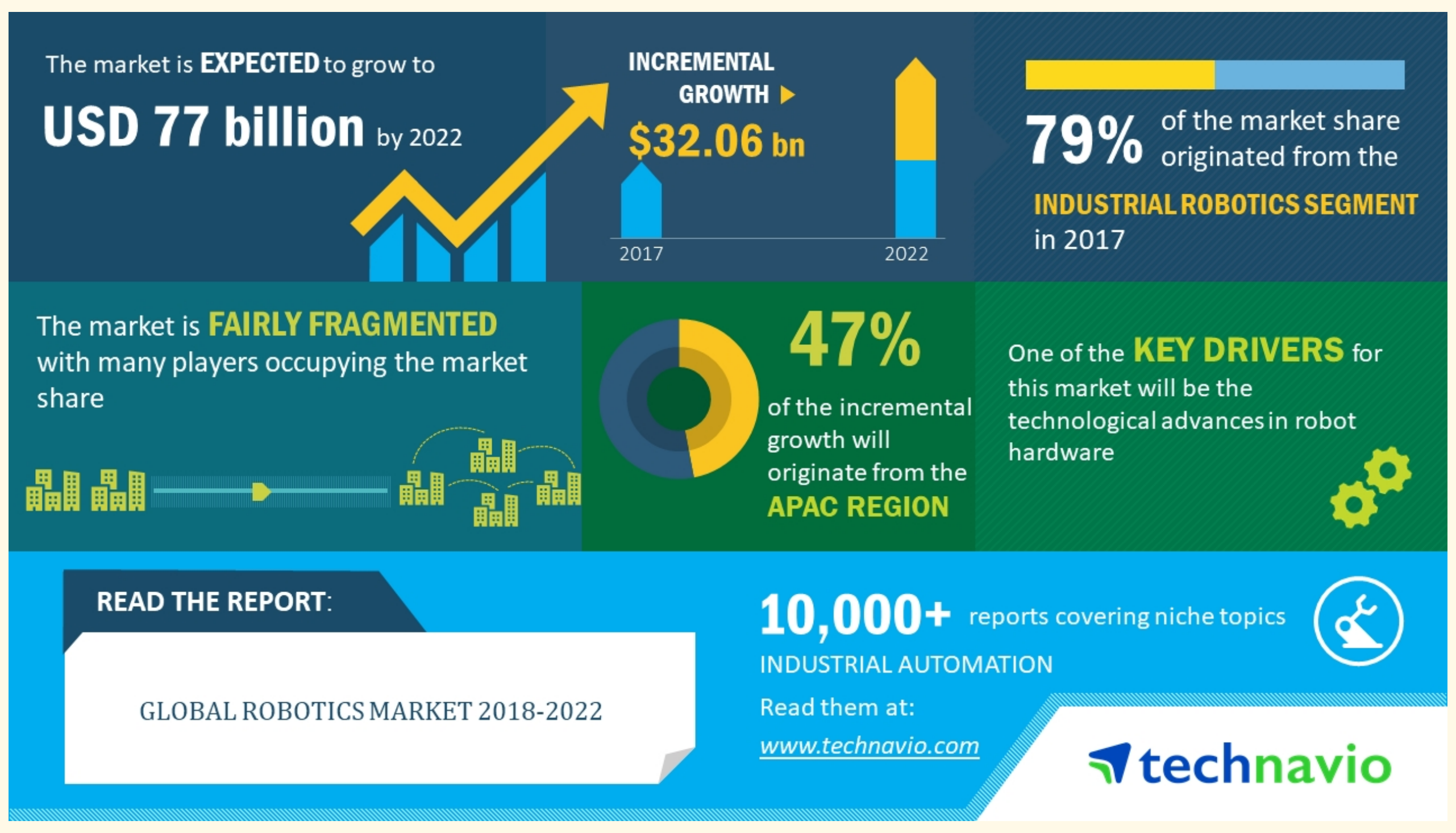
MECHANICAL ARM 3D SIMULATOR FOR EDUCATION

Shao-Chieh Lien Shristi Saraff Wei-Chen Kao Pei-Ching Lien Po-Hsun Chen

1. Introduction

- Since the development of 5G and AI, the mechanical arm and robotics markets have been growing exponentially and are expected to play an important role in every household within the next 30 years.



