

The 5th Kibo Robot Programming Challenge Rulebook



Version 1.2 (Revision date: August 21, 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	-	
July 16, 2024	1.1	3.1	Update the schedule for the final	
			round	
		3.2	Modified description (no change	
			from the preliminary round rules)	
		3.4	Update the API	
			Added detailed descriptions of how	
			to submit an APK and source code	
August 21, 2024	1.2	3	correction of errors	
		3.4	Update the Table 3.4-2 Checklist	



Table of Contents

1. Introduction	1
2. Preliminary Round	2
2.1. Preliminary Round Period 2.2. Game Rules	2
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	
3.1. Final Round Schedule	
3.2.1 Camo Scopario	
3.2.2. Preconditions	
3.2.3. Objects	
3.2.4. Mission Complete Report	
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	
3.2.8. APK Operation on the Day of the Final Round	19
3.2.9. Final Round Run Order	21
3.3. Scoring	
3.3.1. Factors	
3.3.2. Judging	23
3.4. Participating in the Final Round	



3.4.1. Submit APK and Source Code	26
3.5. Organizing the Event	28
Appendix	29



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 Astrobee in the Preliminary round round, please create a program to complete the following game using JAXA's web simulation environment. Within the time limit, move Astrobee 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, Astrobee 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 Astrobee moves in front of the object that the astronaut is looking for and takes a picture of it. The mission ends when the Astrobee shines its Signal Lights, which signal that it has found the object, and tells the astronaut where the object is.

- 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^{*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. 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}Keep Out Zone (KOZ): Areas where Astrobee 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

#	Content						
1	The starting position is the Docking Station, and the timer starts once Astrobee undocks.						
	There are 10 types of Lost Items. Prepare AR tag on the same plane as the printed surface of Lost Item.						
	The search area for Lost Item (hereinafter referred to as "Area") is limited to						
2	four locations.						
Area is specified as a plane, and one Lost Item is placed somewhere v							
	Area.						
	Lost Item placement is random.						
3	Target Item is determined at random from among the Lost Items used in the						
	game.						
	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.						
	1. Orientation of the position of the starting point (StartPoint)						
	2. Astronaut's (ExploringCompletionPoint) position and orientation						
4	3. Location and size of each Area						
	4. Total number of Areas						
	5. Candidates for Lost Item that astronauts are looking for						
	6. Location and size of Keep Out Zone						
	7. Parameters for given angle and distance						
	There are KOZs somewhere in the path of the Astrobee. Coordinates will be						
5	provided to you beforehand. Please see Section 2.2.5 for details.						
	* Astrobee cannot pass through KOZs.						

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

Point	Coordinates			Orientation			
	Х	у	Z	х	у	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



					•		
		x_min	y_min	z_min	x_max	y_max	z_max
	1	10.42	-10.58	4.82	11.48	-10.58	5.57
A roo*	2	10.3	-9.25	3.76203	11.55	-8.5	3.76203
Area	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

Table 2.2.2-3	Coordinate	Information	(Area)
---------------	------------	-------------	--------

* 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	AR tag information (The size remains the same for both the Lost Item					
	Items	and the Target Item)					
		15 cm Lost Item, 1	27 cm	1.25 cm 3.75 cm 3.75 cm 3.75 cm 7.0 cm			
			20.0 cm				
		List of Lost Item kapton tape	top	screwdriver			
			\Diamond	Contraction of the second seco			
		beaker	hammer	pipette			
			A				
		wrench	thermometer	watch			
		25		Ø			
		goggle					
		* Lost Item size varies	depending on image d	ifficulty.			





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 Astrobee'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 Astrobee to move only within the KIZs.

 \xrightarrow{x}



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)



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_maxm z_max) are defined in Figure 2.2.5-4.

			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

Table 2.2.5-1 Location coordinates of the obstacles

*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.					
	Reporting coordinate 2 of the patrol's completion	Scoring is based on the arrival coordinate when the patrol's completion report is submitted. Points will be awarded if the coordinates reached are within 0.30m from the given coordinates.					
2		Successful Unsuccessful Original Original					
3	Photo Angle of Target Item	Scoring based on the angle of view of the camera when reporting the location of the Target Item. 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°.					
4	Photo Position of Target Item	Scoring based on the coordinates when reporting the location of the Target Item. Scores will be given if the coordinates obtained are within 0.9 m from a plane of the area.					
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

Bfore 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.

Kibo-RPC		*	HOME 🜱 SI	MULATION SULLETIN BOARD	a, CHANGE DOGOUT
Phase Preliminary Trial	Country	SIMULATION	Jiminary Pound	RESULTS	×
Executed	Status	Class	Score	Mema	
2021-00-24 04:54:24	Failed			test	VIEW (REMOVE)
	marker of			WWWW -	
2021-03-23 04:13:26	Falled			A.X.A.X.X	VIEW REMOVE
2021-03-23 04:13:26 2021-08 22 12:85:12	Finished	A	01.29 pt	dummy	
2021-03-23 04:13:26 2021-08:22 12:85:12 2021-03-22 08:08:43	Finished	A	01.29 pt	dummy 2222	VEW REMOVE SUBMIT

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.



