2019 FLYSET FTC Workshop

CAD Integration with Unity

(8/24/2019)

Presenter

Max Fan - FTC 8565

I have been in FIRST teams for 8 years.

- 3 years in Jr. FLL team
- 3 years in FLL team
- Currently my third year in FTC team

My role in the team is CAD lead. I have experience in PTC Creo and Fusion 360. Now I have experience in Unity.

My hobbies are reading, playing games, and watching movies.

Project Background

For Fun and Learning New Cool Skills with Unity

FIRST CHAMPIONSHIP

technicbots 8565

Project Design

Goals

- Use gamepad to control robot mechanism movement in Unity Gaming Engine
    - Use C# script to detect gamepad key pressing and manipulate the imported CAD model

Process

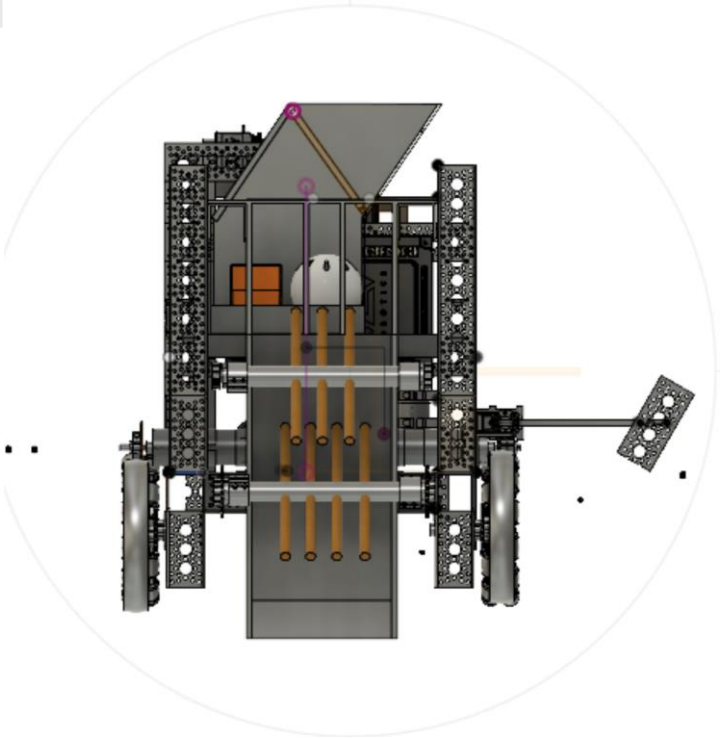
1. Export CAD model into .obj file
2. Import CAD model obj file into Unity
3. Editing imported gaming objects with proper tags so they can be manipulated in script
4. Drag the gaming objects to turn make sure they have the pivot point set correctly
5. Attach C# script to the gaming object to confirm the rotation is on the same pivot point as manual drag
6. Write the gamepad controlling code in C# to move the intended gaming object (i.e., CAD model mechanism)

