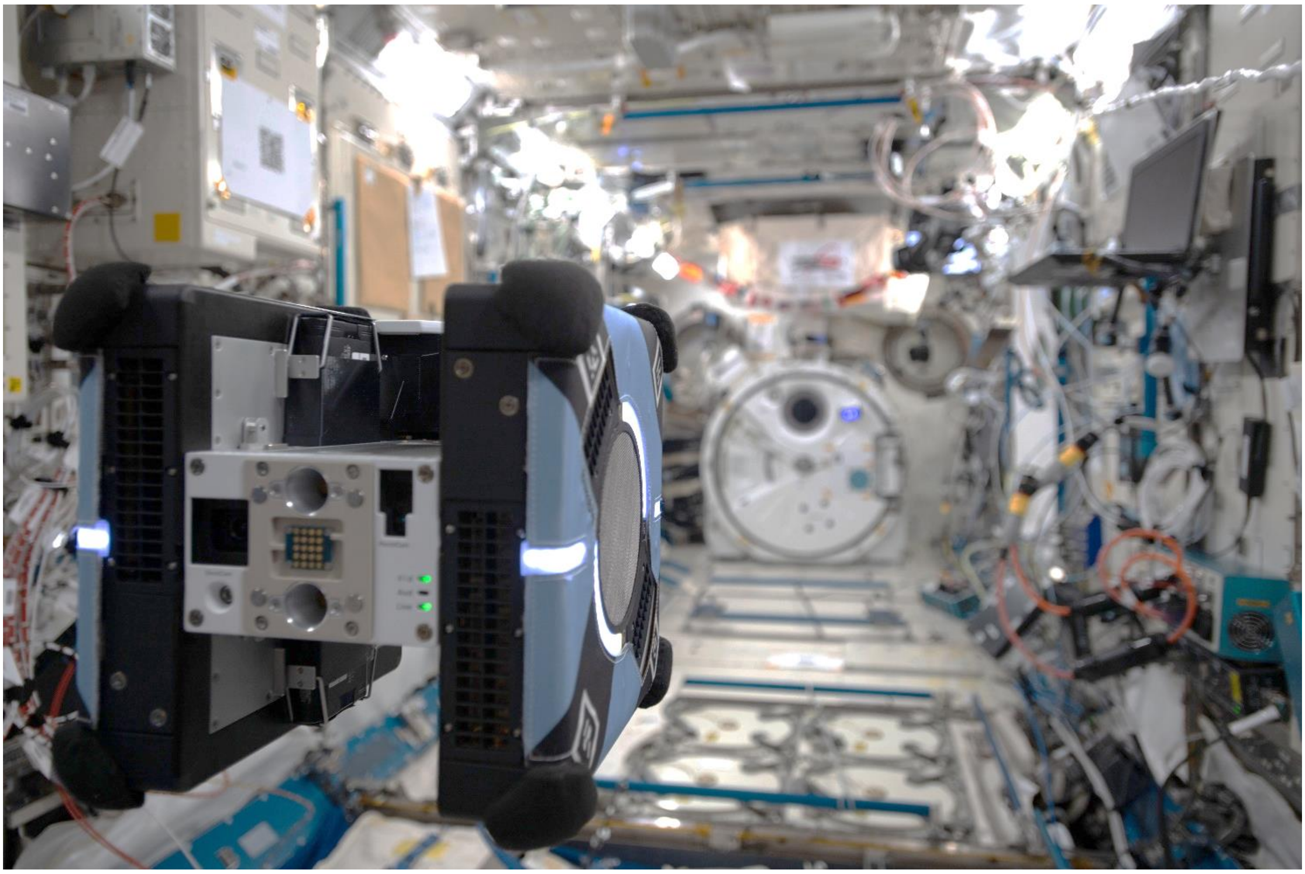
Version 2.1
Released Date: May 12th, 2021

2nd Kibo Robot Programming Challenge

Rulebook



Version 2.1 (Released Date: May 12th, 2021)

Japan Aerospace Exploration Agency (JAXA)

List of Changes

All changes to paragraphs, tables, and figures in this document are shown below;

Release Date	Revision	Paragraph(s)	Rationale
April 1st, 2021	1.0	All	-
April 14th, 2021	2.0	2.2.2	Corrected dock station coordinates.
		Figure 2.2.2-1	Added the position of P-A' for each KOZ pattern.
		Table 2.2.4-1	Changed the value of x_max in KIZ_2.
May 12th, 2021	2.1	2.1	Changed the schedule of the Preliminary Round period

Version 2.1

Released Date: May 12th, 2021

Contents

1. Introduction

Let's challenge your program to be the champion of 2nd Kibo Robot Programming Challenge!

