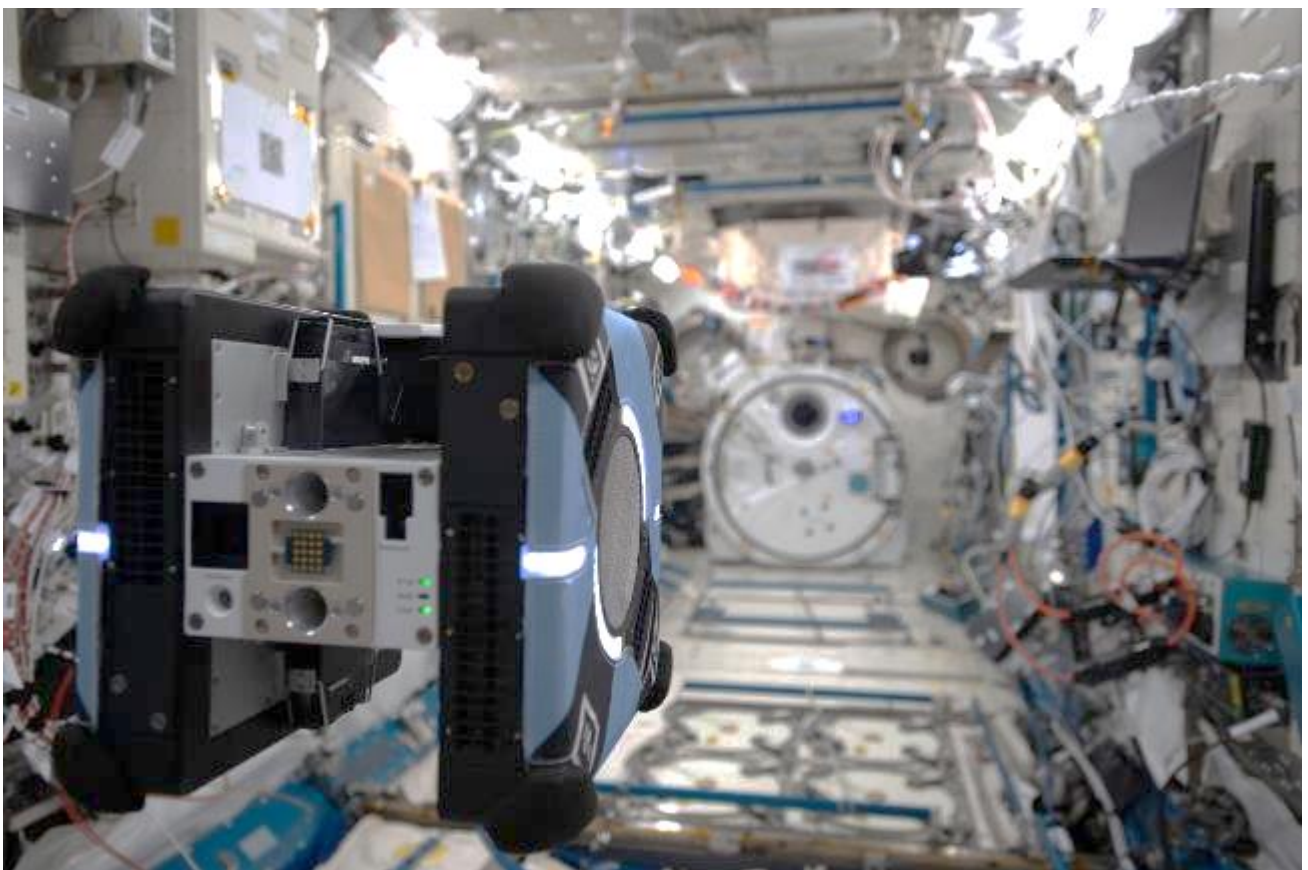


The 5th Kibo Robot Programming Challenge Rulebook



Version 1.0 (Revision date: April 1, 2024)

Japan Aerospace Exploration Agency (JAXA)



Revision History

Revision history is listed below.

Revision Date	Version	Paragraph	Revision Location
April 1, 2024	1.0	All	-



Table of Contents

1. Introduction.....	1
2. Preliminary Round.....	2
2.1. Preliminary Round Period.....	2
2.2. Game Rules.....	3
2.2.1. Game Scenario.....	3
2.2.2. Preconditions.....	5
2.2.3. Objects.....	7
2.2.4. Mission Completion Report.....	8
2.2.5. Keep-In-Zone (KIZ) & Keep-Out-Zone (KOZ)	8
2.2.6. 10 Runs per APK.....	12
2.2.7. Ranking from Average of 10 Runs.....	12
2.3. Scoring	13
2.3.1. Factors	13
2.4. Participation in the Preliminary Round	14
2.4.1. How to Participate in the Preliminary Round	14
2.4.2. Preliminary Round APK Submission	14
3. Final Round.....	16
3.1. Final Round Schedule	16
3.2. Game Rule	17
3.2.1. Game Scenario.....	17
3.2.2. Preconditions.....	18
3.2.3. Objects.....	18
3.2.4. Mission Complete Report.....	19
3.2.5. Keep-In-Zone (KIZ) & Keep-Out-Zone (KOZ)	19
3.2.6. 1 Run per APK.....	19
3.2.7. 5 Minute Time Limit	19
3.2.8. APK Operation on the Day of the Final Round	19
3.2.9. Final Round Run Order.....	21
3.3. Scoring	23
3.3.1. Factors	23
3.3.2. Judging	23
3.4. Participating in the Final Round.....	23



3.4.1. Submit APK and Source Code.....26

3.5. Organizing the Event28

Appendix29

1. Introduction

The 5th Kibo Robot Programming Challenge (Kibo-RPC) is here. Create the best program to see if you can win.

Preliminary round will be held in each country/region to select their representatives. Participants compete using programs they have developed beforehand using JAXA's simulation environment. Game rules and scoring are basically the same in each country/region, although some have adopted their own evaluation criteria, so be sure to check your local Kibo-RPC website for details. Information regarding venue and dates will be made available by the point of contact (POC) in each country. This Rulebook contains general rules for all participants.

The winning teams from each preliminary round get to compete to be the best in the world in the final round where they will program and operate an Astrobee free-flying robot in the Japanese Experiment Module KIBO, which is part of the International Space Station (ISS).

2. Preliminary Round

2.1. Preliminary Round Period

The preliminary round rounds will be held separately in each country/region during the period, so please participate in the preliminary round held in the country/region where you are registered. Preliminary round information for each country/region can be found on the official Kibo-RPC website (<https://jaxa.krpc.jp/>). For more information, please contact your country/region POC.

Note: Participants cannot run web simulations from the Preliminary Round tab. The Preliminary Trial tab is always available.

Question Acceptance Period^{*1}: June 20, 2024, 12:00 (JST)

APK Submission Period: May 27-June 20, 2024, 23:59 (JST)

Preliminary round Period: June 21 to July 7, 2024

^{*1} Any questions submitted after this date will receive a delayed response.

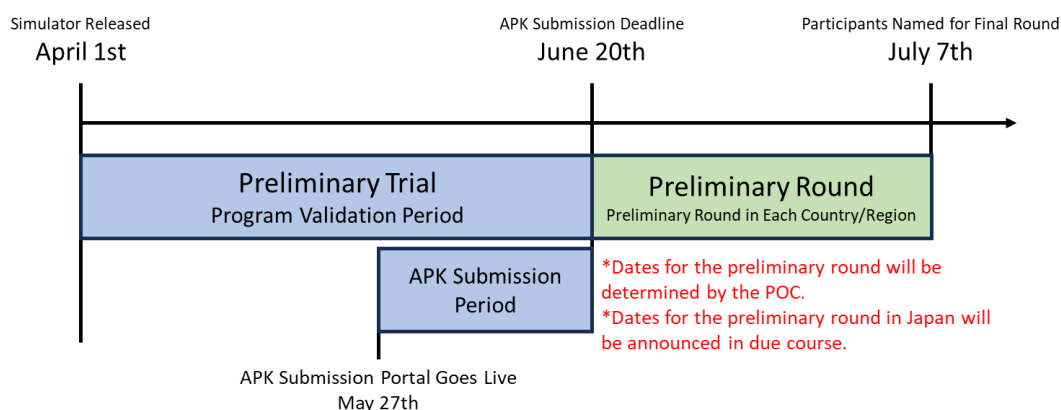


Figure 2.1-1 Preliminary Round Period

2.2. Game Rules

2.2.1. Game Scenario

To control NASA's Astrobees in the Preliminary round, please create a program to complete the following game using JAXA's web simulation environment.

