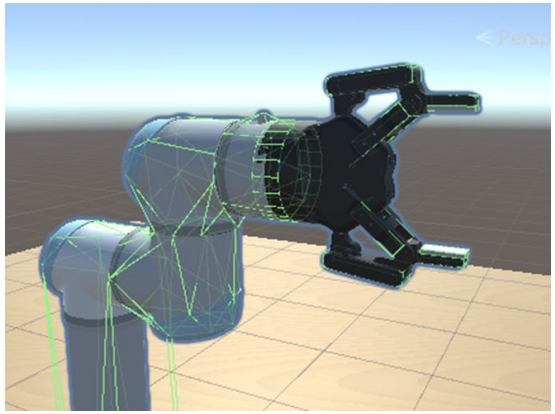
- Rig gripper in Blender, create open/close animation, export as FBX to use back in Unity

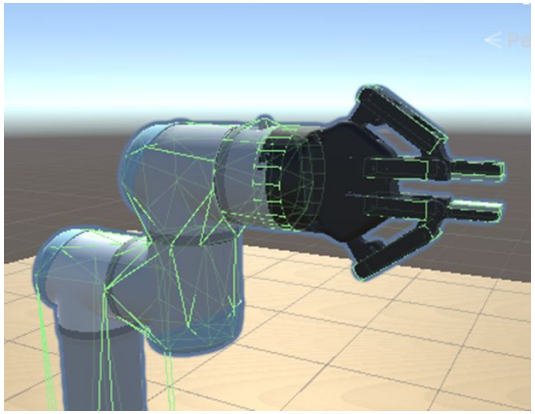


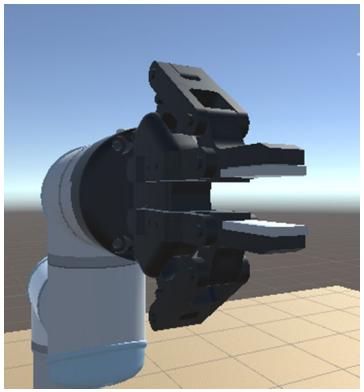


Challenge/Continuation:

- Addition of colliders so gripper can grab items







Thank you. Questions?