The market is EXPECTED to grow to USD 77 billion by 2022

INCREMENTAL GROWTH $32.06 bn

79% of the market share originated from the INDUSTRIAL ROBOTICS SEGMENT in 2017

The market is FAIRLY FRAGMENTED with many players occupying the market share

47% of the incremental growth will originate from the APAC REGION

One of the KEY DRIVERS for this market will be the technological advances in robot hardware

READ THE REPORT: GLOBAL ROBOTICS MARKET 2018-2022

10,000+ reports covering niche topics INDUSTRIAL AUTOMATION Read them at: www.technavio.com

Figure 1. Statistic of the Global Robotics Market [1]

- The development of a mechanical arm requires cross-disciplinary students to master both the software and the hardware knowledge related to the mechanical arm.
- In the future, students need to have sufficient robotic knowledge to contribute to the robotic industry.

2. Problem Statement

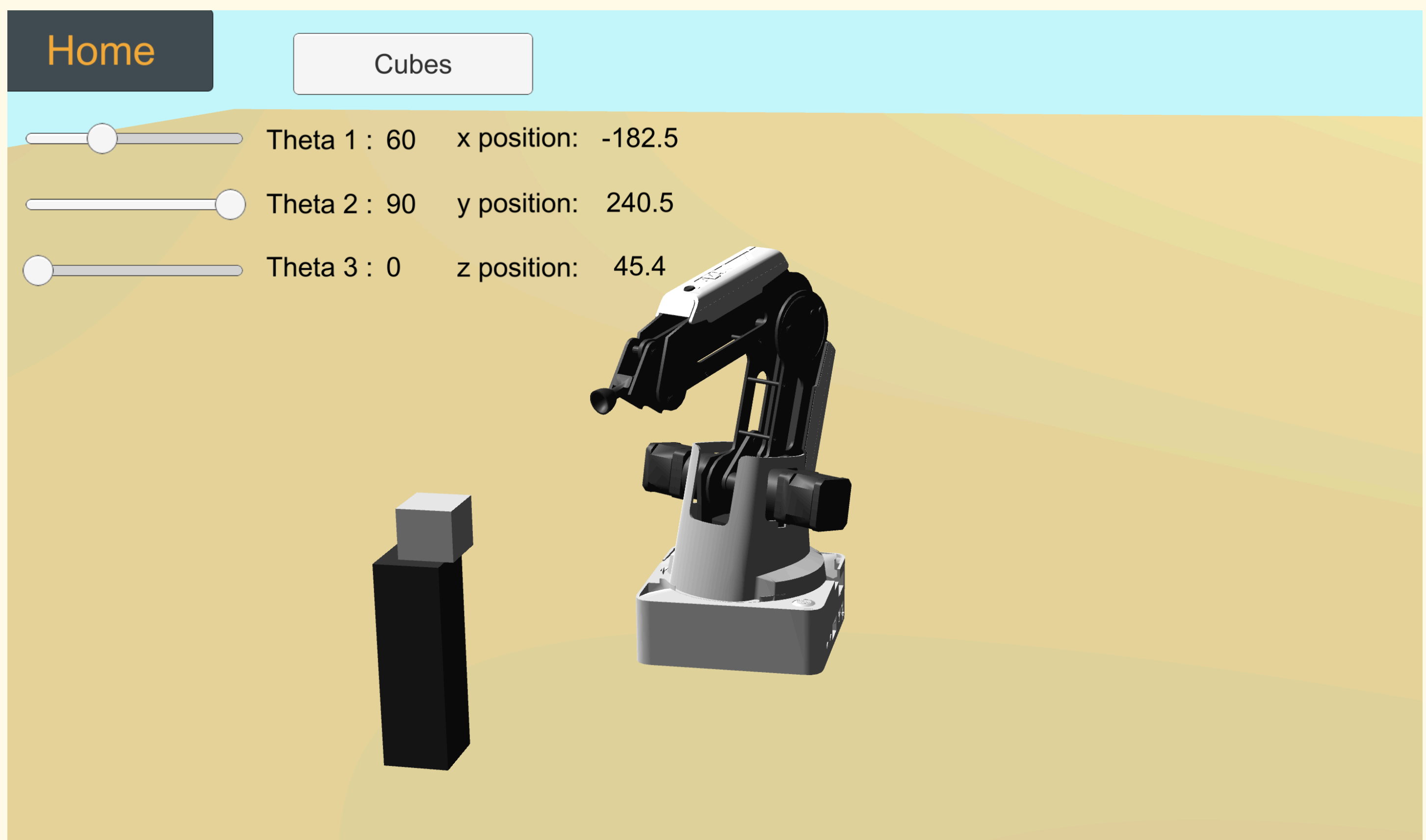
- The high cost of a mechanical arm creates a barrier for students to have access to it.
  (One industrial mechanical arm could cost more than USD 30,000)
- Students could only acquire the conceptual understanding of the mechanical arm but lack real-life experience.
- Lack of training and experience related to a mechanical arm leads to a higher rate of critical mistakes and hazard after the student enters the workforce.
- Students need a place that allows them to safely implement their robotics knowledge that are taught in school into real life.