Within the time limit, move Astrobees from the starting position (docking station) to the position where the images of objects in Kibo are placed, and read all the images. After that, Astrobees will move to the astronauts, read the image of the object that the astronaut is looking for, and ask the astronaut to tell us what he/she is looking for. After reading the images, the Astrobees move in front of the object that the astronaut is looking for and take a picture of it. The mission ends when the Astrobees shine their Signal Lights, which signal that they have found the object, and tell the astronaut where the object is.

1. Astrobees start from the Docking Station
2. After the starting, Astrobees will patrol the several candidate sites for the location of objects prepared in the Kibo module while avoiding KOZ^{*1}.
3. After it finishes its patrol, it moves to the astronaut and reports what objects (images) it found and where.
4. It asks the astronaut to tell it what he/she is looking for by reading the image in his/her hands.
5. Astrobees will move close to the object the astronaut is looking for and it will take a picture of it.
6. After the photo is taken, the Signal Lights will light up to inform the astronaut of the location of the object and the game will end.

^{*1}Keep Out Zone (KOZ): Areas where Astrobees cannot enter; attempts to enter the Keep Out Zone will be denied.

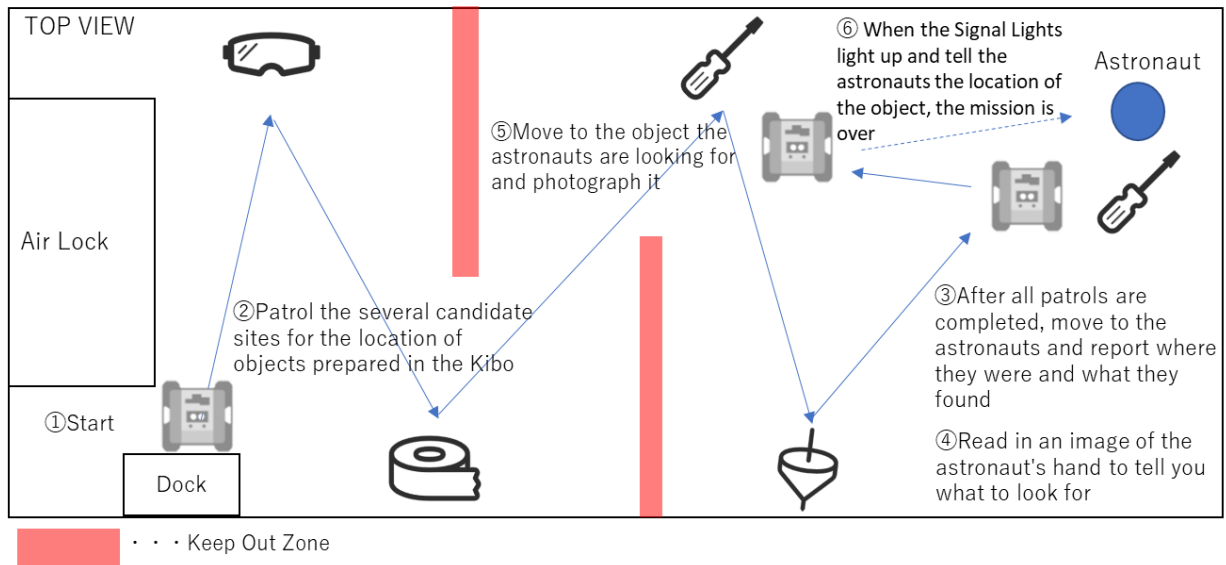


Figure 2.2.1-1: Overview of Preliminary Round

2.2.2. Preconditions

Table 2.2.2-1 Preconditions for preliminary round

#	Content
1	The starting position is the Docking Station, and the timer starts once Astrobee undocks.
2	<p>There are 10 types of Lost Items.</p> <p>Prepare AR tag on the same plane as the printed surface of Lost Item.</p> <p>The search area for Lost Item (hereinafter referred to as "Area") is limited to four locations.</p> <p>Area is specified as a plane, and one Lost Item is placed somewhere within the Area.</p> <p>Lost Item placement is random.</p>
3	Target Item is determined at random from among the Lost Items used in the game.
4	<p>The following information will be presented to participants in advance. Please refer to section 2.2.3 for more information regarding AR tags, Lost Items.</p> <ol style="list-style-type: none"> Orientation of the position of the starting point (StartPoint) Astronaut's (ExploringCompletionPoint) position and orientation Location and size of each Area Total number of Areas Candidates for Lost Item that astronauts are looking for Location and size of Keep Out Zone Parameters for given angle and distance
5	<p>There are KOZs somewhere in the path of the Astrobee. Coordinates will be provided to you beforehand. Please see Section 2.2.5 for details.</p> <p>* Astrobee cannot pass through KOZs.</p>

Table 2.2.2-2 Coordinate information (starting point and astronaut's position and orientation)

Point	Coordinates			Orientation			
	x	y	z	x	y	z	w
Start	9.815	-9.806	4.293	1	0	0	0
Astronaut	11.143	-6.7607	4.9654	0	0	0.707	0.707

Table 2.2.2-3 Coordinate Information (Area)

		x_min	y_min	z_min	x_max	y_max	z_max
Area*	1	10.42	-10.58	4.82	11.48	-10.58	5.57
	2	10.3	-9.25	3.76203	11.55	-8.5	3.76203
	3	10.3	-8.4	3.76093	11.55	-7.45	3.76093
	4	9.866984	-7.34	4.32	9.866984	-6.365	5.57

* Area is displayed as a plane.

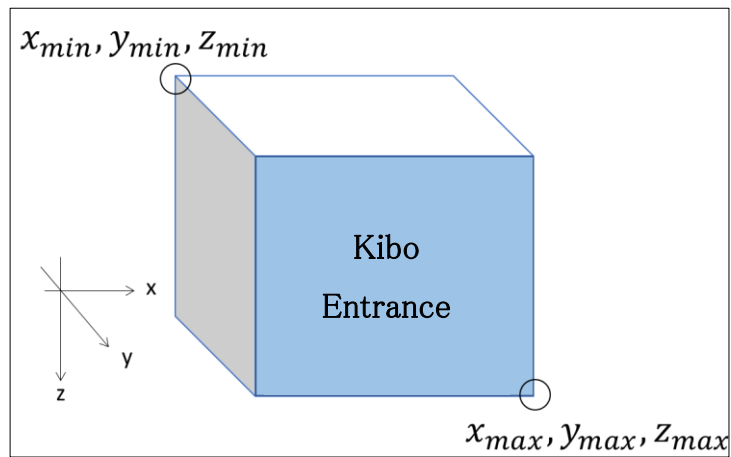
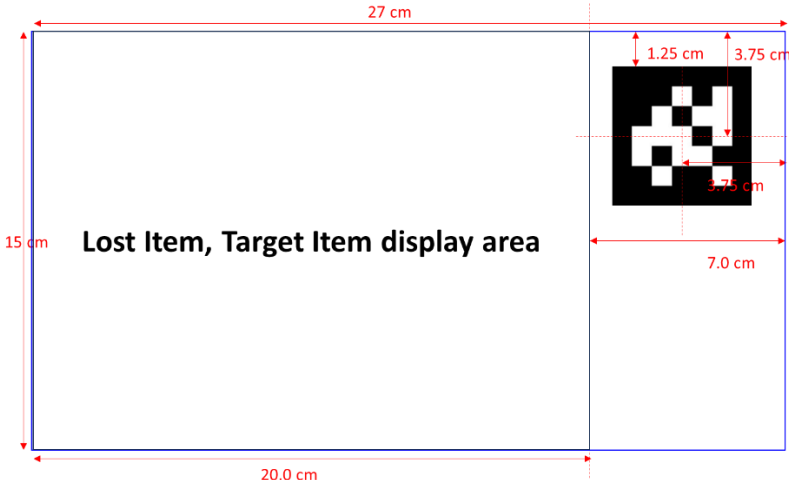
