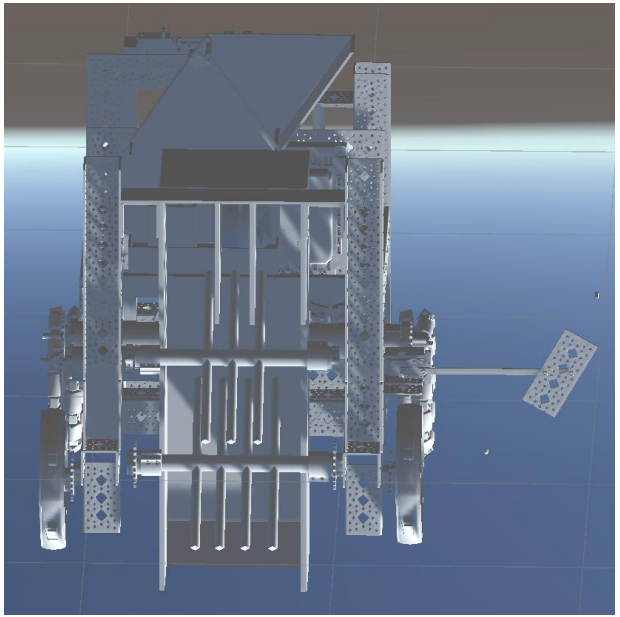
Project Results

Import CAD Models into Unity

- We chose to use our last season's robot as example (it was modeled in Fusion 360)
- At beginning of the summer:
  - Fusion 360 does not support export to .obj file
  - Two steps:
    - Fusion 360: exports to STEP file
    - PTC Creo: reads in STEP file and exports to .obj file
    - Unity: reads in .obj file
- With recent Fusion 360 update to support export to obj file
  - Single Step:
    - Fusion 360: exports to .obj file
    - Unity: reads in .obj file

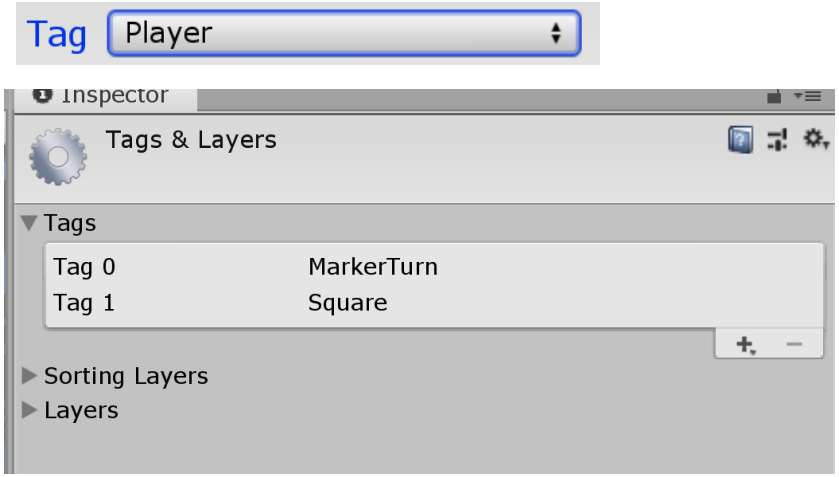
CAD Model vs. Gaming Objects in Unity





technicbots 8565

Editing Gaming Object for Tagging

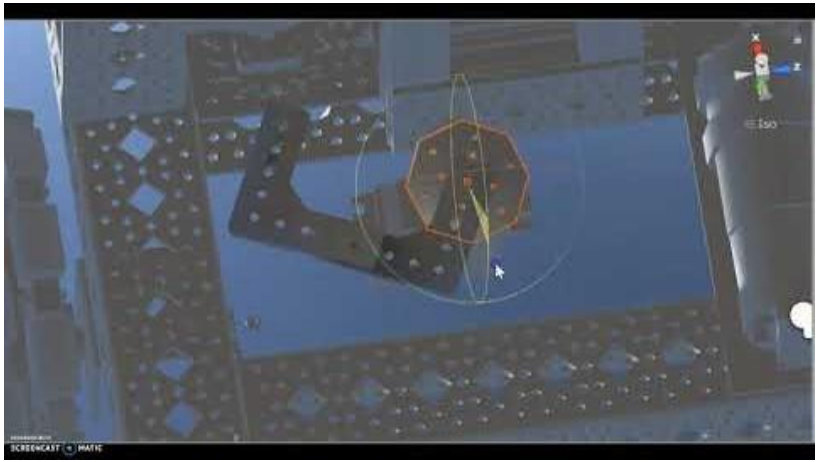


- Tagging allows for identifying certain game objects (ie. Mechanism in CAD model)
- Tagged objects are looked up in code and controlled accordingly
  - Rotation
  - Sliding

```
public Transform objectToTurn = GameObject.FindWithTag("MarkerTurn").transform;

public Transform MarkTurn = GameObject.FindWithTag("Player").transform;
```

Dragging Gaming Object Manually



- The white cube in the middle of the circle is center of object
- Arrows control movement of the object manually
- Pivot point always stays the same no matter where dragged
- Rings turn the angle of the object

