



# FLL Workshop – Session 3

## Chassis and Attachments

Patrick R. Michaud  
[pmichaud@pobox.com](mailto:pmichaud@pobox.com)

University of Texas at Dallas  
September 2017



Robot design and game strategy

# Chassis and attachments

The *chassis* is the part of the robot that is responsible for navigating the field and providing a base for attachments

*Attachments* are the things added to the chassis to solve missions and manipulate models

Design is about creating a chassis and attachments that will perform well in the Robot Game

## Consistency wins

Good programming and strategy are essential to consistently good performance

Programming overcomes the limitations of the hardware

Great robot + poor strategy → inconsistent scores

Fair robot + good strategy → consistent scores

## Robot Game Strategy - Base

The robot must always start from Base

Base is the only place where changes can be made

# Robot Game Strategy - Time

Matches are 2:30

When the Robot is in Base, it's not scoring

- minimize time spent in Base
- don't lose time positioning or aiming robot
- make attachment changes quickly
  - ♦ or have them permanently on the robot

Travel on the field takes time

- minimize time spent moving from place to place
- solve multiple missions in the same region

# Robot Game Strategy - reliability

## Distance:

- Error increases with distance

  - 1 degree is 1.7cm error after 100cm

- Missions that are close become easier

- Missions that are far become harder

- Use field elements (lines, walls, models) to guide the robot to make things seem “close”

## Size:

- Large targets are easy to hit

- Small targets are hard to hit

- Use large attachments to make small targets “bigger”

# Robot game strategy - humans

The Robot does exactly what physics and programming say to do

Humans (drivers) make mistakes and are inconsistent

Design the robot and strategy to prevent human mistakes

- *Always* start robot from same location
- Don't require humans to aim
- Build safeties into robot
- Robot should adapt to humans, not vice-versa

# Republic of Pi's design mantra

Whenever the robot or humans  
make a mistake in scoring,  
redesign the *robot* so that mistake  
*cannot* happen again.

# Tip: Start every mission from same spot

Put solid edges on robot

Align robot with solid edges,  
not by sight-aiming

Robot can always start with  
known location and heading

Faster setup in Base  
between mission runs



Place flat edge  
against wall

Pick a marking  
to align robot

To save match time,  
always start from  
same spot

# Navigation

A key to scoring is to move robot consistently

Things a program(mer) needs to know to navigate:

- Where the robot currently is

- How precisely you know where it is

- Where the robot is going

- What's in the way, or what can guide you there

Robot needs to be able to move in a straight line



# Chassis design

# Common FLL robot characteristics

Two motors for drive wheels - one for each side

Multiple attachments for different missions

Attachments may be passive or powered

- Third and fourth motors can be used for power

- Maximum of four motors allowed during match

# Robot design characteristics

From the Robot Design judging rubrics:

		Beginning	Developing	Accomplished	Exemplary
Mechanical Design	<b>Durability</b>	Evidence of structural integrity; ability to withstand rigors of competition			
	N D	quite fragile; breaks a lot	frequent or significant faults/repairs	rare faults/repairs	sound construction; no repairs
	<b>Mechanical Efficiency</b>	Economic use of parts and time; easy to repair and modify			
	N D	excessive parts or time to repair/modify	inefficient parts or time to repair/modify	appropriate use of parts and time to repair/modify	streamlined use of parts and time to repair/modify
	<b>Mechanization</b>	Ability of robot mechanisms to move or act with appropriate speed, strength and accuracy for intended tasks (propulsion and execution)			
	N D	imbalance of speed, strength and accuracy on most tasks	imbalance of speed, strength and accuracy on some tasks	appropriate balance of speed, strength and accuracy on most tasks	appropriate balance of speed, strength and accuracy on every task

Evaluate the robot:

Does the robot break often?

Does it seem solid? Does it have a lot of “flex”?

Do the wheels make good contact with the surface?

Does it perform well in the game?

# Chassis design considerations

The *chassis* gets the robot from place to place

## Size

- Smaller robots are easier to navigate

- Robot must fit completely in Base when starting

## Consistency and reliability

- Robot needs to act consistently when moving

## Speed

- Faster robot → time to solve more missions

- Slower robot → more consistent and accurate

# Chassis basics

Good motor and wheel design are key to consistency

Motor and wheel frame needs to be solid with very little “flex”

“flex” produces inconsistent runs

Use cross bracing, frames, and angle beams to increase structural stability

# Wheels

Wheel selection is important

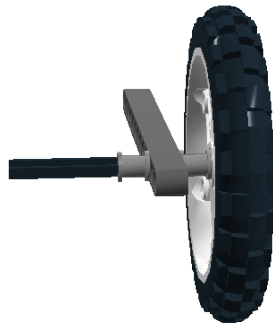
Larger wheels are faster, but may be less accurate

Tire shape, pattern, and field mat surface affect traction and consistency

Wheels that “slip” on the mat produce inconsistency

# Wheels

Wheels should be mounted close to supporting beam (but not rubbing against it):

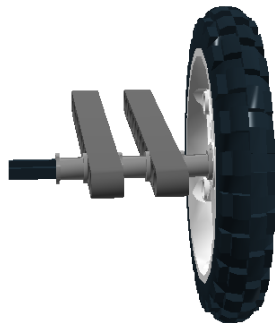


Good

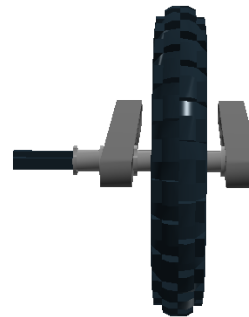


Poor

Axles do better when supported by at least two beams. Beams on both sides of wheel are best.



Better

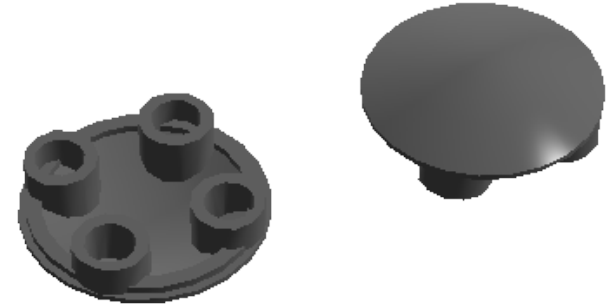


Best

# Wheel styles

## 2 wheels and skid(s)

Works great, may have difficulty with ramps/obstacles



## 2 wheels and caster

Caster wheel will make driving inconsistent

## 2 wheels and ball pivot (3-point design)

Works fine, may be a little unstable



## 2 wheels and 2 balls (4-point design)

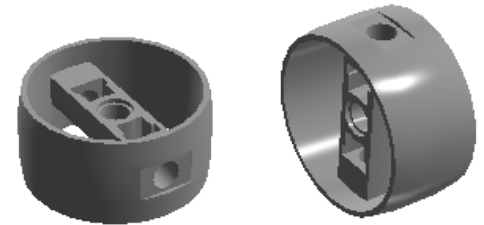
Very nice

# Wheel styles continued

## 4 wheels (4-point design)

Make sure non-driving wheels can slide while turning

#41531 Turbine has worked well for my teams



## 6 wheels

Stable, but generally quite large and turning may be imprecise

## Tracks (tanks)

Good for obstacles, hard to predict turns

## Exotics

## Balance and center of gravity

Balance and center of gravity affects stability and consistency of robot

Center of gravity is the average location of weight of the robot

If the center of gravity is outside the wheelbase, the robot will tip over

High center of gravity will make robot more likely to tip

# Weight

Heavier robots are more accurate, but slower and use more battery

Try to keep weight over driving wheels (but watch the center of gravity!)