Figure 2.2.2-1 Kibo coordinate definition

2.2.3. Objects

#	Object Name	Method																								
1	List of Lost Items	<div>AR tag information (The size remains the same for both the Lost Item and the Target Item)</div> <div></div> <div>List of Lost Item</div> <table><tr><td>kapton_tape</td><td>top</td><td>screwdriver</td></tr><tr><td></td><td></td><td></td></tr><tr><td>beaker</td><td>hammer</td><td>pipette</td></tr><tr><td></td><td></td><td></td></tr><tr><td>wrench</td><td>thermometer</td><td>watch</td></tr><tr><td></td><td></td><td></td></tr><tr><td colspan="3">goggle</td></tr><tr><td colspan="3"></td></tr></table> <div>* Lost Item size varies depending on image difficulty.</div>	kapton_tape	top	screwdriver				beaker	hammer	pipette				wrench	thermometer	watch				goggle					
kapton_tape	top	screwdriver																								
																										
beaker	hammer	pipette																								
																										
wrench	thermometer	watch																								
																										
goggle																										
																										
































2	Lost Item Display Examples	Level	Image Examples
		1	<div><div></div><div> Item 4</div><div></div><div> Item 2</div><div></div><div> Item 3</div></div> <div>* Target Item displays a level 1 image</div>
		2	<div><div></div><div> Item 3</div><div></div><div></div><div> Item 2</div><div></div><div> Item 3</div></div>
		3	<div><div></div><div> Item 1</div><div></div><div></div><div></div><div></div><div></div><div> Item 2</div><div> Item 3</div></div>
		4	<div><div></div><div> Item 1</div><div></div><div></div><div></div><div> Item 3</div><div> Item 3</div></div>
		*Levels are subject to change based on all participants.	

Table 2.2.3-1 Objects for Preliminary Round

2.2.4. Mission Completion Report

To complete the mission, you need to create a report using the QR code you scan.
Please see reportMissionCompletion() in chapter 7 of the Programming Manual for the API to use in the Mission Completion Report.

2.2.5. Keep-In-Zone (KIZ) & Keep-Out-Zone (KOZ)

Astrobees may move within Keep-In-Zones (KIZ), which means basically within the walls of Kibo. These are the Astrobees's pre-set boundaries, and if the destination set is outside a KIZ, the command will be rejected. In other words, it is necessary to program the Astrobees to move only within the KIZs.

Keep-Out-Zone (KOZ) is set up within KIZ and is used by Kibo-RPC as a no-entry zone within Kibo's Japanese Experiment Module. The Astrobee's course must be programmed to avoid these KOZs (Figures 2.2.5-1, 2.2.5-2, 2.2.5-3, Table 2.2.5-1).

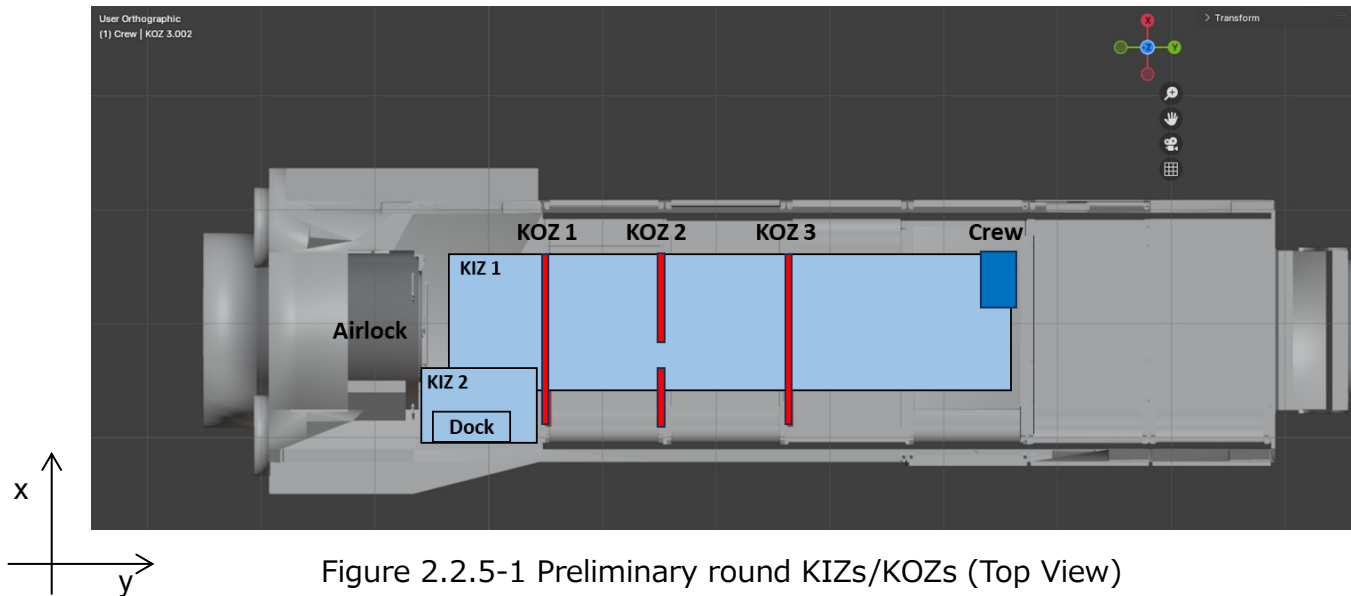


Figure 2.2.5-1 Preliminary round KIZs/KOZs (Top View)

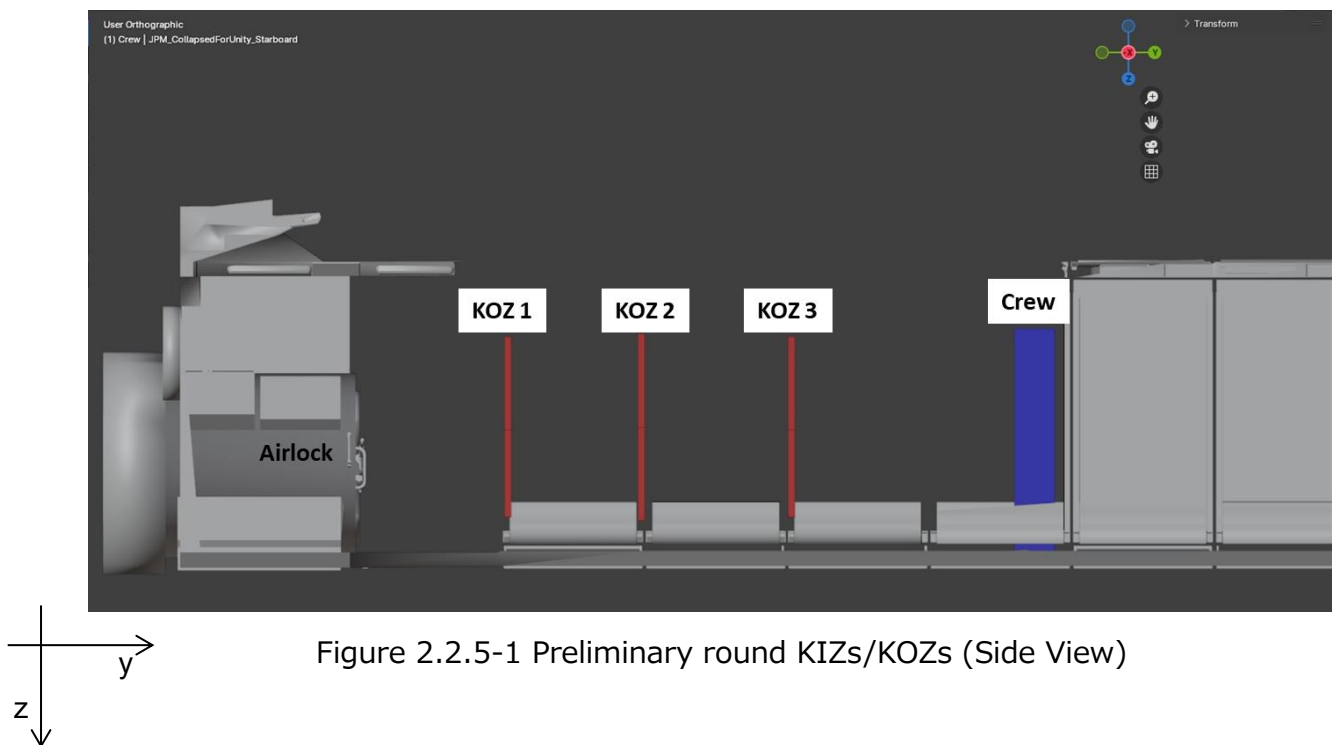
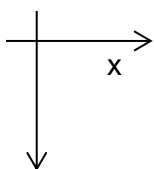