Kibo-RPC	A HOME	SIMULATION & BULLETIN & CHANGE & LOGOUT
	SIMULATION	RESULTS >
Preliminary Trial	Finished	& *** 4
Preliminary Round	Program SampleApk.apk Simulator Version 1.0	K02 Pattern Random
	Merrio dummy	



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: July 31, 2024 at 23:59 (JST)^{*1)}
- 2) In-orbit Final Program Submission Deadline: August 30, 2024 at 23:59 (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. Please refer to section 3.4 for submission instructions.

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.

Please refer to section 3.4 for submission instructions.



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 movements of the Astrobee when reporting mission completion will be different. There are no program changes, but in the real environment Astrobee runs SignalLights according to the API.**

- 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 Frow

3.2.2. Preconditions

Table 3.2.2 Preconditio	ns for Final Round
-------------------------	--------------------

#	Content
	Conditions for start and finish positions, Area, etc. are the same as for the
1-4	Preliminary round. Please refer to section 2.2.2 for details.
	* Coordinate information may be revised in the future.
5	The information on KOZ/KIZ doesn't change from the preliminary round.
(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
9	simulation.

3.2.3. Objects

There is no major changes from the preliminary round. The text in the lower right corner of the image will be changed. This will not affect the progress of the game.

```
Before: Item x (x is figure)
After: Lost Item x (x is figure)
```



3.2.4. Mission Complete Report

There is no major change from the preliminary round.

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

There is no major change from the preliminary round.

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

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		

Table 3.2.9 Team Assignments

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 3.2.9 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.



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.banglad	banglades	bangladesh.ap	banglades
	esh	h	k	h
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.philippin	philippines	philippines.apk	philippines
	es			
Korea	jp.jaxa.iss.kibo.rpc.korea	korea	korea .apk	korea
Singapore	jp.jaxa.iss.kibo.rpc.singapor	singapore	singapore.apk	singapore
	е			
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
Vietnam	jp.jaxa.iss.kibo.rpc.vietnam	vietnam	Vietnam.apk	vietnam

Table 3.4-1	Various Name Rules

(2) Send APK and source code

Please refer to section 3.4.1 of the Rulebook for details.

- (3) Confirm that everything is completedFollow 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 update the API with referring to the latest version of Programming manual Appendix 1. (Kibo-RPC API Ver.5.1.0)



Table 3.4-2 Checklist	Table	3.4-2	Checklist
-----------------------	-------	-------	-----------

No	Item	Description	Related Section(s)	
1	Application ID	Change the application ID of the APK	Section 3.4(2)	
	Application 1D		PG Manual Section 3.3.3	
2	Ponamo tho ADK	Ponamo APK as por the regulations	Section 3.4(2)	
		Rename Ark as per the regulations	PG Manual Section 3.3.3	
3	Ponamo APK filo	Change APK file name as per the	Section 3.4(2)	
	Rename AFR me	regulations		
4	Change the APK	Change APK short name as per the	Section 3.4(2)	
	short name	regulations	PG Manual Section 3.3.3	
5	MD5	Create the APK's MD5	Section 3.4.1(2)	
6		Submit the APK	Section 3.4.1(1)	
7	Submission	Submit the course code	Section 3.4.1(2)	
8	Change the API	Update the game API	PG Manual Appendix 1	
			- · · · · · · · · · · · · · · · · · · ·	
9	9 Establishment of the game	"startMission" is called at the beginning	PG Manual Section 7.1	
		of the program		
		After recognizing the Target Item,		
		"notifyRecognitionItem" is called when	PG Manual Section 7.1	
		moving to the Target Item		
		a take largetitem Snapshot is being	PG Manual Section 7.1	
10		No implementation of an infinite loops		
10		with for or while	PG Manual Section 5.1	
		No danger of infinite loops due to		
	Software Safety	recursion	PG Manual Section 5.1	
		Not being included infinite loops in the		
		image saving process when using	PG Manual Section 5.1	
		GameAPI 5.1.0		
11	Resource	No extra resources in the source code	_	

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. (<u>At the time of the preliminary review, only the source</u> 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 3.4-1 before submitting the file.

Kibo-RPC Kibo Robet Programming Challenge		# H	OME 🖪 SI		ETIN & CHANGE ARD PASSWORD (LOGOUT
Phase Final Refurbish Please subr	Country	SIMULATION /Region	> APK for Fina	RESULTS Team Team	Just 17, 2021, 11:59 PM.
Executed	Status	Class	Score	Memo	
2021-07-15 14:51:45	Finished	С	04:19.8		
2021-07-15 14:34:53	Finished	D			
2021-07-15 02:34:04	Finished	В	00.00 pt		
2021-07-15 02:33:30	Finished	С	03:51.7		

Figure 3.4.1-1 APK Submission Screen



(2) Source Code Submission

Please submit the source code in the folder specified by the secretariat. The secretariat will share with you about where to submit the source code.

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:\U00e4DefaultApk\u00e4app\u00e4build\u00e4outputs\u00e4apk\u00e4
- > 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
- 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 secretariat.

Please make sure that all files (Java source files, md5) are included in the compressed file. Please submit the source code in the folder specified by the secretariat. The secretariat will share with you about where to submit the source code.

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 Astrobee 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

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.

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