# Other chassis considerations

Put solid edges on robot

Align robot with solid edges,  
not by sight

Robot can always start with  
known location and heading

Provides attachment  
mounting points

Can be used for wall navigation  
and aligning with mission models



Place flat edge  
against wall

Pick a marking  
to align robot

To save match time,  
always start from  
same spot



# LEGO structures and pieces

# Beams

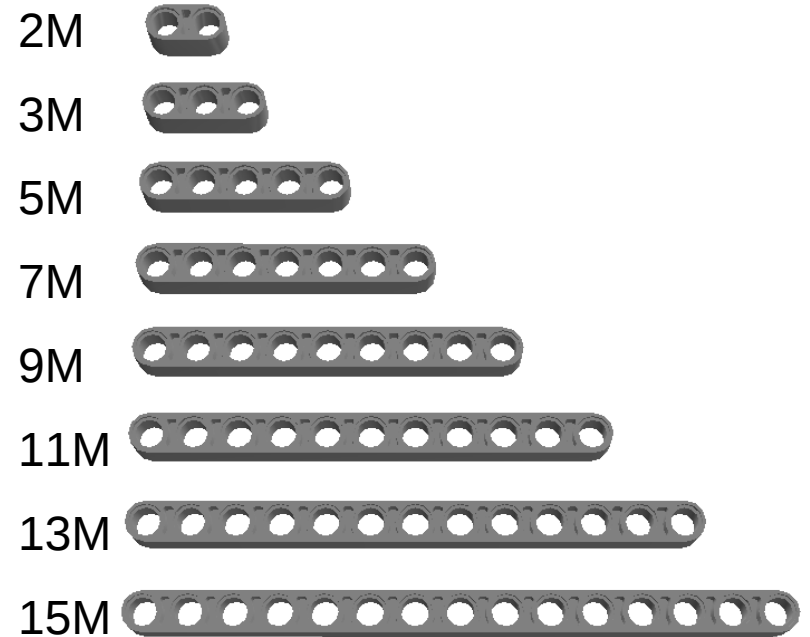
Beams are the basic building pieces for most LEGO robots

Length of beam determined by number of holes

Often called “M” or “L” units

Center-to-center distance is 8mm

Can be “thin” or “thick”



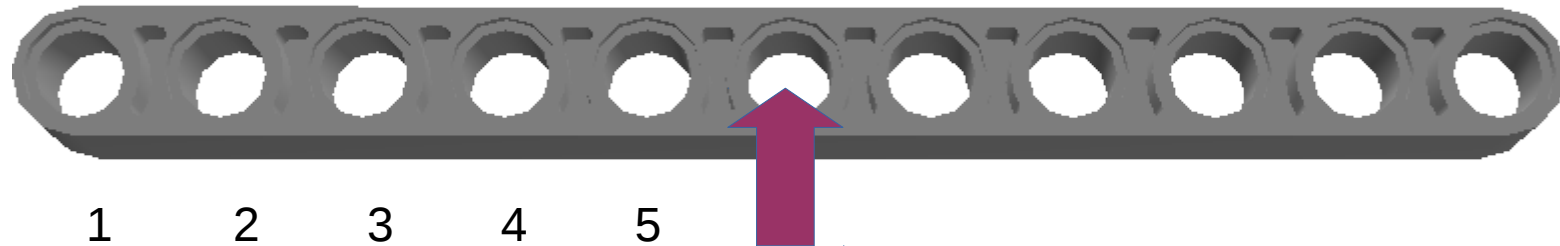
# Quickly determining beam size

To quickly determine the size of a beam

Place a finger over the center hole

Count the holes on one side

Double that and add one



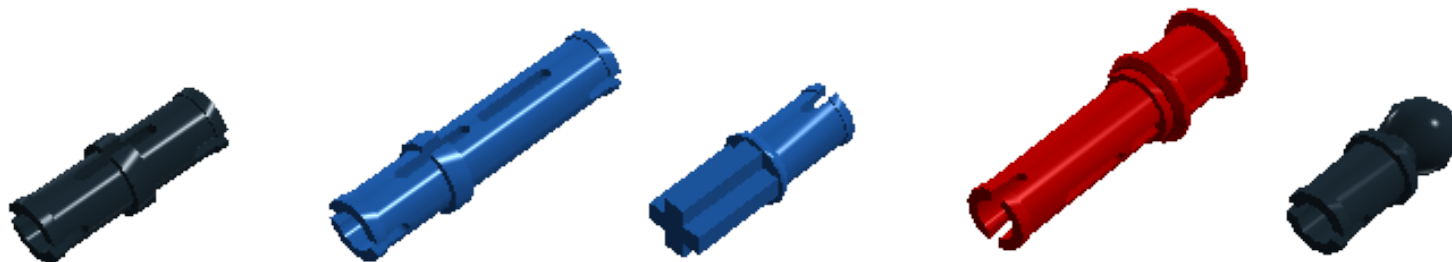
# Pegs

Used to connect beams and other components

Fit inside beam holes

Friction pegs do not turn freely in holes

- Connector peg with friction (“peg”)
- 3M connector peg with friction (“long peg”)
- Connector peg with cross-axle (“axle peg”)
- Connector peg with cross-hole (“bushing peg”)
- Ball with friction snap