Figure 2.2.5-1 Preliminary round KIZs/KOZs (Side View)



Z

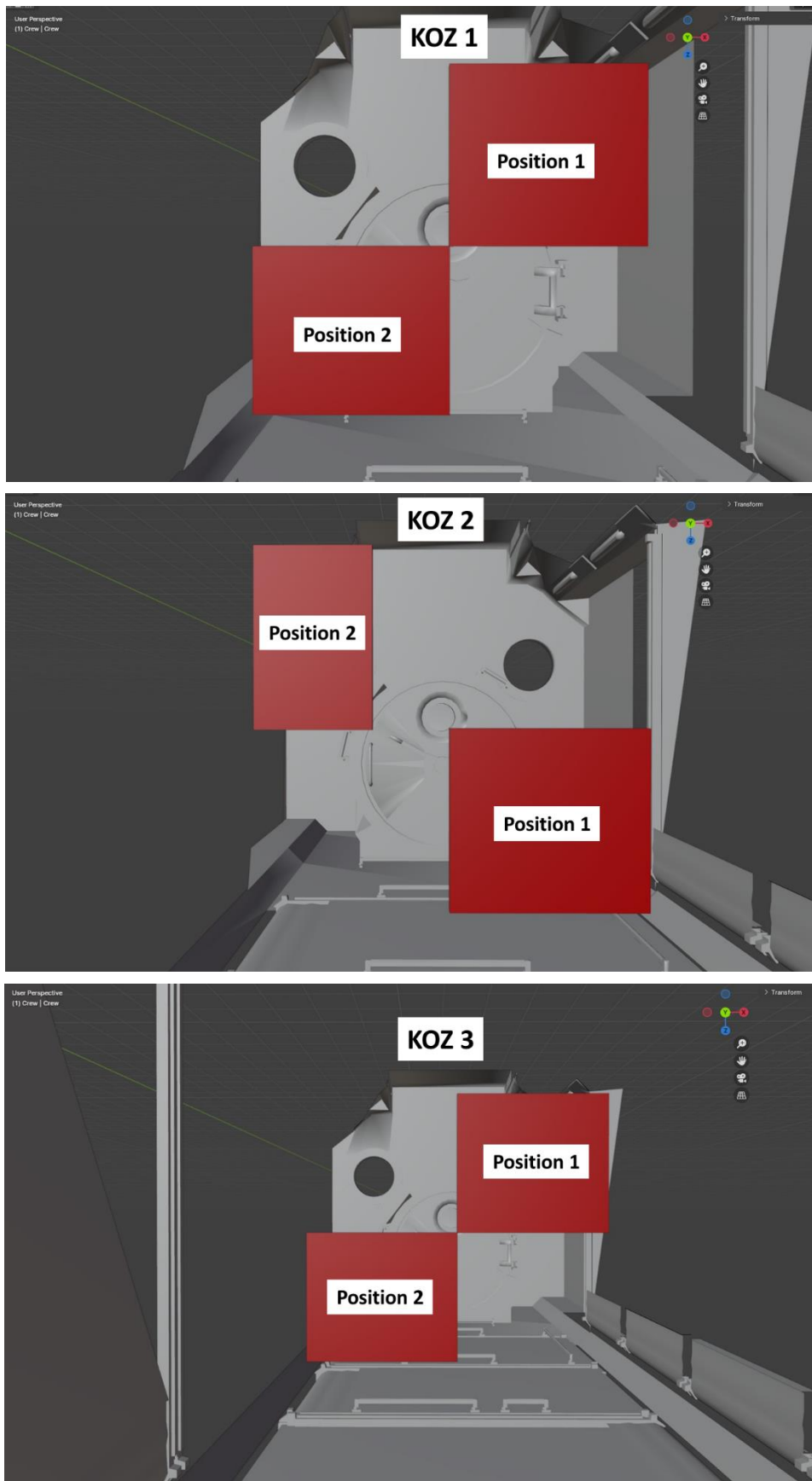


Figure 2.2.5-3 Preliminary round KIZs/KOZs (Front View)

KOZ/KIZ coordinates are shown in Table 2.2.5-1.

(x_{min} , y_{min} , z_{min}) and (x_{max} , y_{max} , z_{max}) are defined in Figure 2.2.5-4.

Table 2.2.5-1 Location coordinates of the obstacles

			x_{min}	y_{min}	z_{min}	x_{max}	y_{max}	z_{max}
KOZ	KOZ 1	Position 1	10.87	-9.5	4.27	11.6	-9.45	4.97
		Position 2	10.25	-9.5	4.97	10.87	-9.45	5.62
	KOZ 2	Position 1	10.87	-8.5	4.97	11.6	-8.45	5.62
		Position 2	10.25	-8.5	4.27	10.7	-8.45	4.97
	KOZ 3	Position 1	10.87	-7.40	4.27	11.6	-7.35	4.97
		Position 2	10.25	-7.40	4.97	10.87	-7.35	5.62
KIZ	KIZ 1		10.3	-10.2	4.32	11.55	-6.0	5.57
	KIZ 2		9.5	-10.5	4.02	10.5	-9.6	4.8

*The origin of the coordinate axis is set outside of Kibo.

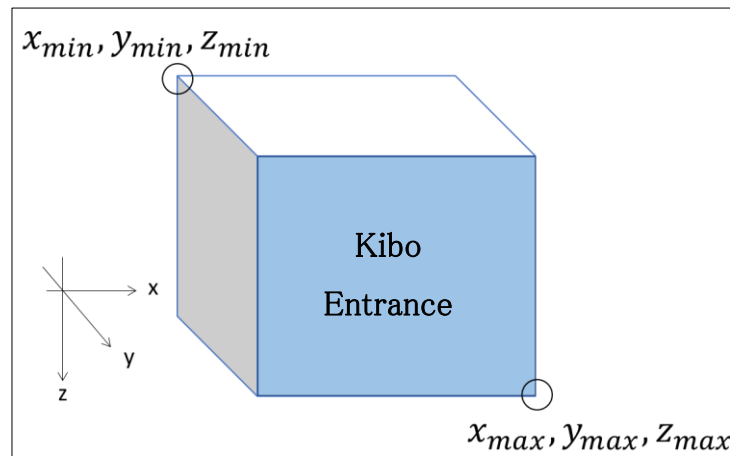


Figure 2.2.5-4 Kibo coordinate definition

2.2.6. 10 Runs per APK

In the preliminary round, each Android Application Package (APK) will be automatically executed 10 times and will include random elements to make it fair for everyone. The position of the active targets and random elements will be different for every run.

This will enable ranking that is not affected by environmental factors or disturbances, and will help to avoid accidental outcomes (both positive and negative). All participants will compete under the same conditions.

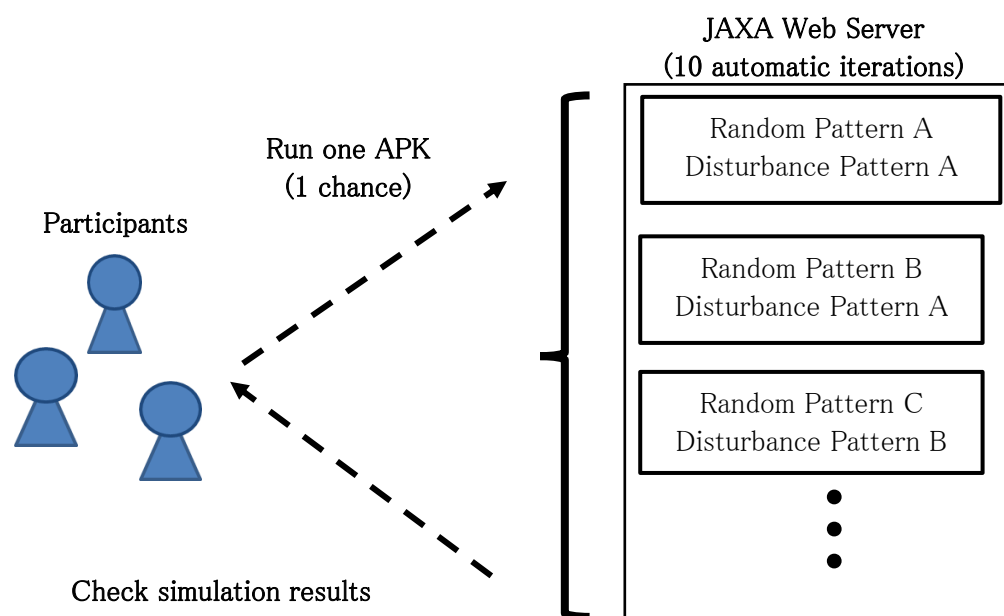


Figure 2.2.6-1 10 Runs

2.2.7. Ranking from Average of 10 Runs

Your APK will be ranked based on your average score from 10 runs.