All participants must need to join a preliminary competition with simulator to be selected as a national representative. Refer to Guidebook Section3 for a game content.

All teams will compete with other participants by your own programs developed before Preliminary Round using JAXA's simulation environment. Please refer to the Programming Manual for how to develop programs. Preliminary Round will be based on common scoring factors and game rules across the countries/regions. Since event cases of Preliminary Round vary from country/region to country/region, the details of information such as venues and schedules will be announced by each country's/region's POC. This Guidebook focuses on general rules across all the events. The winning teams of the Preliminary Round in each country/region are able to participate in the Programming Skills Round. Winners of the Programming Skills Round can advance to the Final Round. In the Final Round, finalist teams compete each other for the world championship by using free-flyer robots, Astrobee, on the ISS/ Kibo module!

2. Preliminary Round

2.1. Preliminary Round period

The Preliminary Round is carried out by all countries/regions in the Preliminary Round period. Information of Preliminary Round in each country is described on the Kibo-RPC official web site (https://jaxa.krpc.jp/). Please contact your country's/region's POC for more details.

Note: Participants will not be able to run web simulations from "Preliminary Round" tab. "Preliminary Trial" tab is always available.

APK submission period: June 7th through 21th, 2021

Preliminary Round period: June 22th through July 16th, 2021 (TBD)

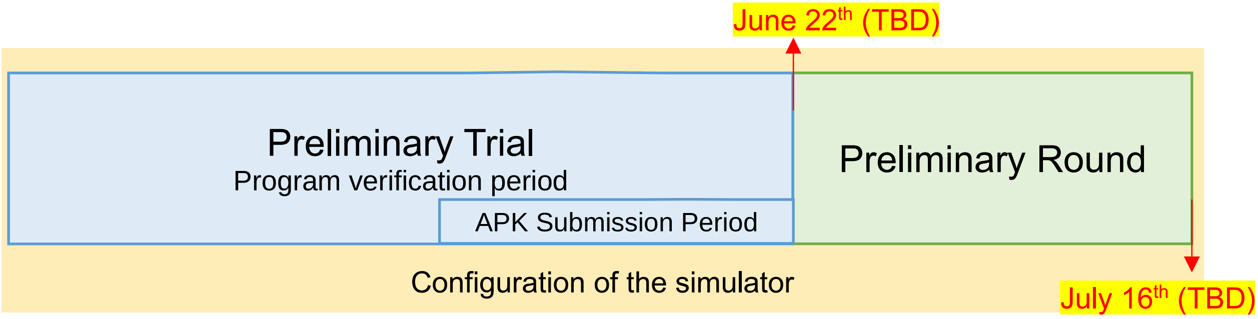


Figure 2.1-1 Preliminary Round Period

2.2. Game Rules

2.2.1. Game flow

All teams need to create a program such as following points to control NASA's Astrobee in JAXA's simulation environment.

1. Move the Astrobee from the dock station to Point-A
2. Read a QR code at Point-A and get the information of a point where target position can be in sight (Point-A') and Keep Out Zone (KOZ)*1 in front of the target.
3. Move to Point-A'
4. Irradiate the center of the target with a laser by utilizing information of relative position from the AR tags while avoiding the obstacle (KOZ).

   If you wish to fine adjust the position for better targeting, you can program to automatic re-try within the time limit. *2
5. Move to Point-B in backward motion with facing the Airlock direction. *3
6. Report "Mission Complete" to Astronaut. *4

*1 Keep Out Zone is an area where Astrobee cannot move into.

*2 The time limit is 5 minutes for one run. If the program does not complete the mission within 5 minutes, the actions after 5 minutes will not be reflected in the score. Make sure your program completes the mission within the time limit.

*3 You can make Astrobee turn around to move face forward instead of backward in JAXA's simulation environment. However, it will be difficult in the Final Round. See section 4.2.2.

*4 "Mission Complete" report is done by an API prepared by JAXA and visually displayed in JAXA's simulation environment. It will be different with the Final Round. See section 4.2.2.