# Pegs

## Non-friction pegs will turn in beam holes

- ♦ Connector peg
- ♦ 3M connector peg
- ♦ Connector peg cross axle

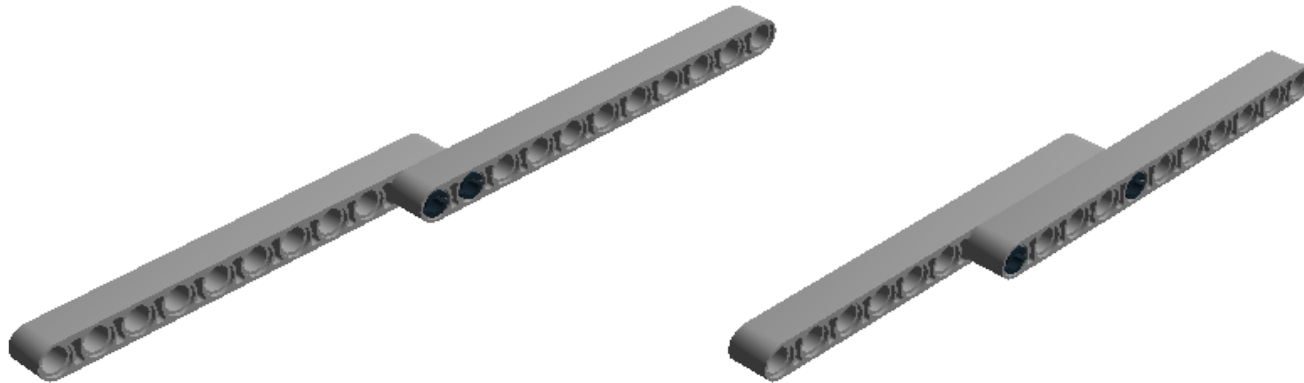


# Connecting beams

Use pegs to connect beams

At least two pegs are needed to make a rigid structure

Greater distance between pegs reduces flex



More pegs increases hold between beams

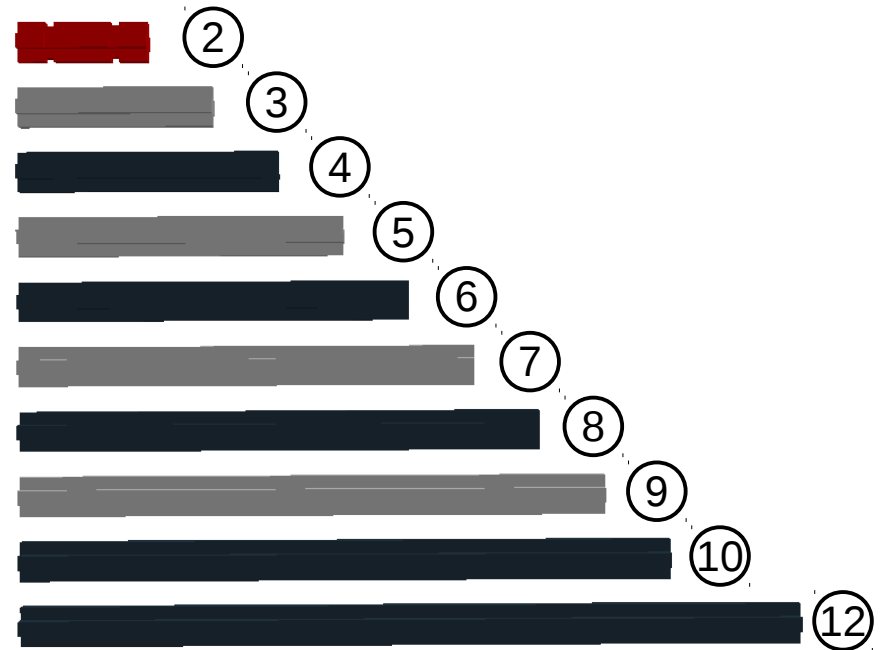
# Axles

Used for wheels, gears,  
and attachments

Length also measured  
in “M” units

Grey axles are typically  
odd lengths, black axles  
are typically even lengths

Axles will rotate and slide in beam holes  
unless constrained



# Wheels

Many types of wheels and tires available

Wheel consists of “rim” and “tire”

Tire measurements printed on sidewall

Cross hole attaches to axles

- ♦ 56908 Rim wide 43.2 x 26
- ♦ 41897 Tyre Low Wide 56 x 28
- ♦ 32020c01 Wheel 62.4 x 20, with Black Tire 62.4 x 20

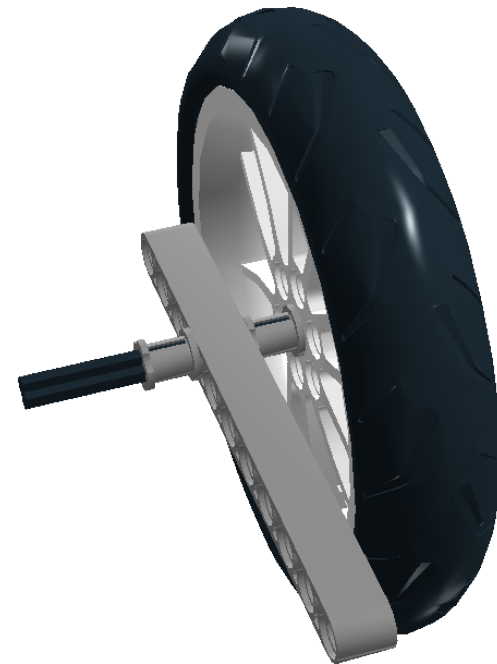


# Bushings

Used to hold axles on beams

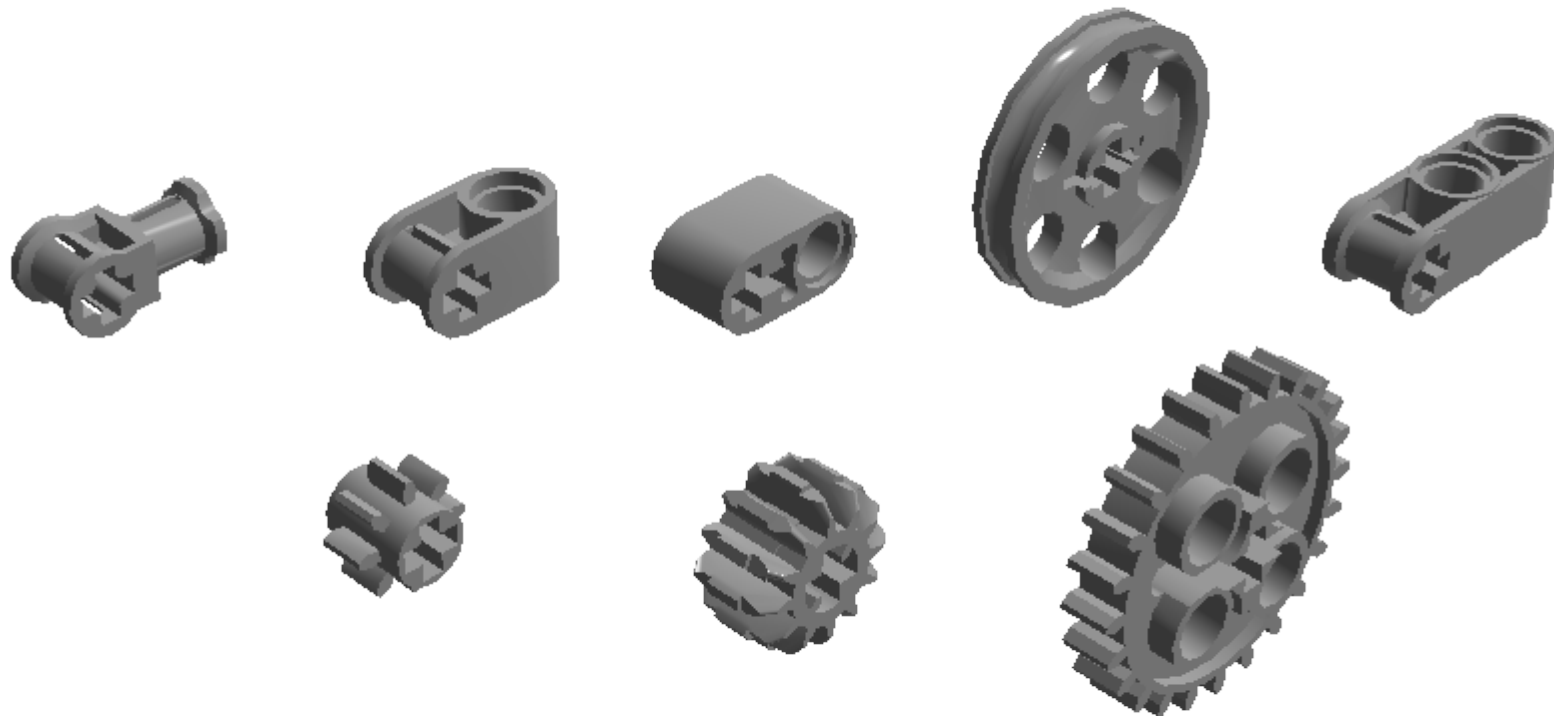
Also used as spacers to prevent tires from hitting beams or other elements