In previous years, rankings were determined using the worst result from 10 runs to ensure that the mission could be completed even under the worst conditions expected in space. However, in the current mission scenario, the distance varies depending on the target position, which makes using the worst value ineffective. Therefore, rankings will be determined using the average value instead.

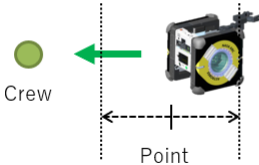
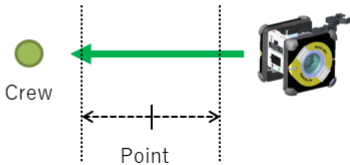
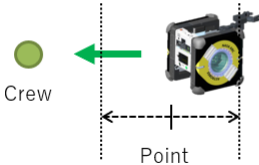
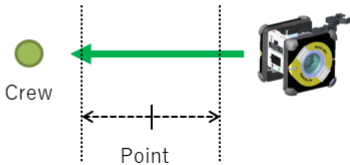
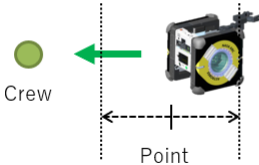
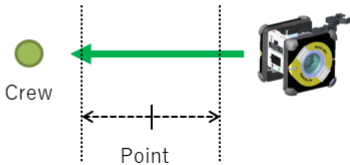
Please see Section 2.3 for details regarding scoring criteria.

2.3. Scoring

2.3.1. Factors

Your team's score will be calculated based on the following factors.

Table 2.3.1-1 Scoring Factors for Qualifiers

#	Factors	Details				
1	Matching of Area and Item	Points will be awarded if the type and number of Lost Items randomly placed in each Area are accurately processed and recognized. Lost Item displays are divided into difficulty levels, with higher levels (more difficult image processing) resulting in higher scores.				
2	Reporting coordinate of the patrol's completion	<p>Scoring is based on the arrival coordinate when the patrol's completion report is submitted.</p> <p>Points will be awarded if the coordinates reached are within 0.30m from the given coordinates.</p> <table><tr><th>Successful</th><th>Unsuccessful</th></tr><tr><td></td><td></td></tr></table>	Successful	Unsuccessful		
Successful	Unsuccessful					
						
3	Photo Angle of Target Item	<p>Scoring based on the angle of view of the camera when reporting the location of the Target Item.</p> <p>The angle of view is determined based on the position and posture of the acquired NavCam and Target Item, and points will be awarded if the angle of view is within 30°.</p>				
4	Photo Position of Target Item	<p>Scoring based on the coordinates when reporting the location of the Target Item.</p> <p>Scores will be given if the coordinates obtained are within 0.9 m from a plane of the area.</p>				
5	Mission Time Remaining	The time limit remaining at the time of reporting the Target Item is converted into points and additional points are awarded. There is a limit to the additional points based on the remaining time. If you				

		complete the mission with a certain amount of time remaining, you will receive a uniform amount of additional points.
--	--	---

2.4. Participation in the Preliminary Round

2.4.1. How to Participate in the Preliminary Round

Participants must submit a preliminary round APK before the deadline.

(All figures below are screenshots from past Kibo-RPCs.)

2.4.2. Preliminary Round APK Submission

Participants can compete in the preliminary round by preparing and submitting a preliminary round APK by the deadline. After submitting one APK they can submit another APK as long as it is before the deadline.

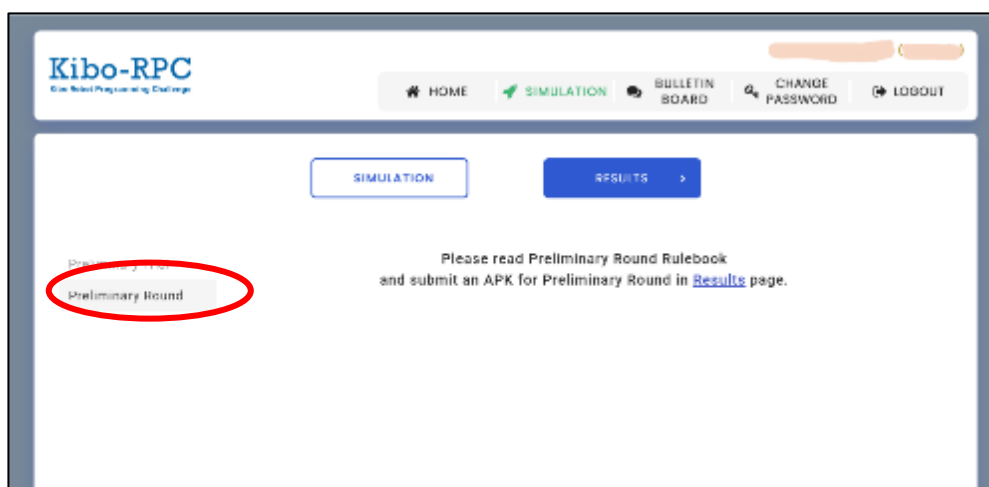


Figure 2.4.2-1 Preliminary Round

Before the preliminary round, the web simulator RESULTS screen will change as shown in Figure 2.4.2-1. Before submitting your APK, you must first run it in the Preliminary Trial simulator. After that, select the program you want to submit from the Preliminary Trial RESULTS screen and click the newly-added SUBMIT button to submit the APK. SUBMIT will change to CANCEL. Click CANCEL and follow the same procedure to send another APK at any time before the deadline.

Note: The SUBMIT button can only be pressed for APKs in Finished status in Preliminary Trial. If you only have an APK with a Failed status, you will not be able to participate in the Preliminary round. Please make sure to create an APK with a Finished status in the Preliminary Trial.

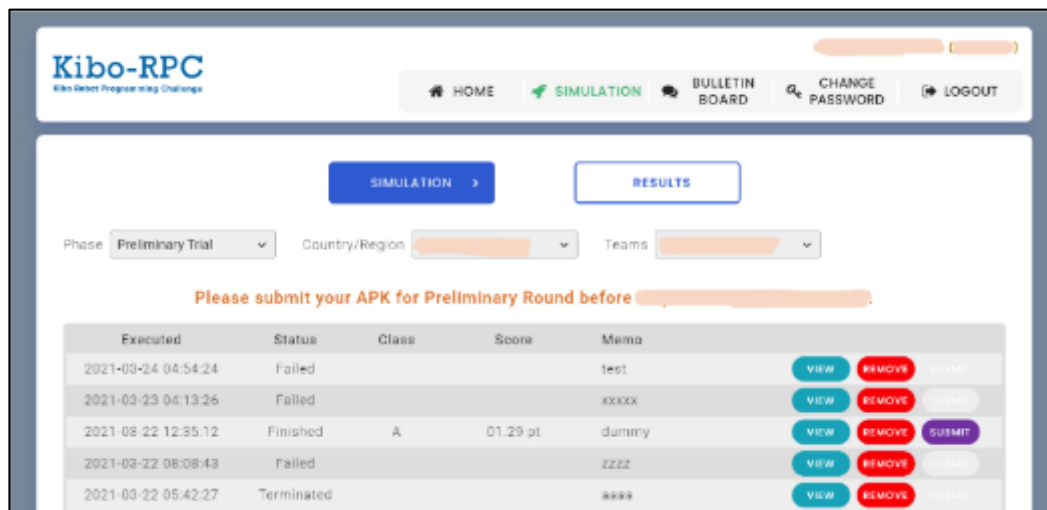
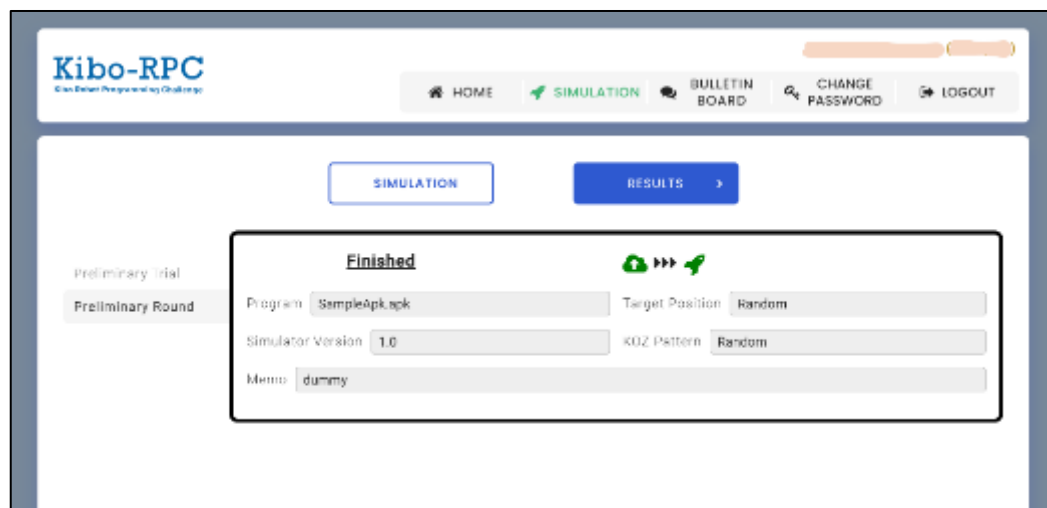