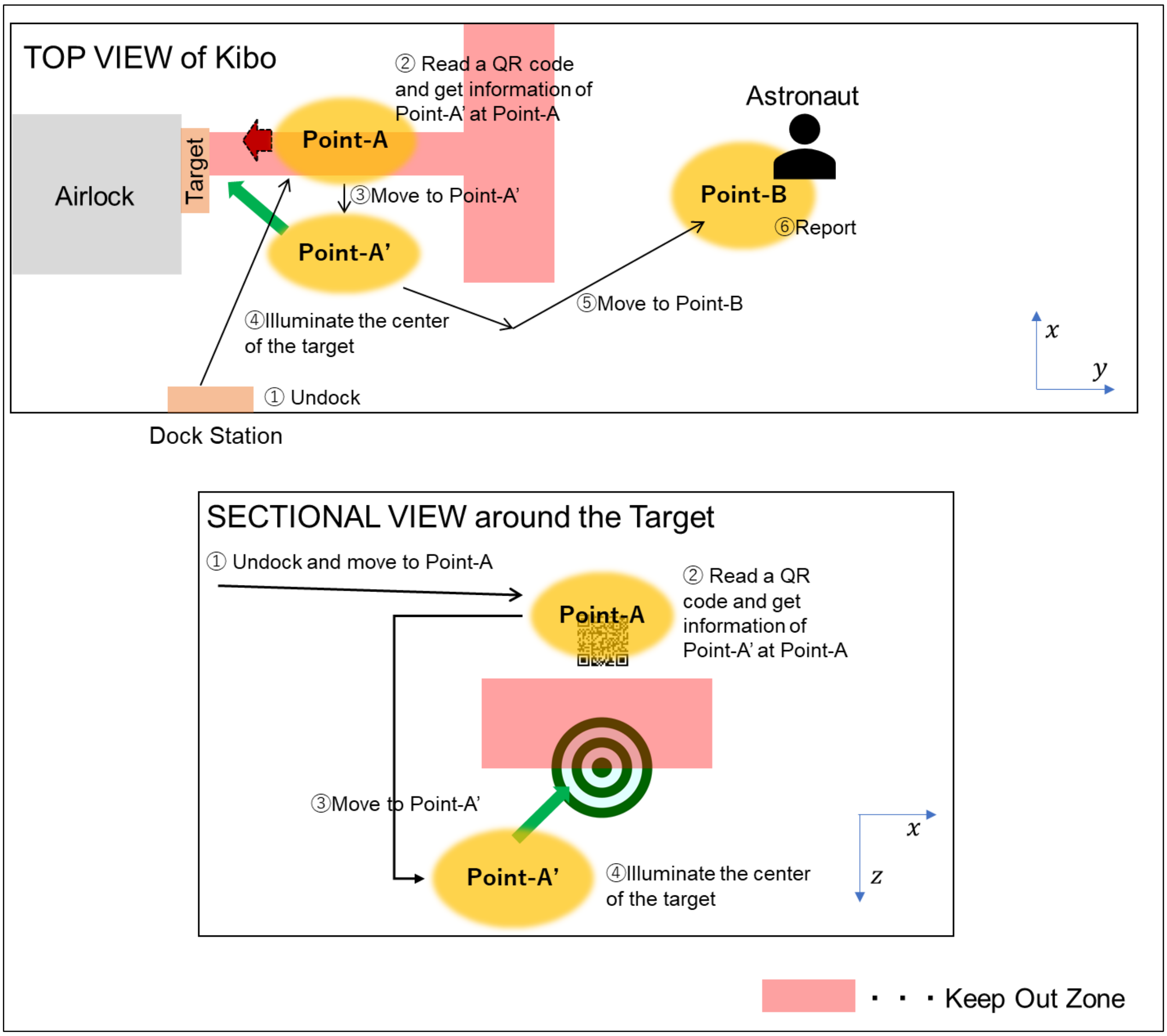

TOP VIEW of Kibo
② Read a QR code and get information of Point-A' at Point-A
Astronaut
Airlock
Target
Point-A
③Move to Point-A'
Point-B
⑥Report
Point-A'
⑤Move to Point-B
④Illuminate the center of the target
x
y
① Undock
Dock Station
SECTIONAL VIEW around the Target
① Undock and move to Point-A
Point-A
② Read a QR code and get information of Point-A' at Point-A
③Move to Point-A'
x
z
Point-A'
④Illuminate the center of the target
・・・Keep Out Zone