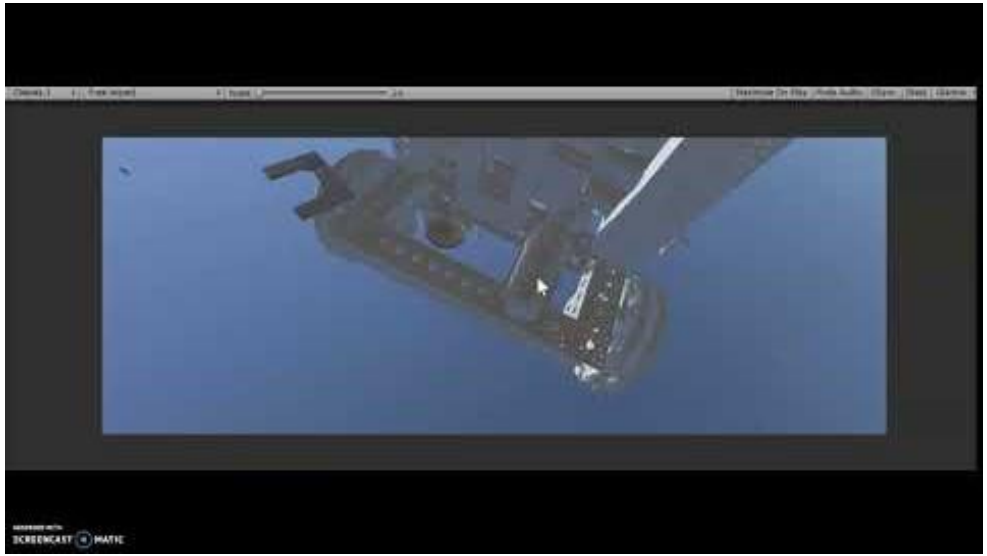
Using Script to Control Gaming Object

- Code lets us control the robot using arrow keys
  - Left arrow to rotate left
  - Right arrow to rotate right

```
if (Input.GetKey(KeyCode.LeftArrow))
   objectToTurn.Rotate(new Vector3 (0, Time.deltaTime * -50, 0));
if (Input.GetKey(KeyCode.RightArrow))
   objectToTurn.Rotate(new Vector3 (0, Time.deltaTime * 50, 0));

Vector3 originPoint = new Vector3(1.08f, -1.25f, 2.96f);
```