Figure 2.4.2-2 After Changing to Preliminary Trial

Submitted APKs can be viewed from the Preliminary Round tab in Figure 2.4.2-3. You can cancel an APK you have submitted on the RESULTS screen, but be aware of the submission deadline, as the SUBMIT and CANCEL buttons cannot be pressed after the submission deadline.

Figure 2.4.2-3 Preliminary Round



3. Final Round

3.1. Final Round Schedule

Only representative teams can participate in the final round. Teams may refine their programs from the Preliminary Round for the in-orbit finals and submit the APK and source code before the deadline. Please see Section 3.4 for details.

1) Draft source code submission deadline: Late July 2024 (JST)^{*1)}

2) In-orbit Final Program Submission Deadline: Late August 2024 (JST)^{*2)}

*1) JAXA will check APK source codes to ensure that they will not have a negative impact on the Astrobee and if necessary ask participants to modify the code.

Please submit only the source code for the pre-check. Submission instructions will be provided at a later date.

Due to the short revision period, please make arrangements in advance.

(Many modifications are required each year.)

*2) You will be required to **submit both APK and source code when submitting** the final version.

Submission instructions will be provided at a later date, but please refer to section 3.4.

3.2. Game Rule

3.2.1. Game Scenario

In the final round, each team will create a program to move an Astrobee on the ISS from the starting position to the targets while avoiding KOZs, and then use the laser to irradiate the activated targets and report to the astronaut within the time limit.

The game scenario is basically the same as in the preliminary round, but the **number of targets and the movements required of the Astrobee when reporting mission completion will be different.**

1. Astrobee starts from the Docking Station
2. After the starting, Astrobee will patrol the several candidate sites for the location of objects prepared in the Kibo module while avoiding KOZ^{*2}.
3. After it finishes its patrol, it moves to the astronaut and reports what objects (images) it found and where.
4. It asks the astronaut to tell it what he/she is looking for by reading the image in his/her hands.
5. Astrobee will move close to the object the astronaut is looking for and it will take a picture of it.
6. After the photo is taken, the Signal Lights will light up to inform the astronaut of the location of the object and the game will end.

^{*1} Time limit: **5 minutes** including mission completion report.

^{*2} Keep Out Zone (KOZ): An area Astrobee cannot enter and attempts to enter the Keep Out Zone will be denied.

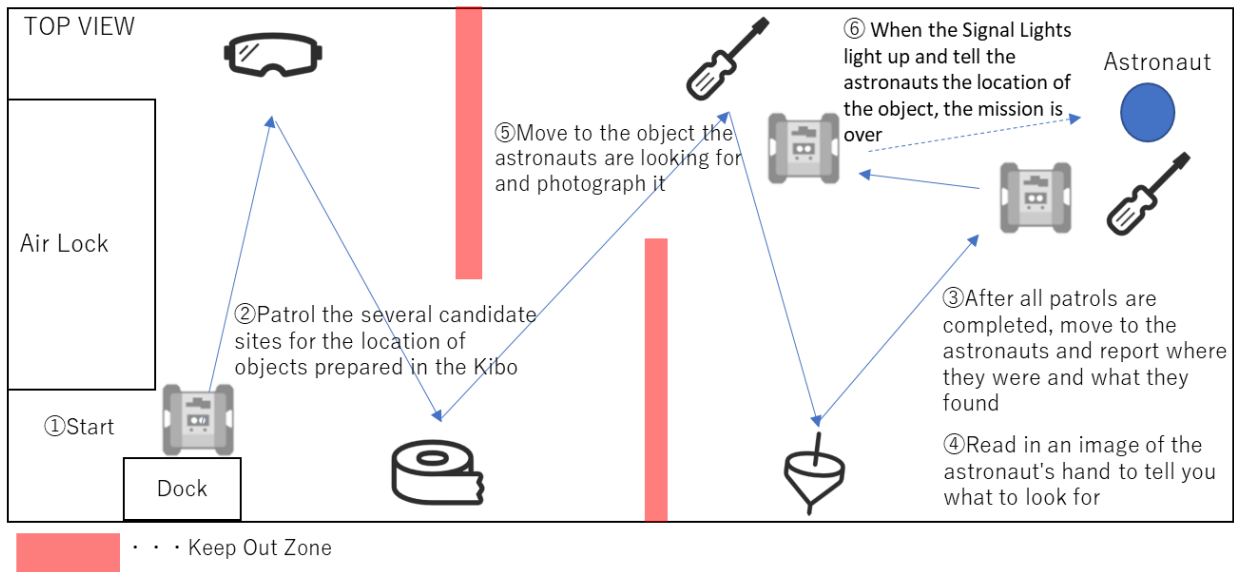


Figure 3.2.1-1 Final Round Game Flow

3.2.2. Preconditions

Table 3.2.2 Preconditions for Final Round

#	Content
1-4	Conditions for start and finish positions, Area, etc. are the same as for the Preliminary round. Please refer to section 2.2.2 for details. * Coordinate information may be revised in the future.
5	Some information on KOZ/KIZ may change from the preliminary round. We will release details as soon as they are finalized.

Creating a program that can perform well in the actual environment on board the ISS is important as the environmental conditions in orbit differ from those of the simulation.

3.2.3. Objects

There will be no major changes from the preliminary round. Details will be released as soon as they are finalized.



3.2.4. Mission Complete Report

There will be no major changes from the preliminary round. Details will be released as soon as they are finalized.

3.2.5. Keep-In-Zone (KIZ) & Keep-Out-Zone (KOZ)

Details will be released as soon as they are finalized.

3.2.6. 1 Run per APK

In the same way as in the preliminary round, teams will submit one APK, but in the final round it will only be run once on the ISS using the Astrobee. It will be impossible to redo or stop once started, so do your best because this will be a one-off chance.

If the Astrobee experiences a problem such as getting stuck, under the rules in Section 3.2.10 you will be given the opportunity for a re-run.

3.2.7. 5 Minute Time Limit

If the time limit is exceeded, the APK will automatically stop. Please create a program that can complete the mission within the time limit. Additionally, even if the time limit is not reached, if the Astrobee becomes stuck or loses its self-position, it will automatically be considered game over. If the Astrobee stops for a long time for reasons other than to process target laser irradiation images, or if it is judged that further action cannot be expected, the run may be terminated before the time limit.

3.2.8. APK Operation on the Day of the Final Round

Participants may not operate their APKs on the day of the final round.



Submitted APK will be checked by the JAXA/NASA technical team and preinstalled on the Astrobee. APKs are started with an execution command from ground operators.

3.2.9. Final Round Run Order

In the final round, teams will be divided into three tiers according to the results of the preliminary round and runs will be performed in that order. An example of team tier grouping is shown in Table 3.2.10.