Figure 2.2-1 Game Outline of the Preliminary Round

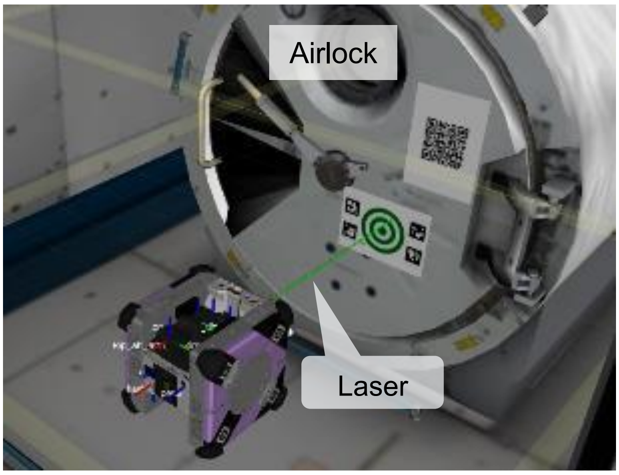


Figure 2.2-2 Astrobee Simulator Server Sample Screenshots

2.2.2. Preconditions

Table 2.2.2-1 Preconditions of the Preliminary Round

#	Content
1	The starting position is Dock station (Astrobee is docked). The coordinates are; Position (x, y, z) = (9.815, -9.806, 4.293) Orientation (x, y, z, w) = (1, 0, 0, 0)
2	Reading position of QR code is at Point-A to get the information of Point-A' and KOZ. The Point-A coordinates are; Position (x, y, z) = (11.21, -9.8, 4.79) Orientation (x, y, z, w) = (0, 0, -0.707, 0.707) At least one AR tag is in the NavCam FOV from Point-A'. The coordinates (x', y', z') of Point-A' are randomly determined in the following range. x': between $x_{target} + 0.08$ and $x_{target} + 0.16$ or between $x_{target} - 0.16$ and $x_{target} - 0.08$ y': $y' = y$ z': berween $z_{target} + 0.08$ and $z_{target} + 0.16$ or between $z_{target} - 0.16$ and $z_{target} - 0.08$ where, (x, y, z) is the Point-A coordinates and (x_{target}, y_{target}, z_{target}) is the coordinates of the center of the target. See also Figure 2.2.2-1 about the Point-A'.
3	There are AR tags near the target point. The position of the target point along with AR tags varies randomly. The relative distances of the target and AR tags are always the same.
4	There is a KOZ in front of the target.
5	The coordinates of Point-B are; Position (x, y, z) = (10.6, -8.0, 4.5) Orientation (x, y, z, w) = (0, 0, -0.707, 0.707)
6	About the QR code, AR tag and target, please refer to 2.2.3 Objects.
7	A Keep-Out Zone (KOZ)* that simulates an obstacle is set somewhere in the path from the Point-A' to Point-B. This KOZ is given as a precondition. About the detail, please refer to 4.5 Keep-Out-Zone. * Astrobee cannot move into KOZ.

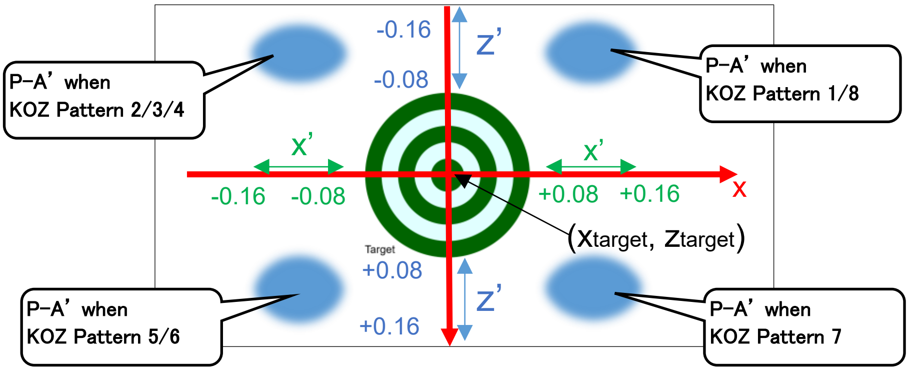
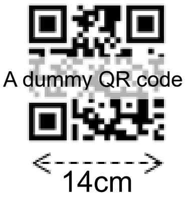
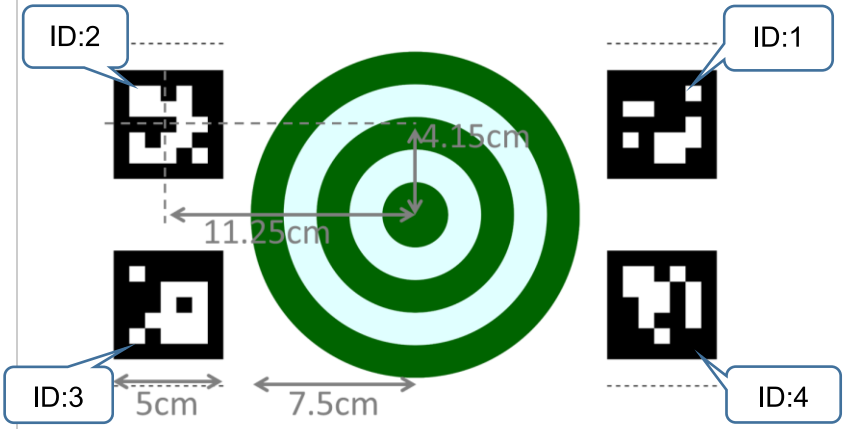