- ♦ 32123 Half-bushing
- ♦ 6590 Bushing



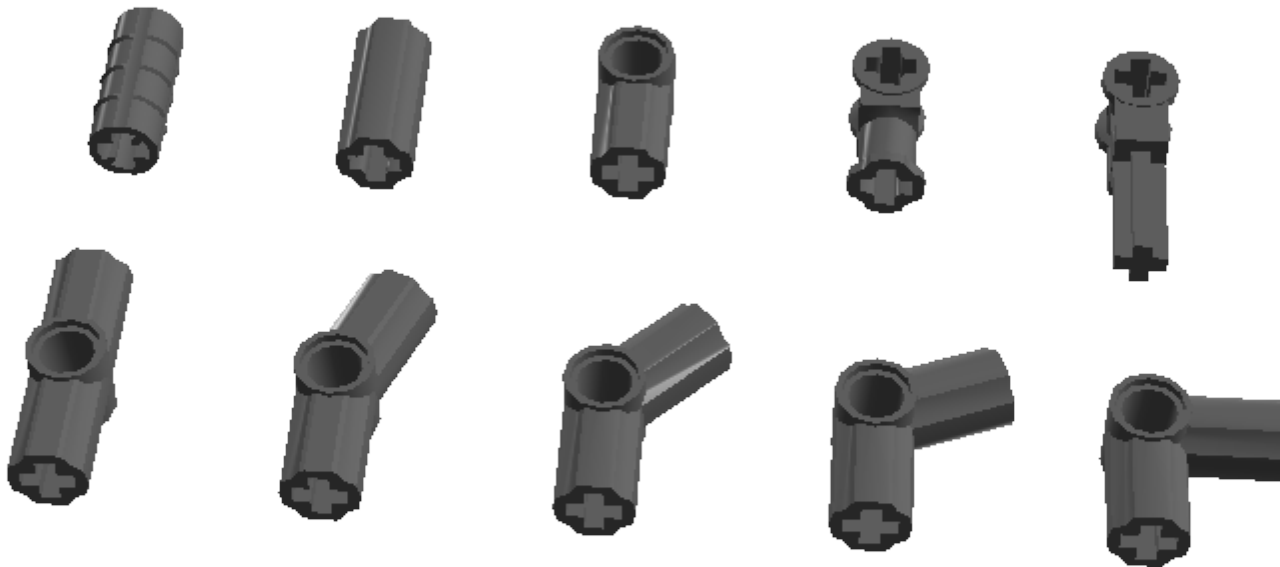
# Bushings

Other elements can also be used as bushings or spacers



# Axle connectors

Axles can be joined using a wide variety of connectors



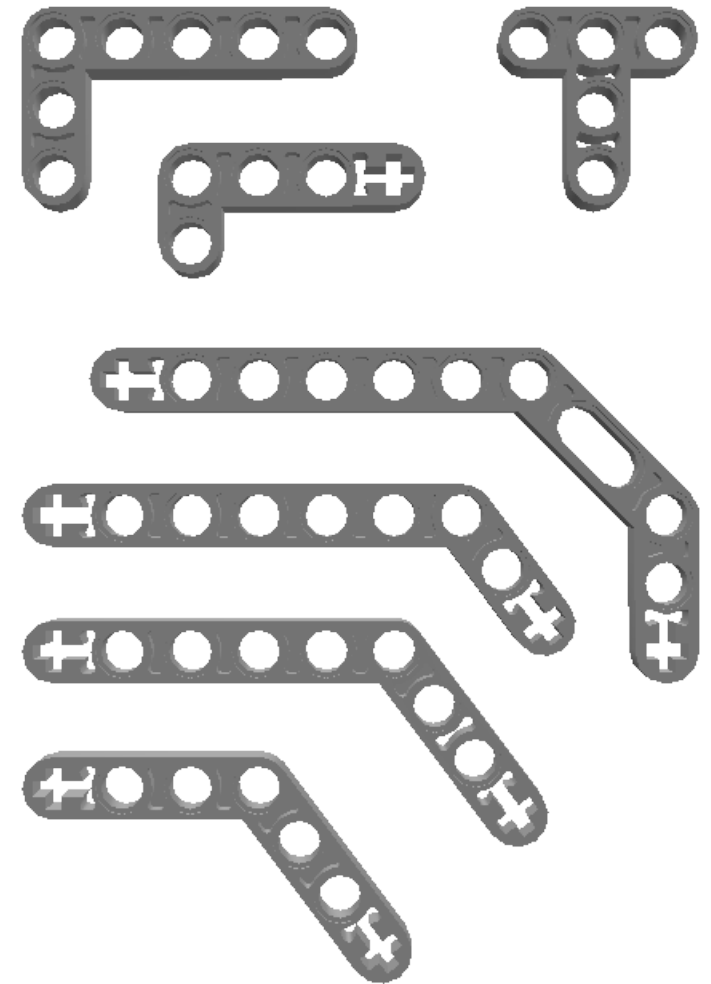
# Angle beams

Allow beams to be connected at rigid angles

Excellent for structure

Some beams have cross holes

- ♦ 32526: 3x5 L beam
- ♦ 32140: 2x4 L beam
- ♦ 60484: 3x3 T beam
- ♦ 32009: 3x7 double-angle beam
- ♦ 32271: 3x7 angle beam
- ♦ 6629: 4x6 angle beam
- ♦ 32348: 4x4 angle beam