3. Proposed Solution

- A 3D simulator that could simulate a mechanical arm and provide a GUI for students to understand the robot's architecture and provide a coding environment for students to learn robotic programming using high-level languages.
- Meet two needs in one solution:
  - Computer Science students could learn mechanical arm architecture
  - Mechanical Engineering students could learn to program robots
- Cheap and small size (<200 MB) suitable for educational use.
- Low risk associated with application.

4. Simulator

- Simulate a 3-axises mechanical arm.
- Provide three sliders that could control the rotating angle of each motor on GUI.
- Show the angle of each motor and the Cartesian coordinate position of the suction cup.
- Provide simulated objects that allow the users to test the mechanical arm's ability to grab object by the suction cup.



Theta 1 : 60 x position: -182.5

Theta 2 : 90 y position: 240.5

Theta 3 : 0 z position: 45.4

Figure 2. Screenshot of the simulator that is simulating the mechanical arm [2]

PURDUE UNIVERSITY School of Electrical and Computer Engineering
PURDUE UNIVERSITY Honors College

4.1. Technical Side of the Simulator

- Made with Unity3D and built into a macOS App.
- Separated the mechanical arm into four 3D major components, combined them with three motors and attached each object with C# script.
- The simulator served as a server that listened to the machine code sent from the coding environment and rotated the motors, accordingly.