Script Triggered Rotation Pivot Point Issue

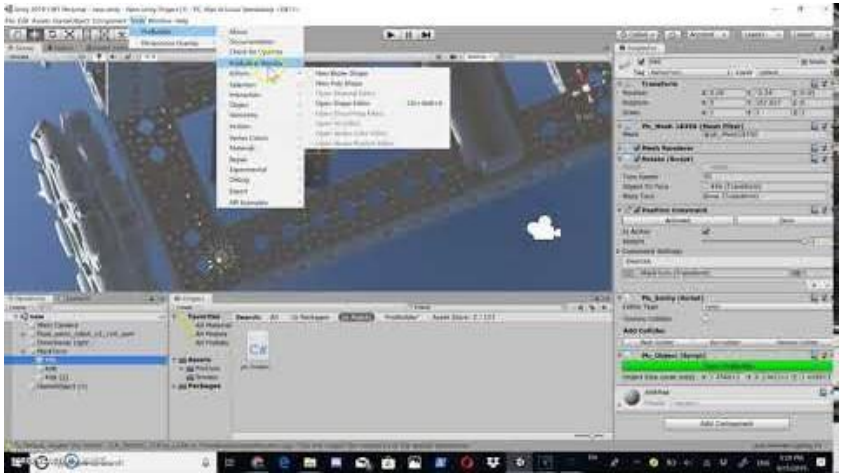


technicbots 8565

Options to Resolve the issue

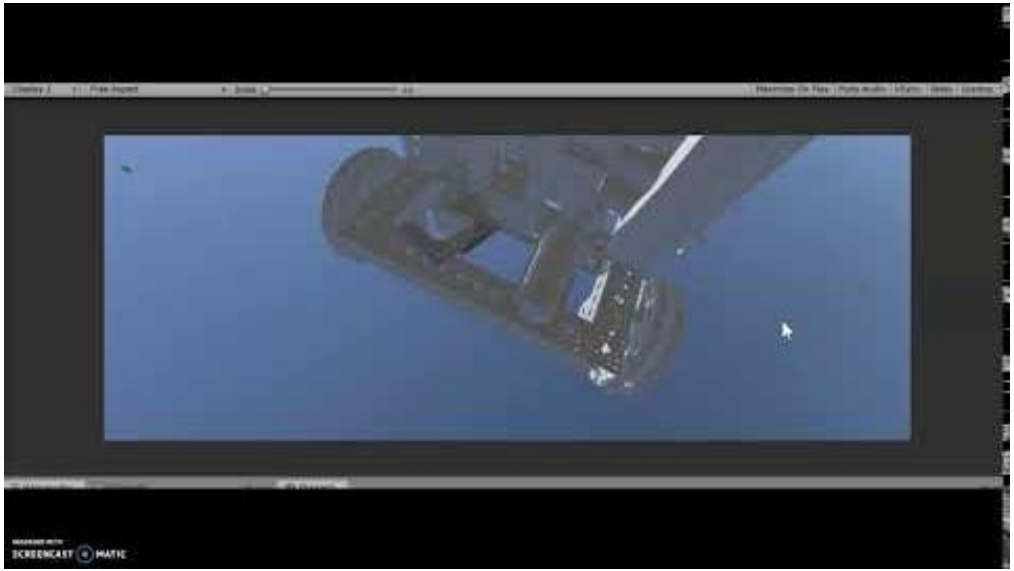
- **Option #1:** set the pivot point correctly in CAD before exporting
  - Confirmed in Fusion 360 that they are good (just after importing they changed locations)
- **Option #2:** attach the mechanism gaming object to an empty gaming object and manipulate the empty gaming object
  - Typical work around in Unity Community
  - For some reason didn't work, after attachment, manual drag works but not script triggered
- **Option #3:** Use ProBuilder to edit the pivot point of gaming object including the ones imported
  - ProBuilder is a free Unity asset even for the basic personal plan of Unity

Edit the Pivot Point Using ProBuilder



- Use the ProBuilder Window in Unity
- Select object from hierarchy to probuilderize
- Drag the object to the pivot point
- Click on “Freeze Transform” from the ProBuilder Window
- Drag the object back

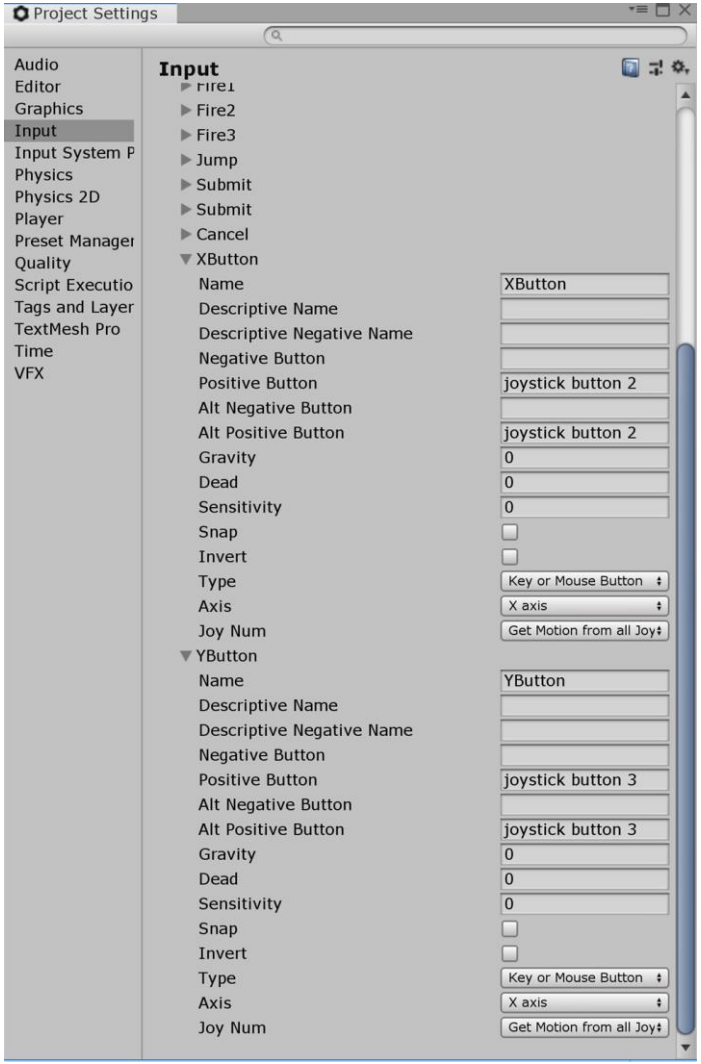
Script Triggered Rotation Pivot Point fixed