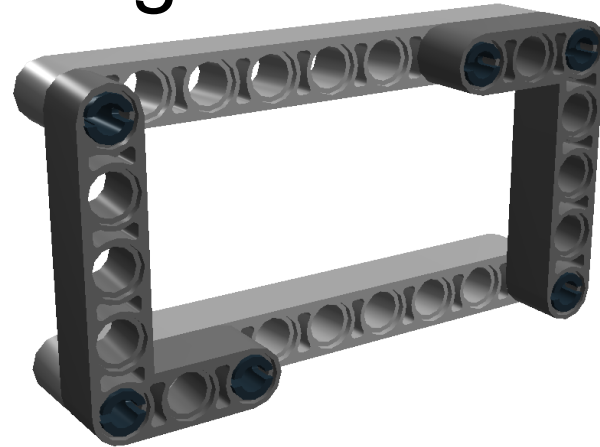


# Structural strength

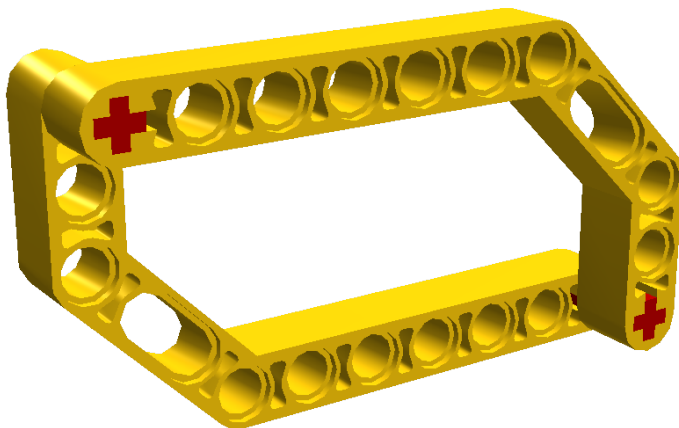
Weak



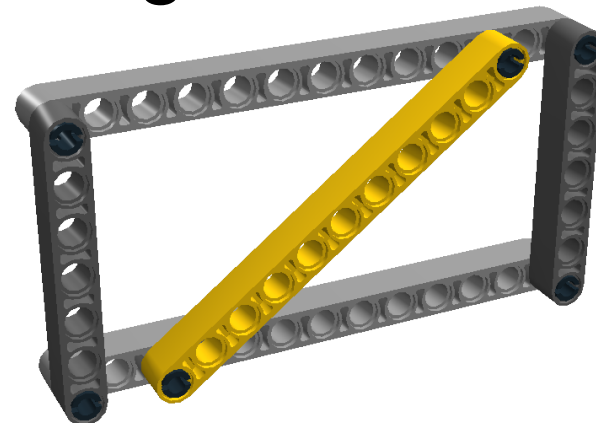
Strong



Strong



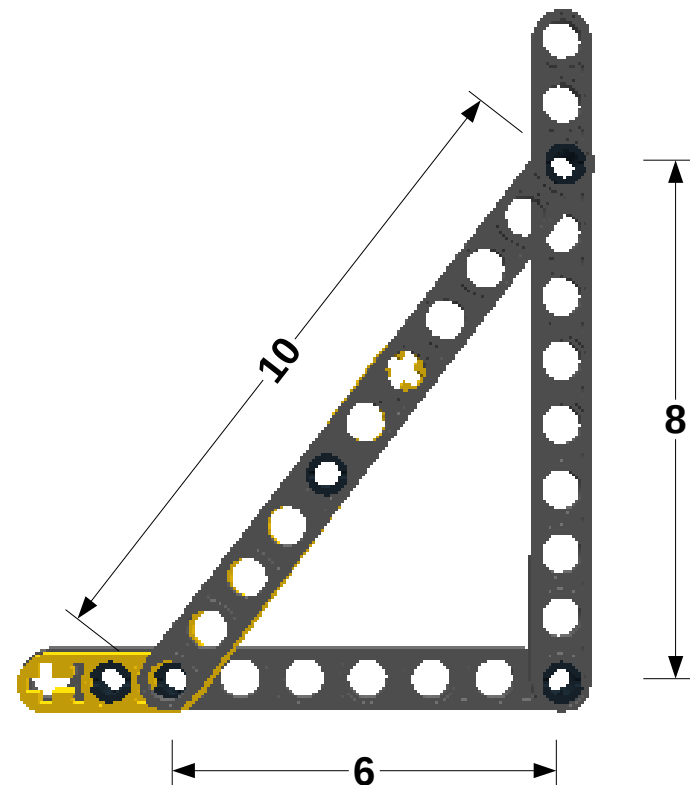
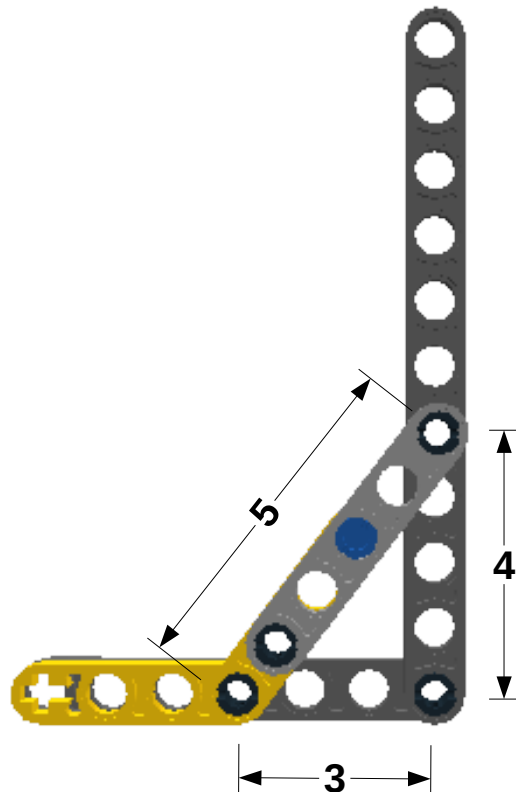
Strong



## 3:4:5 triangles

Angled bracing is very strong

Use 3:4:5 spacing to ensure right angles and proper alignment

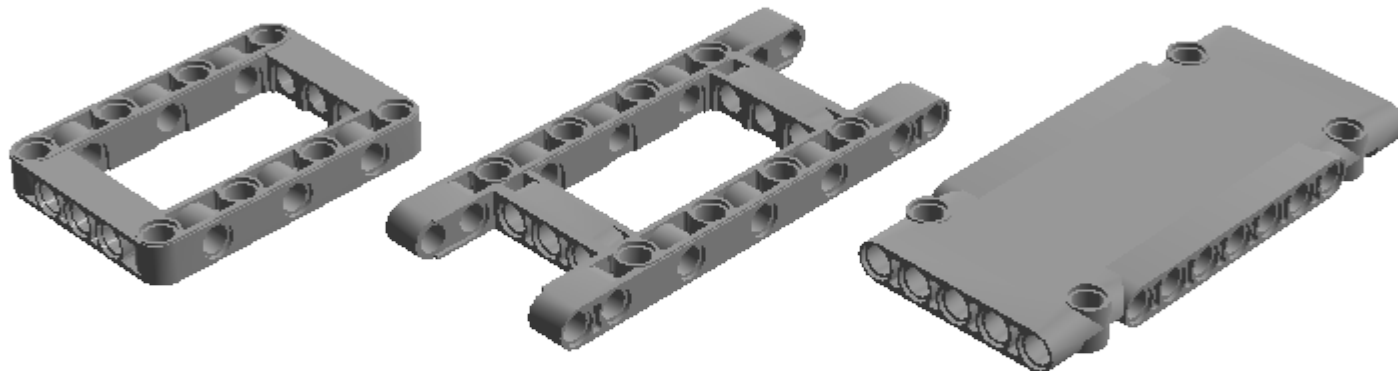


# Useful LEGO pieces - frames and panels

These pieces are excellent for building large structures and boxes

Holes in all three axes for multiple mounting options

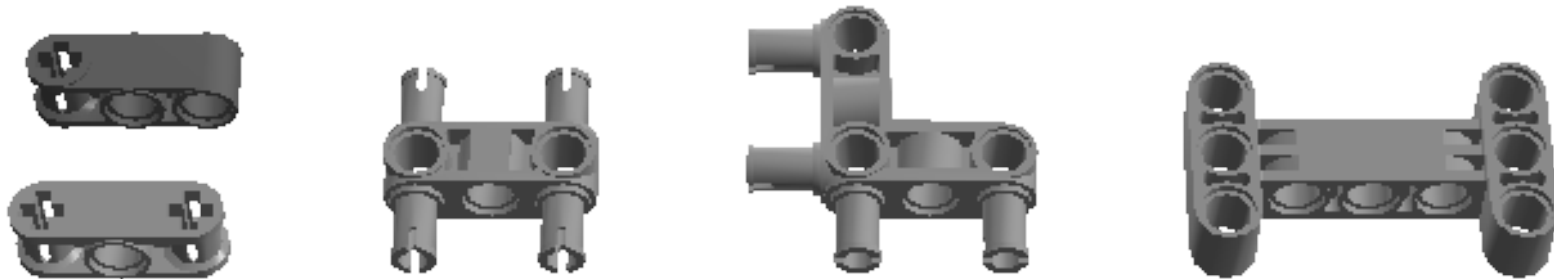
- ♦ 64179: Beam frame 5x7 (“box frame”)
- ♦ 64170: Beam H frame 5x11 (“H frame”)
- ♦ 64782: Flat Panel



# Useful LEGO pieces - cross blocks and beams

These allow connections in multiple directions

- ♦ 42003: Cross block 3M
- ♦ 32184: Double cross block
- ♦ 48989: Beam 3M with 4 snaps
- ♦ 55615: Angular beam 90 degrees with 4 snaps
- ♦ 14720: Beam I-Frame 3x5 90 degrees



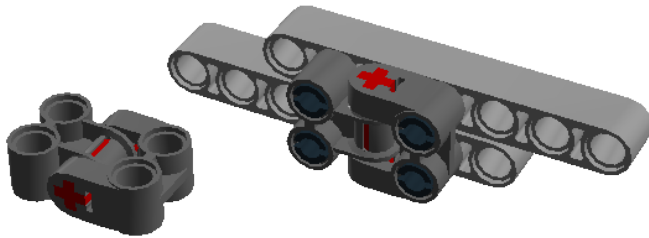
# Useful LEGO pieces - cross blocks

These cross blocks have a wide variety of uses

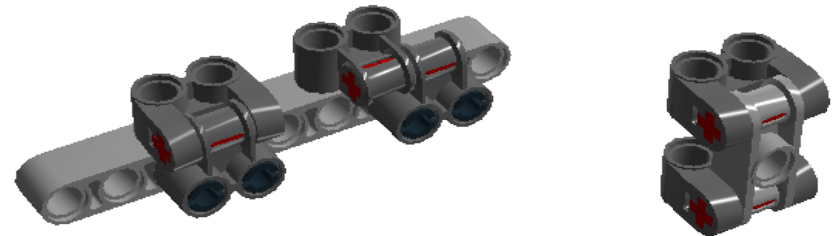
- ♦ 32291: Cross block 2x1 (“Mickey”)
- ♦ 41678: Cross block fork 2x2 (“Minnie”)