*Please note that changes may be made to the tiers

Table 3.2.10 Team Assignments

Tier	Preliminary round Score Results
1st Tier	1 st place
	2 nd place
	3 rd place
	4 th place
2nd Tier	5 th place
	6 th place
	7 th place
	8 th place
3rd Tier	9 th place
	10 th place
	11 th place
	12 th place

If the Astrobee gets stuck due to a problem in orbit, the team will be given another chance to run the mission again before moving on to the next tier as long as there is enough time left in the event. However, if the problem is caused by the APK created by the participant, there will be no rerun. Please note that there is limited time to conduct the competition in orbit, and teams with lower rankings in the preliminary round may not be able to run their mission on the day of the final round. For more information, please refer to Figure 3.2.10.

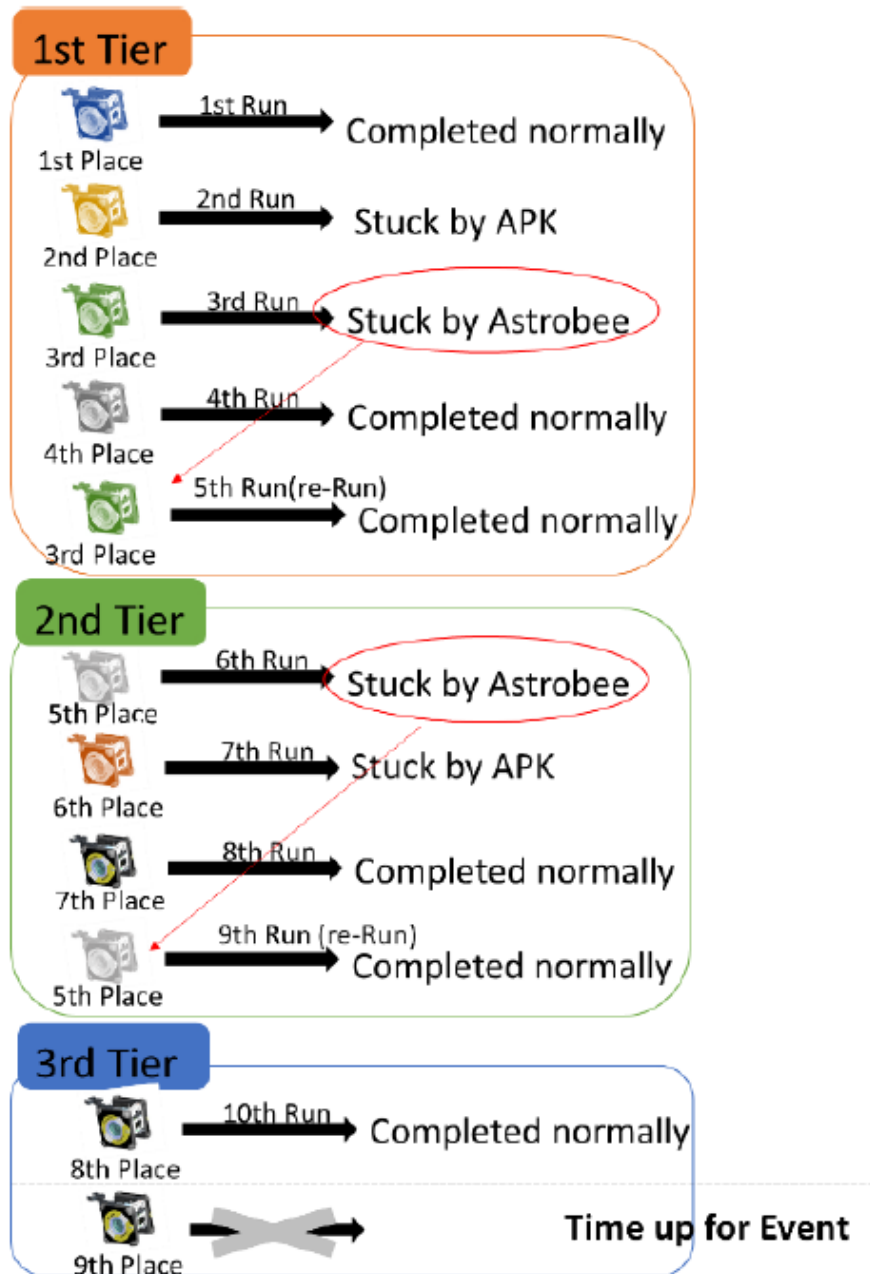


Figure 3.2.10 Final Round run order

3.3. Scoring

3.3.1. Factors

The scoring factors are the same as in the Preliminary round. Please refer to section 2.3.1 for details.

3.3.2. Judging

Only one run is performed for each team in the final round and that result will be their score.

3.4. Participating in the Final Round

Participants in the final round must do the following.

- (1) You must change the APK application ID and name as shown in Table 4.4 and included you country/region name. When submitting your APK, check that you have made the changes before uploading it to the Web Simulator. The Kibo-RPC Secretariat uses these names to identify the file when installing and executing the APK. Please refer to Section 3.3.3 of the Programming Manual for details on how to set the application ID, etc.

Table4.4 Various Name Rules

Country	Application ID	APK name	APK file name	Short name
Australia	jp.jaxa.iss.kibo.rpc.australia	australia	australia.apk	australia
Bangladesh	jp.jaxa.iss.kibo.rpc.bangladesh	bangladesh	bangladesh.apk	bangladesh
Japan	jp.jaxa.iss.kibo.rpc.japan	japan	japan.apk	japan
Malaysia	jp.jaxa.iss.kibo.rpc.malaysia	malaysia	malaysia.apk	malaysia
Nepal	jp.jaxa.iss.kibo.rpc.nepal	nepal	nepal.apk	nepal
Philippines	jp.jaxa.iss.kibo.rpc.philippines	philippines	philippines.apk	philippines
Korea	jp.jaxa.iss.kibo.rpc.korea	korea	korea .apk	korea
Singapore	jp.jaxa.iss.kibo.rpc.singapore	singapore	singapore.apk	singapore
Taiwan	jp.jaxa.iss.kibo.rpc.taiwan	taiwan	taiwan.apk	taiwan
Thailand	jp.jaxa.iss.kibo.rpc.thailand	thailand	thailand.apk	thailand
UAE	jp.jaxa.iss.kibo.rpc.uae	uae	uae.apk	uae
UNOOSA	jp.jaxa.iss.kibo.rpc.unoosa	unoosa	unoosa.apk	unoosa
USA	jp.jaxa.iss.kibo.rpc.usa	usa	usa.apk	usa

(2) Send APK and source code

Please refer to section 3.4.1 of the Rulebook for details.

(3) Confirm that everything is completed

Follow the checklist in Table 3.4-2 to confirm that you have completed the items to be performed for the Final Round.

(4) API Updates

Please comply with the instructions of the Secretariat with regard to any API updates.

Table 4.4-2 Checklist

No	Item	Description	Related Section(s)
1	Application ID	Change the application ID of the APK	Section 4.4(2) PG Manual Section 3.3.3
2	Rename the APK	Rename APK as per the regulations	Section 4.4(2) PG Manual Section 3.3.3
3	Rename APK file	Change APK file name as per the regulations	Section 4.4(2)
4	Change the APK short name	Change APK short name as per the regulations	Section 4.4(2) PG Manual Section 3.3.3
5	MD5	Create the APK's MD5	Section 4.4.1(2)
6	Submission	Submit the APK	Section 4.4.1(1)
7		Submit the source code	Section 4.4.1(2) Section 4.4.1(3)
8	Change the API	Update the game API (If changes are necessary, there will be separate instructions from the secretariat.)	—

PG Manual: Programming Manual

3.4.1. Submit APK and Source Code

You need to submit your program by the deadline for the final round. After submission, JAXA and NASA will review the source code in advance for safety reasons. Therefore, please submit the APK and source code according to the following procedure. **(At the time of the preliminary review, only the source code will be submitted.)**