Figure 2.2.2-1 Definition of Point-A'

2.2.3. Objects

Table 2.2.3-1 Objects of the Preliminary Round

#	Name of object	Method
1	QR code tag	The size of QR code tag is 14cm square. Information format is followings. The format of the string obtained from the QR code is as follows.  Format : {"p":<pattern>,"x":<x>,"y":<y>,"z":<z>} <pattern> : KOZ pattern between 1 and 8. <x> : Position x of Point-A'. <y> : Position y of Point-A'. <z> : Position z of Point-A'. The sample is as follows. Sample : {"p":1,"x":10.23,"y":-8.12,"z":5.45} The orientation of Point-A' is always (0, 0, -0.707, 0.707)
2	Target Point	The size of AR tag is 5cm square. -It is located on both sides of the target. "Aruco.DICT_5X5_250" is used as dictionary. The size of target is 7.5cm radius. The distance between the center of AR and center of target is (|x|, |z|) = (11.25 cm, 4.15 cm) The IDs of the AR tags are 1, 2, 3 and 4, counterclockwise from the upper right. * The text "Target" will be printed in the Final Round, but not in the simulator.  ID:2 ID:1 4.15cm 11.25cm ID:3 5cm 7.5cm ID:4

2.2.4. Keep-In-Zone (KIZ) and Keep-Out-Zone (KOZ)

Keep-In-Zone (KIZ) is defined as the area where Astrobee can move around, and basically set along the walls of Kibo. It is a preset boundary in Astrobee and if the destination of the moving path of Astrobee is outside the KIZ, that is rejected. You need to design each moving path of Astrobee within the KIZ.

The Keep-Out-Zone (KOZ) are set inside the KIZ as a volumetric zone and used as some obstacles inside Kibo in the Kibo-RPC. You need to design each moving path of Astrobee to avoid the KOZ. (Figure 2.2.4-1, 2.2.4-2, 2.2.4-3 and Table 2.2.4-1) The information of KOZ (KOZ_1-n) in front of the target is given from QR code. It has 8 patterns and one of them is selected randomly. (Figure 2.2.4-4)

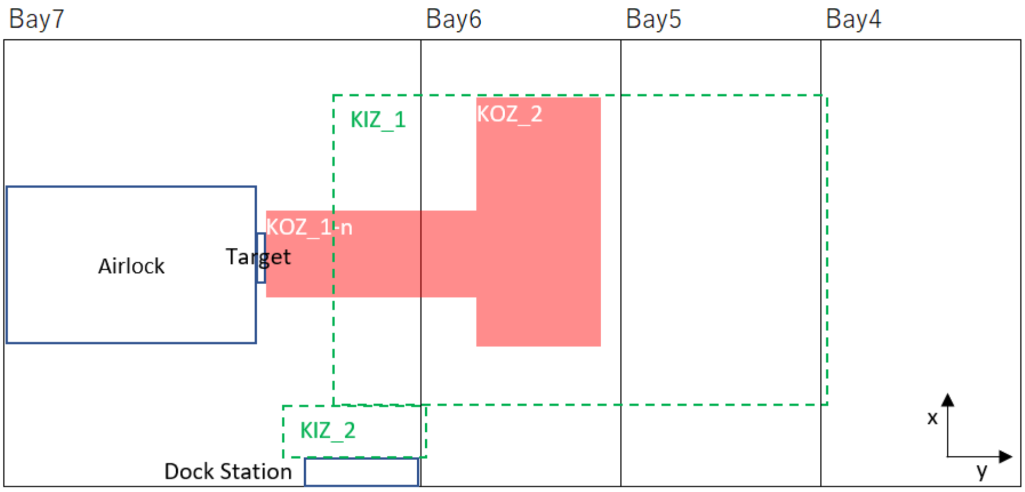


Figure 2.2.4-1 KIZ and KOZ of the Preliminary Round (Top View)

Table 2.2.4-1 shows the coordinates of KOZ and KIZ. Definition of the coordinates (x_min, y_min, z_min) and (x_max, y_maxm z_max) are shown in Figure 2.2.4-3. Also, the numbers of KOZ 1-n correspond to the numbers in Figure 2.2.4-4.