Connect two parallel beams



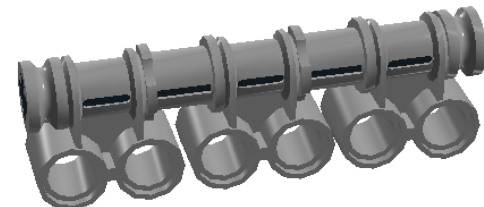
Create holes at right angles



Mount axles in different directions

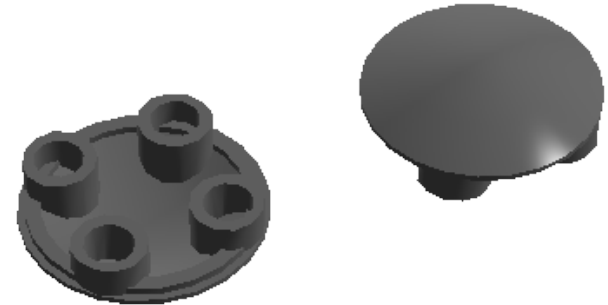


Create “beams” with even # of holes

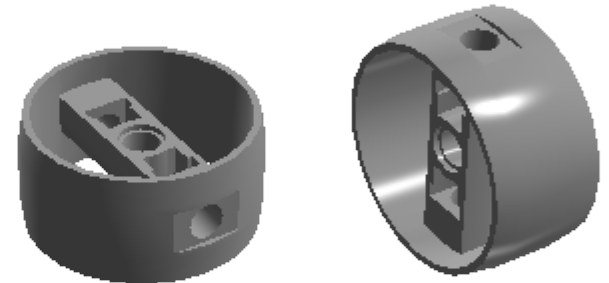


# Useful LEGO pieces - misc

- ♦ 2654: Slide shoe round 2x2  
(good for skids)



- ♦ 41531: Turbine 31.01 x 2  
(wheels that also slide)



# Places to find and buy LEGOs

<http://bricklink.com/>

Website to connect buyers and sellers of LEGOs



Attachments

# Attachments

*Attachments* are the things added to the chassis to solve missions and manipulate models

Good attachment design makes solving missions easy

# Attachment design principles

Robot precision often limited to 1.5cm

If a target is small, try to make the attachment big

Use mission models and walls for precise alignment

Things that seem easy for humans can be hard for a robot

- Manually test attachments with eyes closed

# Mounting attachments

The best attachments are those that never need to be added or removed from the robot

→ saves time during matches

If an attachment must be added or removed, make sure it can be done quickly

- Avoid using pegs for removable attachments

- Use axles and axle pegs

- Use gravity

Removing is usually faster than adding

Rubber bands can be used to snap attachments into place

# Attachment types

FLL missions usually involve

Pushing

Pulling

Lifting

Dropping / dumping

Placing / delivering

Releasing

Capturing / collecting

Shooting

Turning

# Attachment building tips

Tend to use axles and plates when possible

Axles are easy to adjust, resize, and relocate

Plates and frames are better than walls of beams

Sources of energy for attachments

(in order of preference)

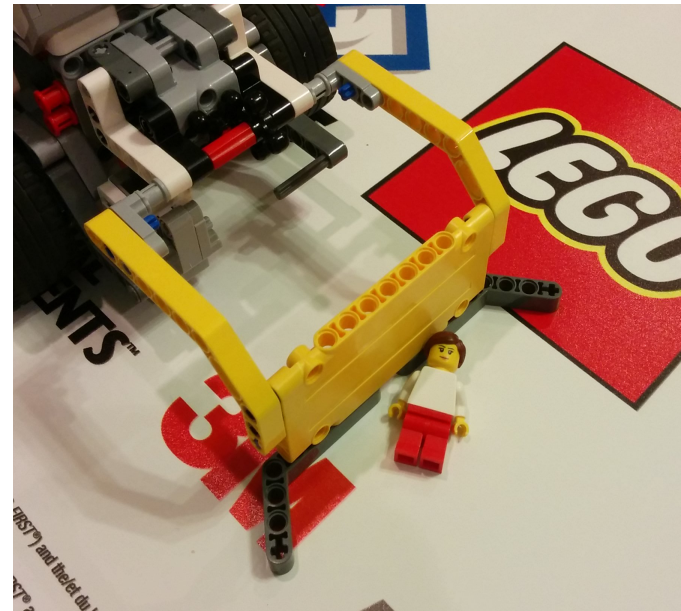
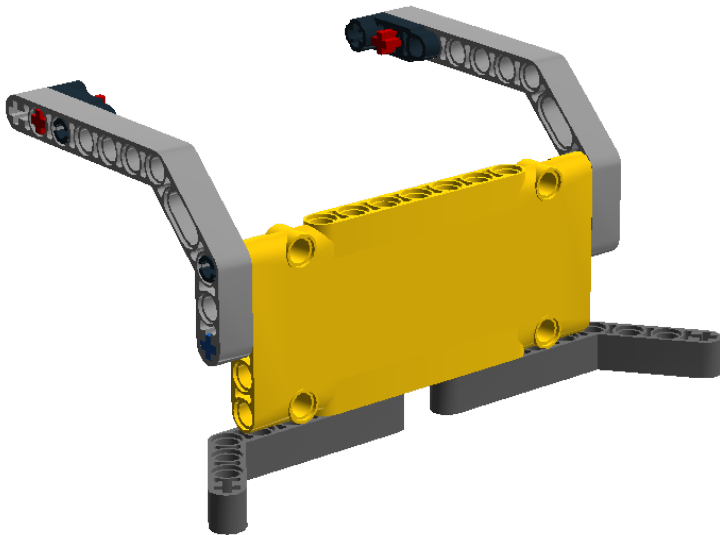
1. Gravity
2. Leverage
3. Elastics
4. Motors

# Passive attachment: Pusher

One of the simplest (and useful) attachments is a bumper.

A bumper can easily push/deliver objects

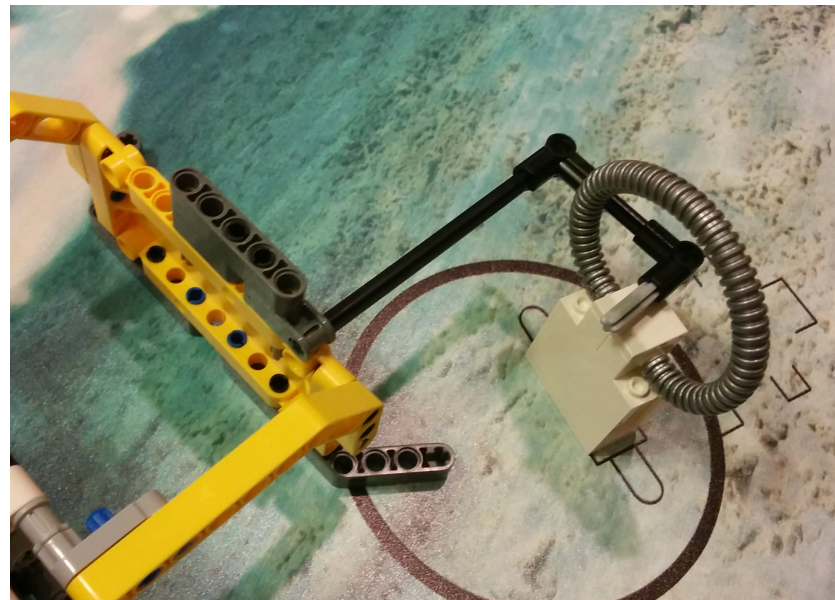
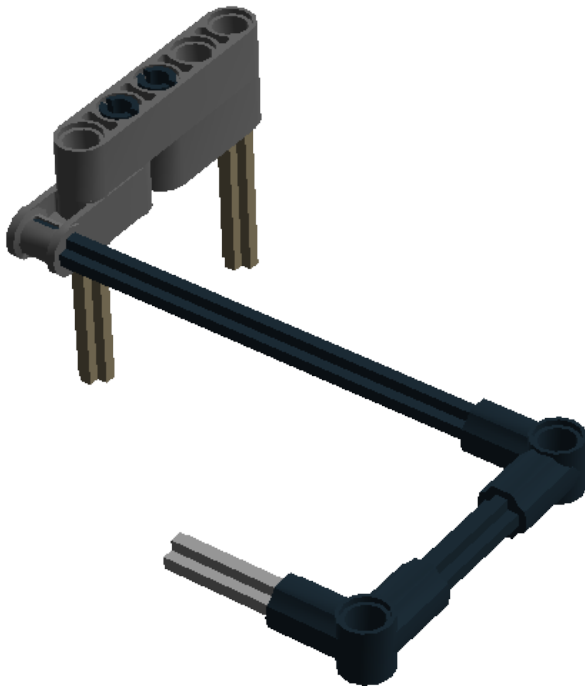
It can also provide places for other attachments