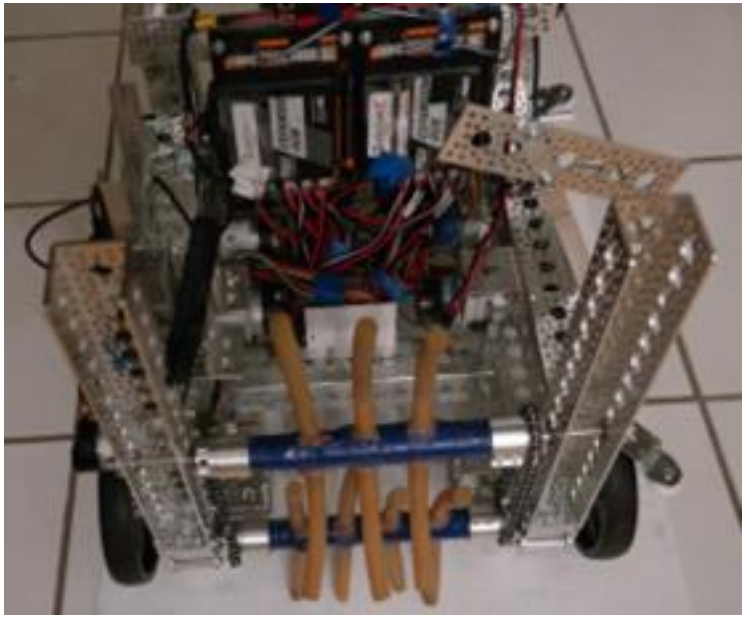
technicbots 8565

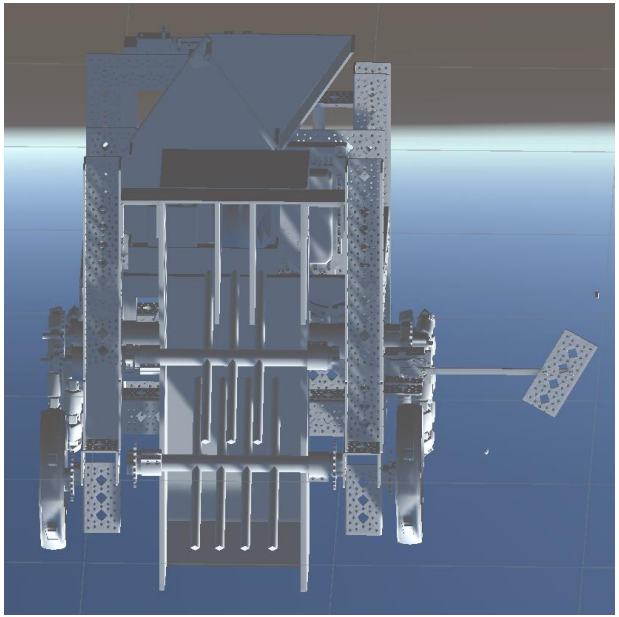
Using Script to Detect Gamepad Keys

```
if (Input.GetKey(KeyCode.LeftArrow) || Input.GetButton("XButton"))
    objectToTurn.Rotate(new Vector3(0, Time.deltaTime * -50, 0));
if (Input.GetKey(KeyCode.RightArrow) || Input.GetButton("YButton"))
    objectToTurn.Rotate(new Vector3(0, Time.deltaTime * 50, 0));
```