Table 2.2.4-1 Installed Coordinate of Obstacles

Type	No.		x_min	y_min	z_min	x_max	y_max	z_max
KOZ	1	1	x_{target} -0.3	y_{target}	z_{target} -0.3	x_{target} -0.075	y_{target} +1.785	z_{target} -0.075
		2	x_{target} -0.075	y_{target}	z_{target} -0.3	x_{target}	y_{target} +1.785	z_{target} -0.075
		3	x_{target}	y_{target}	z_{target} -0.3	x_{target} +0.075	y_{target} +1.785	z_{target} -0.075
		4	x_{target} +0.075	y_{target}	z_{target} -0.3	x_{target} +0.3	y_{target} +1.785	z_{target} -0.075
		5	x_{target} -0.3	y_{target}	z_{target} -0.075	x_{target} -0.075	y_{target} +1.785	z_{target}
		6	x_{target} -0.075	y_{target}	z_{target} -0.075	x_{target}	y_{target} +1.785	z_{target}
		7	x_{target}	y_{target}	z_{target} -0.075	x_{target} +0.075	y_{target} +1.785	z_{target}
		8	x_{target} +0.075	y_{target}	z_{target} -0.075	x_{target} +0.3	y_{target} +1.785	z_{target}
		9	x_{target} -0.3	y_{target}	z_{target}	x_{target} -0.075	y_{target} +1.785	z_{target} +0.075
		10	x_{target} -0.075	y_{target}	z_{target}	x_{target}	y_{target} +1.785	z_{target} +0.075
		11	x_{target}	y_{target}	z_{target}	x_{target} +0.075	y_{target} +1.785	z_{target} +0.075
		12	x_{target} +0.075	y_{target}	z_{target}	x_{target} +0.3	y_{target} +1.785	z_{target} +0.075
		13	x_{target} -0.3	y_{target}	z_{target} +0.075	x_{target} -0.075	y_{target} +1.785	z_{target} +0.3
		14	x_{target} -0.075	y_{target}	z_{target} +0.075	x_{target}	y_{target} +1.785	z_{target} +0.3
		15	x_{target}	y_{target}	z_{target} +0.075	x_{target} +0.075	y_{target} +1.785	z_{target} +0.3
		16	x_{target} +0.075	y_{target}	z_{target} +0.075	x_{target} +0.3	y_{target} +1.785	z_{target} +0.3
	2		10.71	-8.8	4.32	11.55	-8.5	5.57
KIZ	1		10.3	-10.2	4.32	11.55	-7.4	5.57
	2		9.5	-10.5	4.02	10.5	-9.6	4.8

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

*(x_{target}, y_{target}, z_{target}) is the coordinates of the center of the Target.