# Passive attachment: Hook

A hook can be used to capture objects

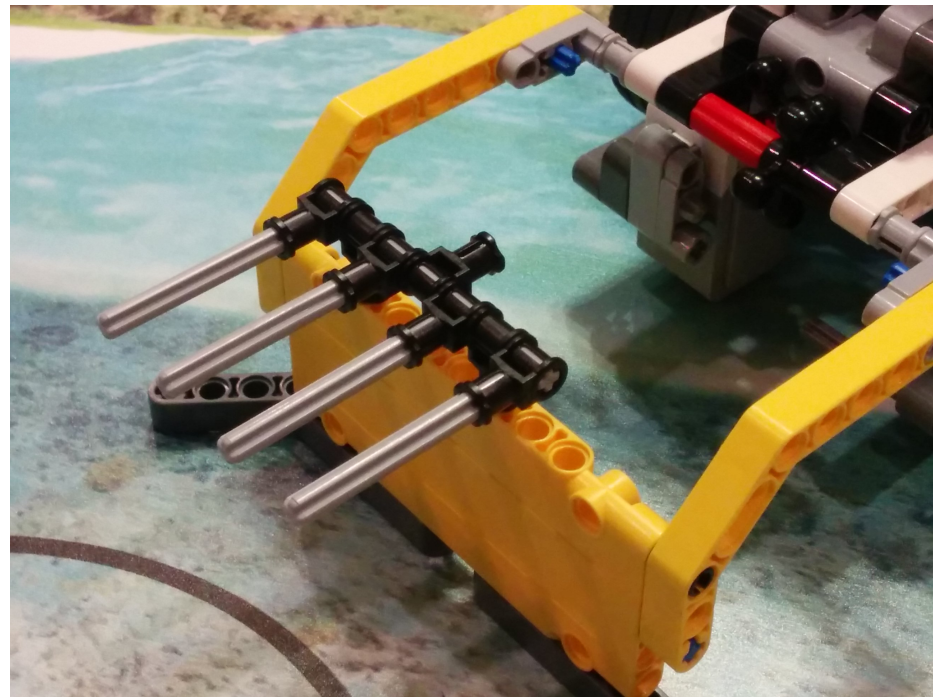
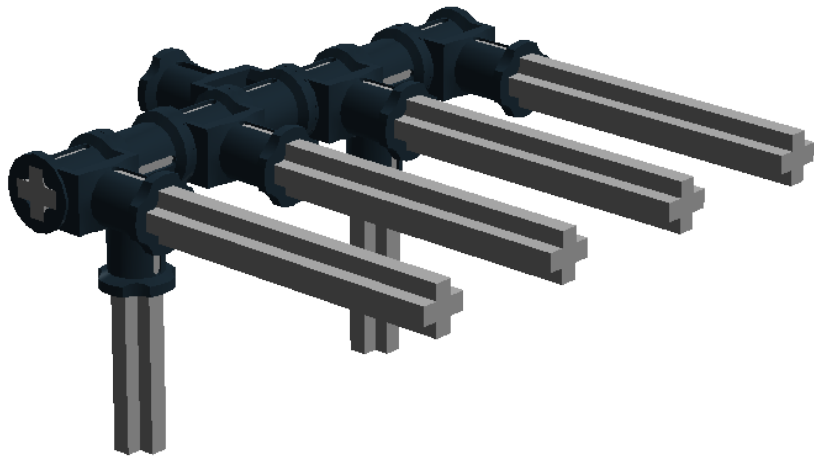
Axles allow quick attach / removal



# Passive attachment: Fork

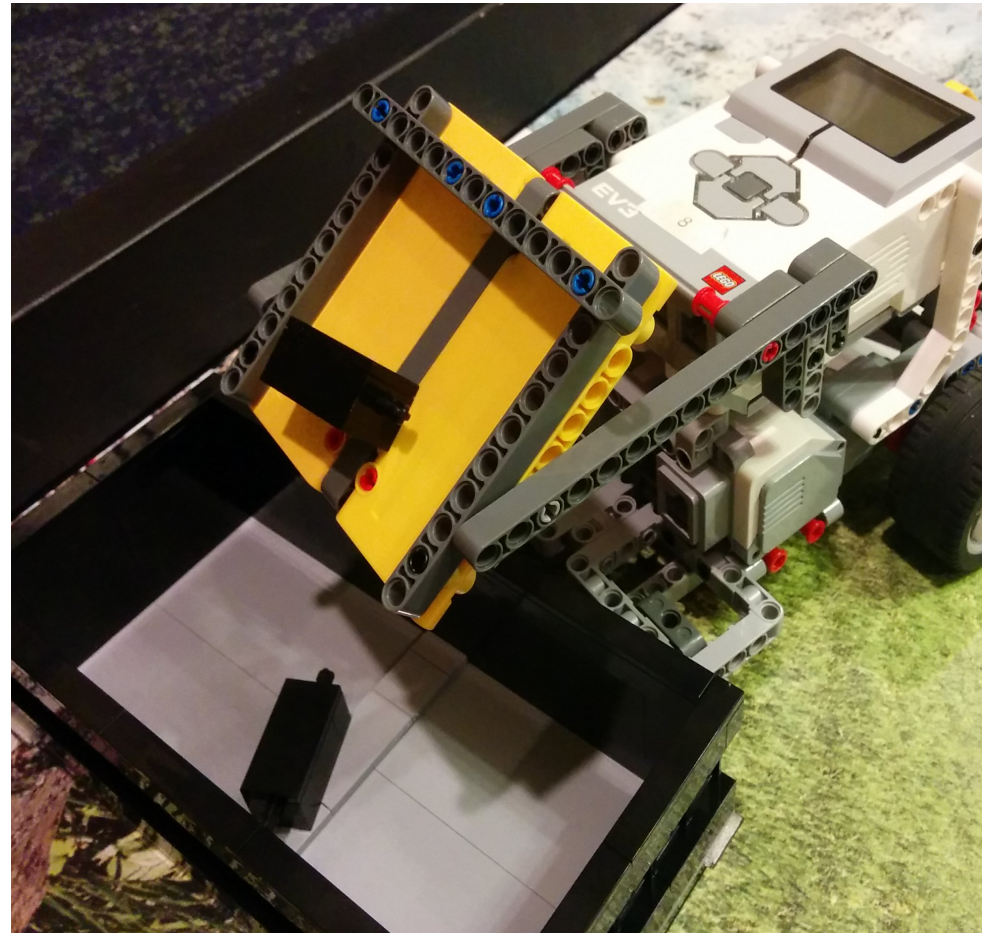
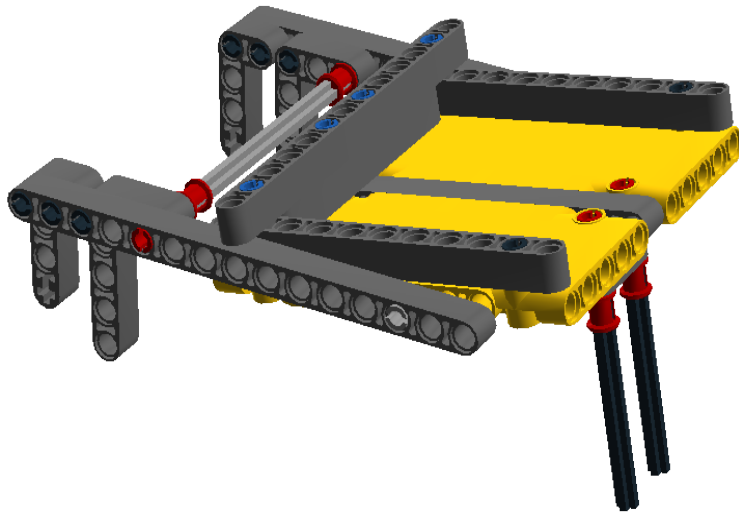
A fork has multiple prongs for capturing objects