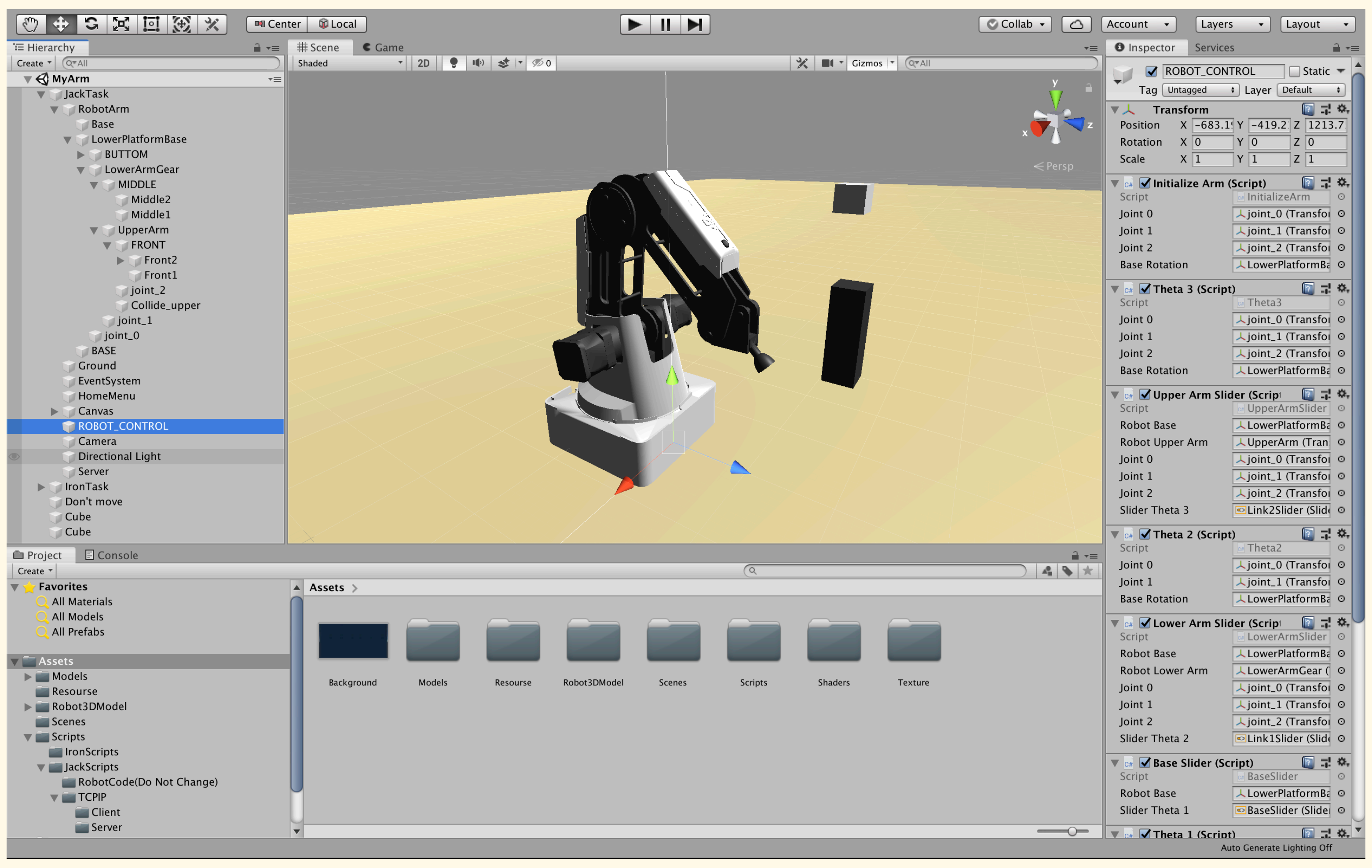
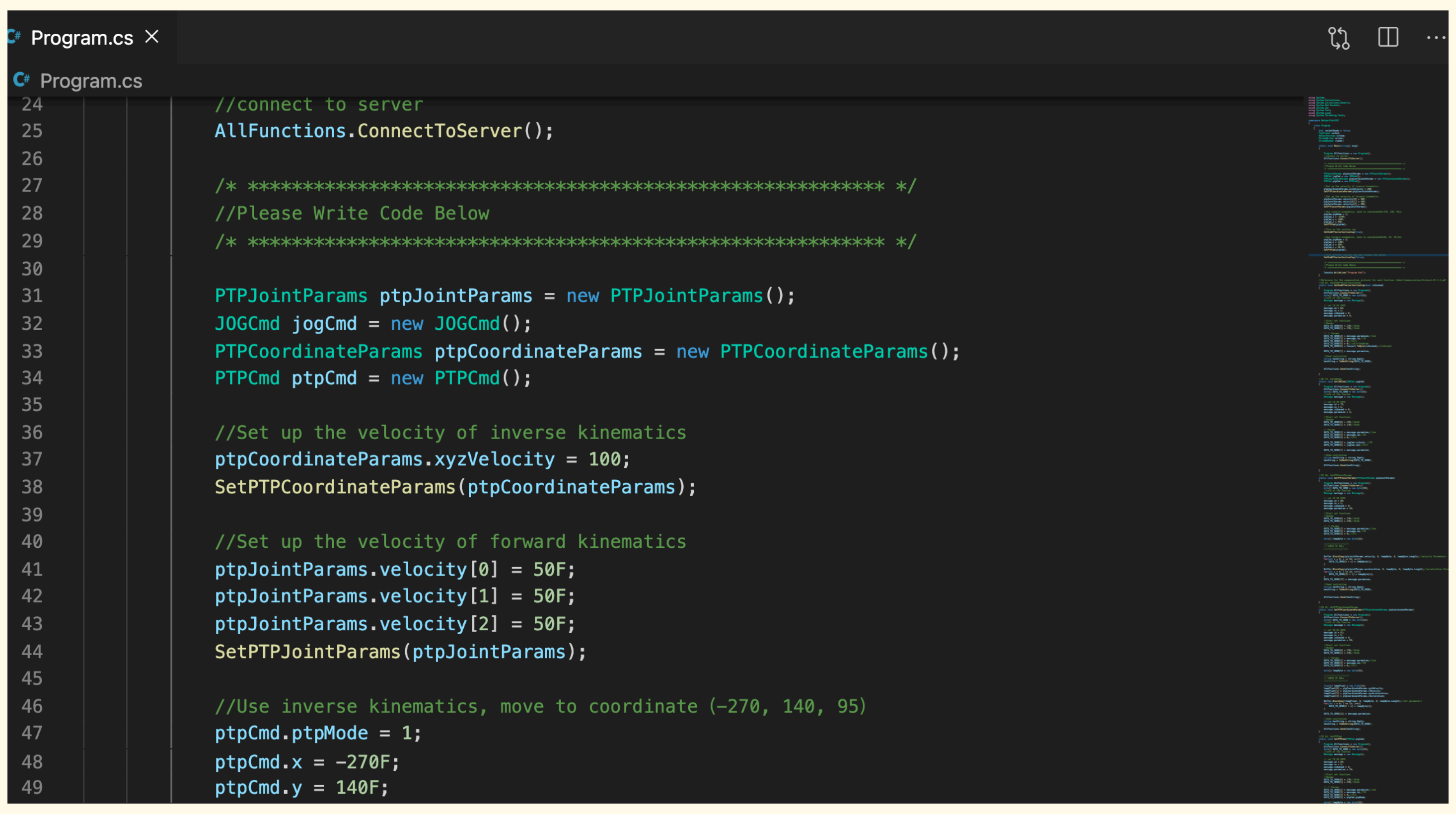