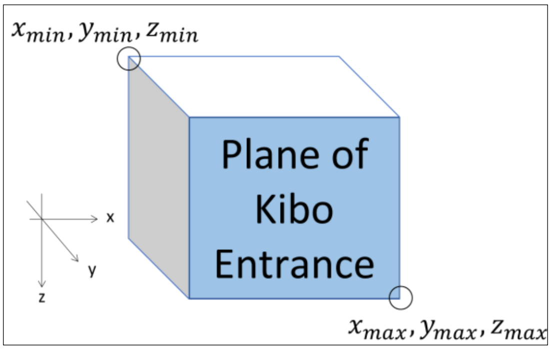


Figure 2.2.4-3 Definition of the coordinates

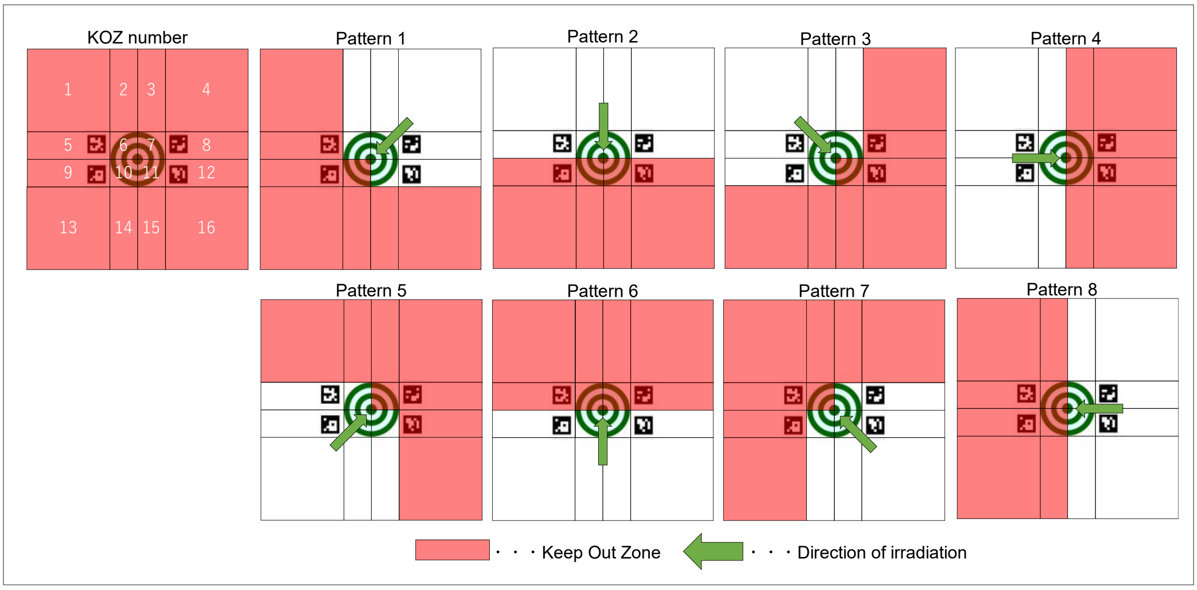


Figure 2.2.4-4 8 patterns of KOZ in front of target

2.2.5. 8 automatic runs per APK

In the Preliminary Round, 8 runs are automatically executed for one APK to keep the fairness against the random elements included in the simulation. In these 8 runs, the conditions such as the pattern of KOZ, the position of the target and other random elements are all different.

This method avoids the accidental results (good and bad results by happening) and ranked by the environmental and disturbance conditions. All participants can challenge in the same condition.

Note: You can select a certain pattern out of 8 KOZ directions in JAXA's simulator environment during Preliminary Trial so that you can work on each direction. The function will be released in the end of April.

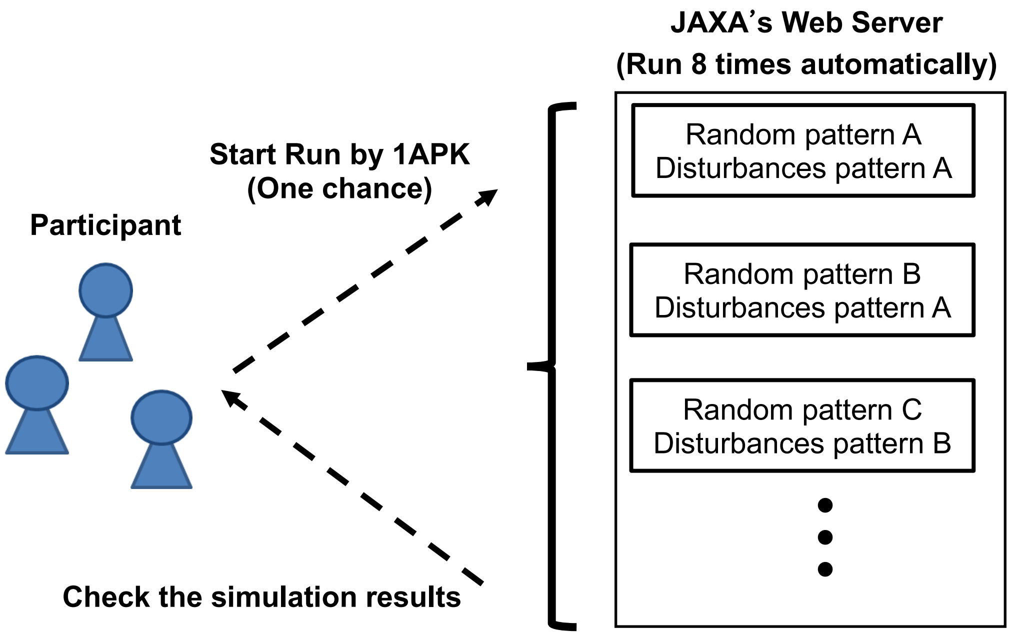


Figure 2.2.5-1 8 times run

2.2.6. Ranked by the worst result of 8 automatic runs

The ranking will be determined by the worst score in 8 runs.