This helps make a wider target



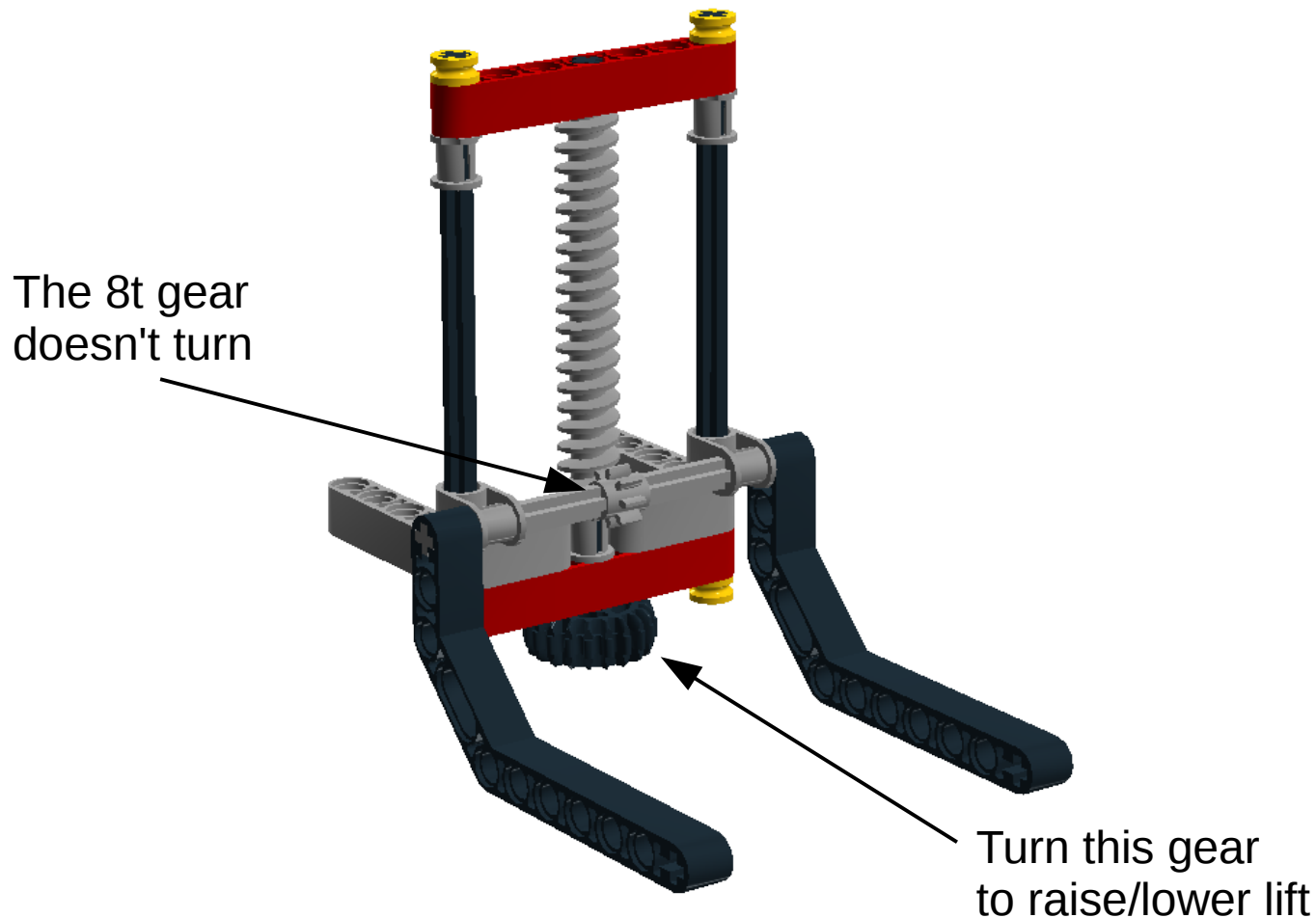
# Passive attachment: Dumper

Dumpers use gravity and simple pegs to release contents



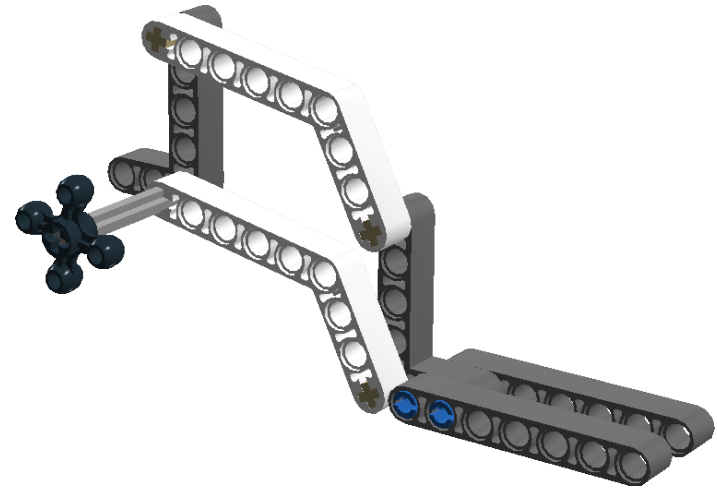
# Powered attachment: Vertical lifts

Forklift will raise or lower as worm gear turns

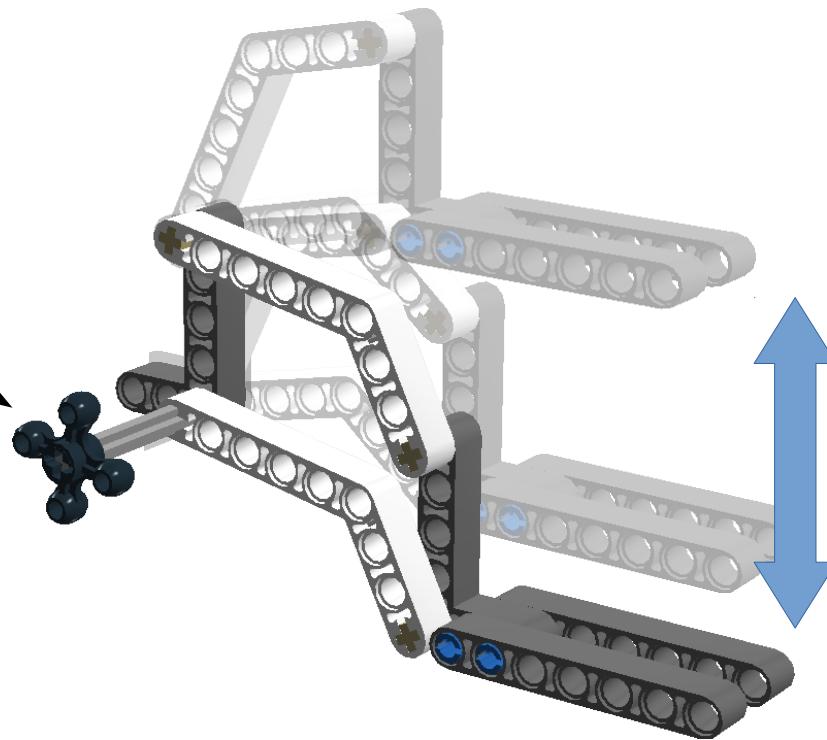


# Powered attachment: Four-bar linkage

Raise/lower bar  
without rotating it



Turn this gear  
to raise/lower lift



Fork moves up  
and down without  
rotating