Figure 3. Screenshot of Unity3D that we used to develop the simulator [2]

5. Coding Environment

- Use Visual Studio Code as the coding environment with the .NET framework.
- Supply high-level language and Application Programming Interface(API) using C#.
- Provide Forward Kinematic and Inverse Kinematic functions that allow the users to control the mechanical arm based on motor rotation or the designated coordinates of the suction cup.
- Provide grabbing object function via the suction cup



```
            //connect to server
            AllFunctions.ConnectToServer();

            /* ********************************************************** */
            //Please Write Code Below
            /* ********************************************************** */

            PTPJointParams ptpJointParams = new PTPJointParams();
            JOGCmd jogCmd = new JOGCmd();
            PTPCoordinateParams ptpCoordinateParams = new PTPCoordinateParams();
            PTPCmd ptpCmd = new PTPCmd();

            //Set up the velocity of inverse kinematics
            ptpCoordinateParams.xyzVelocity = 100;
            SetPTPCoordinateParams(ptpCoordinateParams);

            //Set up the velocity of forward kinematics
            ptpJointParams.velocity[0] = 50F;
            ptpJointParams.velocity[1] = 50F;
            ptpJointParams.velocity[2] = 50F;
            SetPTPJointParams(ptpJointParams);

            //Use inverse kinematics, move to coordinate (-270, 140, 95)
            ptpCmd.ptpMode = 1;
            ptpCmd.x = -270F;
            ptpCmd.y = 140F;
```

Figure 4. Screenshot of the coding environment [2]

5.1 Technical Side of the Coding Environment

- Using TCP/IP protocol between coding environment and simulator connection.
- The coding environment will translate the high-level API into machine code that could be sent to the server which is hosted by the simulator.
- The communication protocol of the machine code is open-sourced and provided in the documentation for developers or contributors to extend or improve the function.

6. Future Work

- Support platforms besides MacOS, aiming for Windows, Android, and iOS platforms.
- Increase the accuracy of the simulator and aim for an industry-level mechanical arm simulator.

7. Reference

- [1] Business Wire, "Robotics Market Forecast", 2018, https://www.businesswire.com/news/home/20180911005588/en/Robotics-Market-Forecast-2x-Growth-in-Market-Size-by-2022-Technavio
- [2] Shao-Chieh Lien, "3D-Physic-Simulating-Platform", 2018, https://github.com/ShaoChiehLien/3D-Physic-Simulating-Platform