## FIRST LEGO League Judging Helper—ROBOT DESIGN Room #\_\_\_\_\_ Judges:\_\_\_\_\_

Instructions: Fill in the table for each team you see (details for top 2-4 teams). This will be helpful when rooms merge lists at the end of the day. You'll be expected to know about each of the areas on the sheet for top teams. If a team doesn't discuss something in the presentation, ask them about it.

Robot Design rank  Sturdiness – can withstand rigor  Enhancement – Improvement or new version of existing design Efficiency – Effective vs. excessive parts  Sensors – Which ones, how many, and for what?  Programing – myBlocks, loops, switches, etc.  Programming Comments – Present and relevant  Strategy – Process, etc.  Innovation – Unique features  Strength(s) — [] Mech. Design [] Programming	Team #:							
Tank  Sturdiness – can withstand rigor  Enhancement – Improvement or new version of existing design  Efficiency – Effective vs. excessive parts  Sensors – Which ones, how many, and for what?  Programming – myBlocks, loops, switches, etc.  Programming Comments – Present and relevant  Strategy – Process, etc.  Innovation – Unique features  Strength(s) — Unique features  [] Mech. Design   [] Programming   [] Strategy & [] Strat								
Stundiness - can withstand rigor  Enhancement - Improvement or new version of existing design  Efficiency - Effective vs. excessive parts  Sensors - Which ones, how many, and for what?  Programing - my8locks, loops, switches, etc.  Programming Comments - Present and relevant  Strategy - Process, etc.  Innovation - Unique features  Strength(s) -	_							
withstand rigor Enhancement – Improvement or new version of existing design Improvement or new version of existing design   Efficiency – Efficiency – Efficiency – Effective vs. excessive parts Improvement or new version of existing design   Sensors – Which ones, how many, and for what? Improvement or new version of existing design   Programing – myBlocks, loops, switches, etc. Improvement or new version of existing design   Programming Comments – Present and relevant Improvement or new version of existing design   Strategy – Process, etc. Improvement or new version of existing design   Innovation – Unique features Improvement or new version of existing design   Strength(s) – Unique features Improgramming of items of item								
Enhancement - Improvement or new version of existing design  Efficiency - Effective vs. excessive parts  Sensors - Which ones, how many, and for what?  Programing - MyBlocks, loops, switches, etc.  Programming - Programming - Programming   Improgramming   Improgramming								
Improvement or new version of existing design  Efficiency — Effective vs. excessive parts  Sensors — Which ones, how many, and for what?  Programming — myBlocks, loops, switches, etc.  Present and relevant — Present and relevant — Strategy — Process, etc.  Innovation — Unique features  Strength(s) — Unique features  Eight Mech. Design [] Mech. Design [] Programming [] Programming [] Programming [] Programming [] Strategy & [] Strate								
new version of existing design  Efficiency — Effective vs. excessive parts  Sensors — Which ones, how many, and for what?  Programing — myBlocks, loops, switches, etc.  Programming Comments — Present and relevant  Strategy — Process, etc.  Innovation — Unique features  Strength(s) — Unique features  Strength(s) — Unique features  Elimech. Design, [] Programming [] Programming [] Programming [] Programming [] Programming [] Strategy & [] Strategy								
existing design  Efficiency –  Effective vs. excessive parts  Sensors – Which ones, how many, and for what?  Programing – myBlocks, loops, switches, etc.  Programing Comments –  Present and relevant  Strategy –  Process, etc.  Innovation –  Unique features  Strength(s) –  Unique features  [] Mech. Design for programming for programm	•							
Efficiency – Effective vs. excessive parts Sensors – Which ones, how many, and for what?  Programing – myBlocks, loops, switches, etc.  Programing Comments – Present and relevant Strategy – Process, etc.  Innovation – Unique features  Strength(s) – Mech. Design Mech. Design [] Programming [] Programming [] Programming [] Programming [] Programming [] Programming [] Strategy & [								
Effective vs. excessive parts  Sensors – Which ones, how many, and for what?  Programing – myBlocks, loops, switches, etc.  Programming Comments – Present and relevant  Strategy – Process, etc.  Innovation – Unique features  Strength(s) — Unique features  Strength(s) — [] Mech. Design [] Programming [] Pr								
excessive parts  Sensors – Which ones, how many, and for what?  Programing – myBlocks, loops, switches, etc.  Programing Comments – Present and relevant  Strategy – Process, etc.  Innovation – Unique features  Strength(s) — Mech. Design [] Programming [] Progra	_							
Sensors – Which ones, how many, and for what?  Programing – myBlocks, loops, switches, etc.  Programming Comments – Present and relevant  Strategy – Process, etc.  Innovation – Unique features  Strength(s) – Mech. Design [] Mech. Design [] Programming [] Progra								
ones, how many, and for what?  Programing — myBlocks, loops, switches, etc.  Programming Comments — Present and relevant  Strategy — Process, etc.  Innovation — Unique features  Strength(s) — Mech. Design [] Mech. Design [] Mech. Design [] Programming [] Strategy & [] Strate								
and for what?  Programing – myBlocks, loops, switches, etc.  Programming Comments – Present and relevant  Strategy – Process, etc.  Innovation – Unique features  Strength(s) — Unique features  [] Mech. Design [] Programming [] Programming [] Programming [] Programming [] Programming [] Strategy & [] Strategy								
Programing – myBlocks, loops, switches, etc.       Programming Comments – Present and relevant       Image: Comments – Process, etc.       Image: Comments								
myBlocks, loops, switches, etc.  Programming Comments – Present and relevant  Strategy – Process, etc.  Innovation – Unique features  Strength(s)— [] Mech. Design [] Programming [] Strategy &								
switches, etc.  Programming Comments – Present and relevant Strategy – Process, etc.  Innovation – Unique features  Strength(s) — Mech. Design, Programming, Programming, Programming, Programming, Programming, Programming, Programming,  Strategy &  Strategy &								
Programming Comments – Present and relevant Strategy – Process, etc.  Innovation – Unique features  Strength(s) — Mech. Design, Programming, Programming Programmi								
Comments – Present and relevant  Strategy – Process, etc.  Innovation – Unique features  Strength(s)— Mech. Design, Programming, Progra	·							
Present and relevant  Strategy - Process, etc.  Innovation - Unique features  Strength(s)— Mech. Design, Programming, Prog								
Strategy – Process, etc.  Innovation – Unique features  Strength(s) — Mech. Design, Programming, Programming, Programming,  [] Strategy &  []								
Strategy – Process, etc.  Innovation – Unique features  Strength(s) — Mech. Design, Programming, Programming, Programming, Programming, Programming, Programming, Process, etc.    Mech. Design   Mech. D								
Innovation – Unique features  Strength(s)— Mech. Design, Programming,	relevant							
Innovation – Unique features  Strength(s)— Mech. Design, Programming, Programming, Programming, Programming, Programming, Programming, Programming, Programming, Programming, Programming	Strategy –							
Unique features  Strength(s)— Mech. Design, Programming, Programming, Programming, Programming, Programming, Programming, Programming, Programming Programming, Programming Pr	Process, etc.							
Unique features  Strength(s)— Mech. Design, Programming, Programming, Programming, Programming, Programming, Programming, Programming, Programming Programming, Programming Pr	la a costi a a							
Strength(s)— Mech. Design Programming,  [] Mech. Design [] Mech. Design [] Mech. Design [] Programming [] Programming [] Programming [] Strategy & [] Strate								
Mech. Design, [] Programming [] Strategy & [] Strate	Unique reatures							
Mech. Design, [] Programming [] Strategy & [] Strate								
Mech. Design, [] Programming [] Strategy & [] Strate								
Mech. Design, [] Programming [] Strategy & [] Strat	Strength(s)—	[] Mech. Design						
Programming, [] Strategy &	_	•	_		•	•		•
Strat./Innovation   Innovation   Innovation   Innovation   Innovation   Innovation   Innovation   Innovation	Strat./Innovation	Innovation						