Live Demonstration





technicbots 8565

Conclusions

Observations

- CAD model parts and Unity gaming objects correspond to one another
- Most challenging part is the pivot point change
    - Researching the solution to fix pivot points took about 60-70% project time
- Unity forums and Unity answers are VERY helpful

Tips

- Unity updates often, keep projects in one version
- Research for assets in the Unity Store, they will help
- Search in Unity Answers community for hints and solutions

technicbots 8565

Additional Info

Fusion 360 with REV Robotics Parts

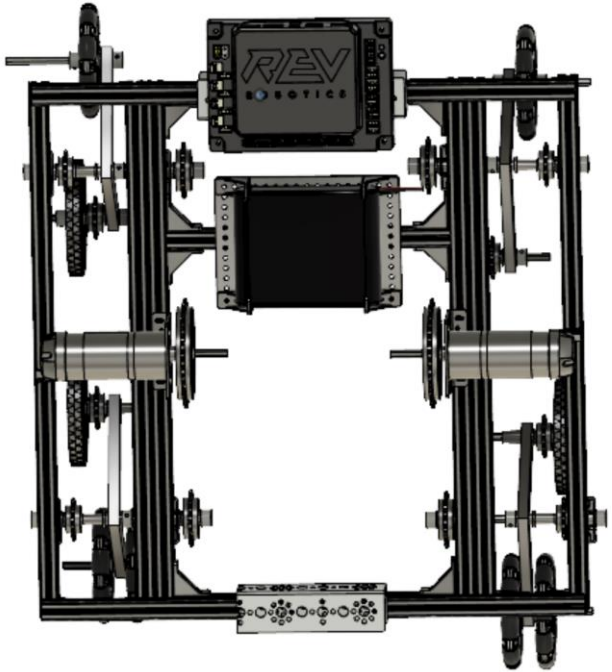
Challenges:

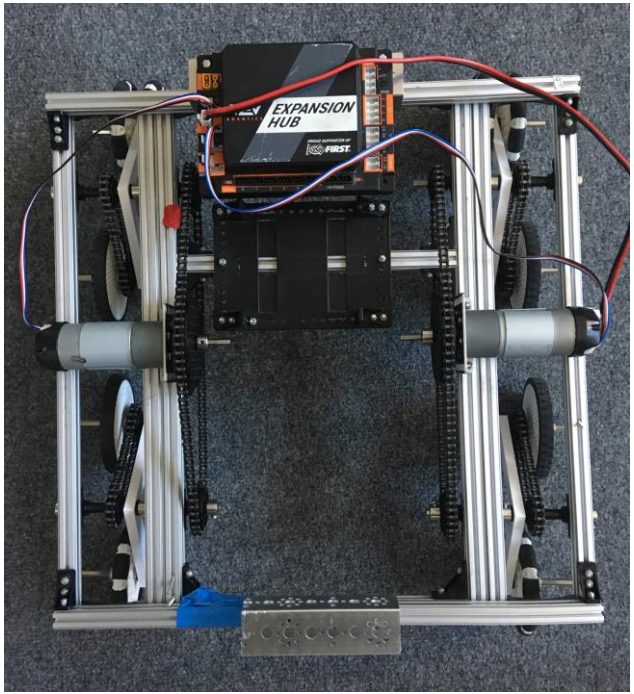
- Transition to Fusion 360 last season with help from Team 9010

Goals:

- Model our 8 wheel suspension chassis (based on REV Kits of parts) in Fusion 360 to see how hard or how easy it might be

Fusion 360 vs Real Robot





Conclusions and Tips

- Successfully created the chassis model in Fusion 360
- Not hard, only difficult to select lines
  - Shouldn’t be too hard, auto selects lines
  - Zoom in far enough to see all the lines cleary
- Can not do chains because then have to do one chain at a time
- REV pieces different than others

Questions?