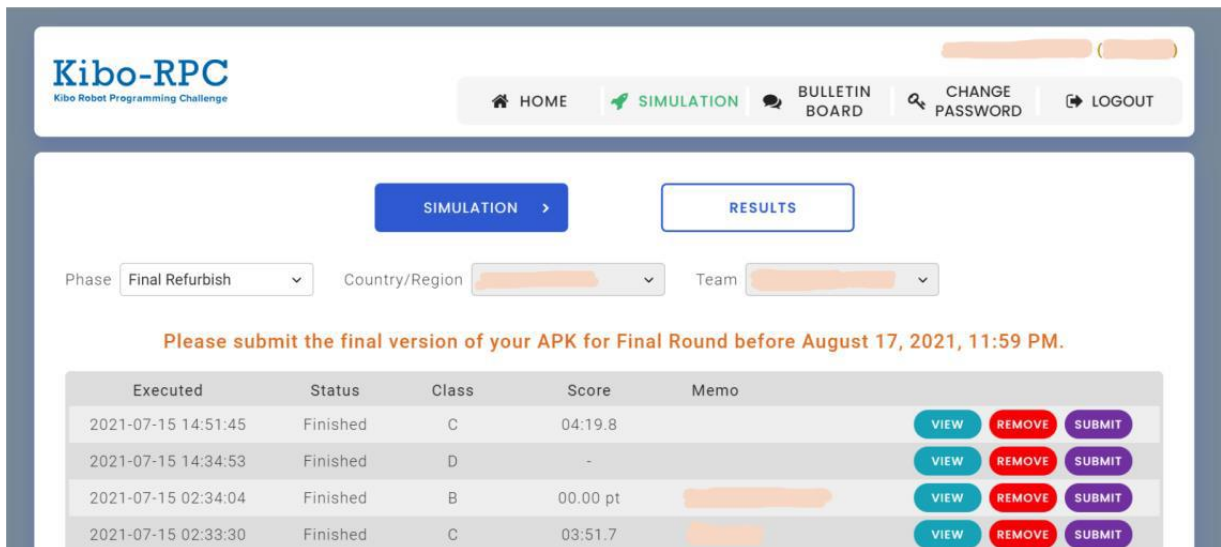
Due to the short time available for code modifications, please keep your schedules clear.

(Many modifications are required each year.)

(1) APK Submission

Select the APK in the RESULTS tab and check the SUBMIT. (Please see Figure 3.4.1-1 (All figures below are screenshots from previous Kibo-RPCs.))

*Please rename the APK file as per Table 3.4 before submitting the file.



The screenshot shows the Kibo-RPC (Kibo Robot Programming Challenge) web interface. At the top, there is a navigation bar with links for HOME, SIMULATION, BULLETIN BOARD, CHANGE PASSWORD, and LOGOUT. Below this, there are two tabs: SIMULATION and RESULTS. The RESULTS tab is selected. Under the RESULTS tab, there are dropdown menus for Phase (set to 'Final Refurbish'), Country/Region, and Team. A message states: 'Please submit the final version of your APK for Final Round before August 17, 2021, 11:59 PM.' Below this message is a table with columns: Executed, Status, Class, Score, Memo, and three action buttons: VIEW, REMOVE, and SUBMIT.

Executed	Status	Class	Score	Memo	VIEW	REMOVE	SUBMIT
2021-07-15 14:51:45	Finished	C	04:19.8		VIEW	REMOVE	SUBMIT
2021-07-15 14:34:53	Finished	D	-		VIEW	REMOVE	SUBMIT
2021-07-15 02:34:04	Finished	B	00.00 pt		VIEW	REMOVE	SUBMIT
2021-07-15 02:33:30	Finished	C	03:51.7		VIEW	REMOVE	SUBMIT

Figure 3.4.1-1 APK Submission Screen



(2) **Source Code Submission**

Please submit the source code by email to the Kibo-RPC Secretariat in accordance with the following instructions.

1. Create the PK file MD5

The Kibo-RPC Secretariat will review the APK and MD5 submitted through the website.

(A) For Windows

Execute the following command from a command prompt.

```
> cd [path to apk directory]
```

```
> certutil -hashfile [apk file name] MD5 > apk.md5
```

"apk.md5" is created and it includes 32-digit hash value.

(e.g.)

```
> cd C:\DefaultApk\app\build\outputs\apk\
```

```
> certutil -hashfile app-debug.apk MD5 > apk.md5
```

(B) For Ubuntu

Execute the following command from a terminal.

```
$ cd [path to apk directory]
```

```
$ md5sum [apk file name] > apk.md5
```

"apk.md5" is created and it includes 32-digit hash value.

(e.g.)

```
$ cd ~/DefaultApk/app/build/outputs/apk/
```

```
$ md5sum app-debug.apk > apk.md5
```

2. Delete APK and large files/directories.

Delete the APK file.

(Be careful not to delete MD5 at this time)

```
- [root dir]/app/build/outputs/apk/*.apk
```

Next, delete the following directories.

```
- [root dir]/app/build/generated/
```

- [root dir]/app/build/intermediates/
- [root dir]/app/build/tmp/
- [root dir]/.gradle/

3. Compress (zip, tar, etc.) the root directory and send it to the office.
The approximate size of a compressed file is several hundred KB to several MB. Please make sure that all files (Java source files, md5) are included in the compressed file and send it to the office.
Sending instructions will be provided at a later date.

3.5. Organizing the Event

The 4th Kibo-RPC will be held in a similar format as that of the 3rd Kibo-RPC, which is shown in Figure 4.5. JAXA will run the Astrobeer and finalists' APKs in advance. Footage of the competition is scheduled to be broadcast live, and finalists will be able to watch footage of their own runs. A final round event, featuring commentary by experts watching the pre-run footage, will be held at a later date. Finalists will be contacted by the Secretariat via email with more details.

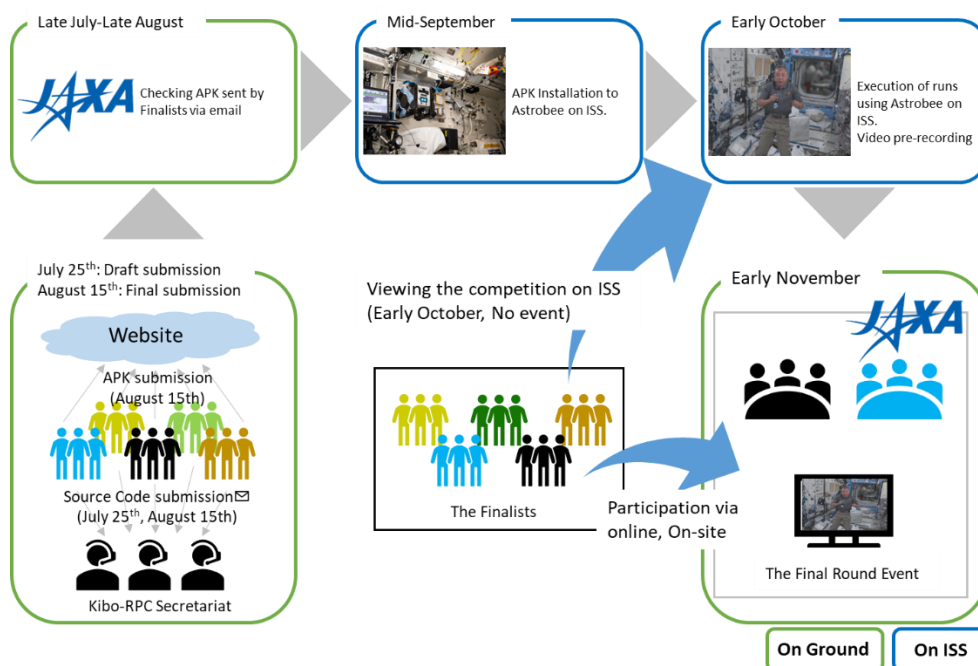


Figure 3.5 Flow up to the day of the event

Appendix

This will be the definition of terms used in the 5th Kibo-RPC.

Terms	Definition
Kibo-RPC	Abbreviation for Kibo Robot Programming Challenge, a programming competition using robots on the ISS.
ISS	Abbreviation for International Space Station.
Kibo	The Japanese Experiment Module developed by JAXA on the ISS. Also known as JEM (Japanese Experiment Module), this is where this game will take place.
Astrobee	Free-flyer robot developed by NASA that will be used in this game.
Area	A plane representing a potential location for a Lost Item, set on an ISS wall or Airlock.
Lost Item	Generic name for images placed in each Area.
Target Item	A word to describe what the astronaut is looking for during the game.
AR Tag	An AR marker used to identify the location and orientation of an Item.
KIZ	Abbreviation for Keep-In-Zone, the range within which an Astrobee can move.
KOZ	Abbreviation for Keep-Out-Zone, which is set up within the KIZ and is used by Kibo-RPC as a no-entry zone within the Japanese Experiment Module Kibo.