 $<sup>^{\</sup>circ}$  2019 Jared Hasen-Klein, Robotics Resource Hub. Used by special permission. Visit us on the web at hub.jaredhk.com.

## Sample questions & helpful hints

Category	Look for	Sample questions			
Sturdiness – can withstand rigor	Does the robot appear to withstand normal operating wear during an event?	When or how often do you need to repair your robot? What types of things cause the robot to break?			
Enhancement – Improvement or new version of existing design	Team explains the basis for their design and how they arrived at the current iteration.	Where did you get inspiration for your design? How did you think of the design?			
Efficiency – Effective vs. excessive parts	All or most parts have a clear purpose and are sufficient but not excessive to accomplish goals set out by team.	What is the purpose of (for mechanisms with unclear purpose)? Are there any components of your robot which you do not use now?			
Sensors – Which ones, how many, and for what?	Sensors are present and have a stated and documented purpose. There is an explained benefit to the team by using sensors.	Which sensors do you use and for what purpose? (If no sensors) Do you run into any problems without using sensors?			
Programing – myBlocks, loops, switches, etc.	Programming is sophisticated but streamlined, has a variety of block types, and is well explained.				
Programming Comments – Present and relevant	Comments are present throughout the programming and are helpful, serving a purpose.				
Strategy – Process, etc.  Team has a well-docume process for their mission building strategies.		How did you decide which missions to do?			
Innovation – Unique features	Team has unique features not seen in most cases.	What is unique or different about your robot? What is something fun about your robot?			

<sup>© 2019</sup> Jared Hasen-Klein, Robotics Resource Hub. Used by special permission. Visit us on the web at hub.jaredhk.com.