Space missions are performed under very severe conditions, such that even one failure is not allowed to make, and that a malfunction cannot be recovered directly by human hands. This mission scenario is also an important mission that cannot be failed to help the astronaut's life at the ISS crisis, so it is necessary to complete the mission even with the worst result. Therefore, in order to evaluate programming that can perform well under any conditions, the ranking uses the worst result.

Detailed scoring criteria are provided in Section 2.3.2

2.3. Scoring

2.3.1. Factors

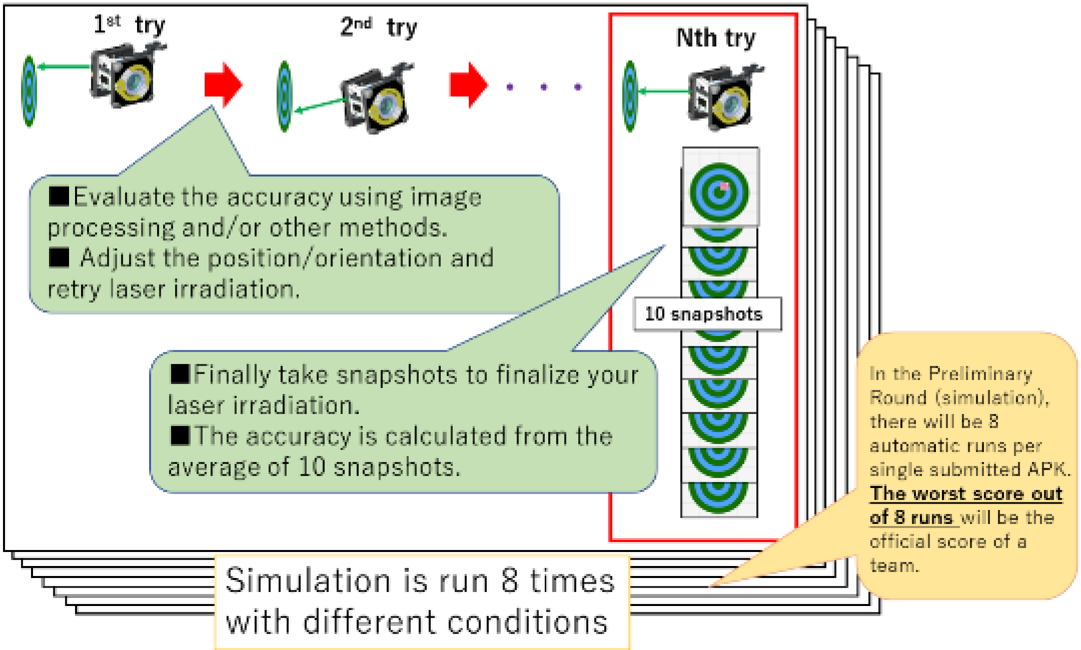
Your class and score are calculated from the following factors.

Table 2.3.1-1 Scoring Factors of the Preliminary Round

#	Factor		Detail
1	Reading QR code		Checking a record of read QR code.
2	Laser Accuracy	Distance from the center of the target to a point irradiated with laser pointer	Calculating a mean distance in 10 snapshots. The snapshots are taken when calling snapshot API after a laser irradiation.
4	Reporting "Mission Complete" to Astronaut (Issuing a JAXA prepared API)	Elapsed time from start to report	–

* Taking snapshots

- During irradiating the target with the laser, take 10 snapshots at intervals of a second. Score is evaluated by the average of distance from irradiating point to the center of target of 10 snapshots. Taking snapshots is allowed only once in each run.
- If you program automatic re-trial for fine adjustment of Astrobee position, you can retry aiming and laser irradiation until taking the snapshots to finalize the laser irradiation. (Refer to Figure 2.3.1-1) It might be better to consider a unique strategy such as using image processing to assess the accuracy.



■Evaluate the accuracy using image processing and/or other methods.
■ Adjust the position/orientation and retry laser irradiation.

■Finally take snapshots to finalize your laser irradiation.
■The accuracy is calculated from the average of 10 snapshots.

In the Preliminary Round (simulation), there will be 8 automatic runs per single submitted APK. **The worst score out of 8 runs** will be the official score of a team.

Simulation is run 8 times with different conditions

Figure 2.3.1-1 Method of repeating 8 times and scoring

2.3.2. Ranking method

The results are categorized according to the degree of achievement of the mission (Table 2.3.2-1) with the worst run score, and a team which marked the higher score in the same class ranks high.

Rank is given in order of class A > B > C > D (Table 2.3.2-2).

Table 2.3.2-1 Classification by achievement

Class Check Point	A	B	C	D
Found QR code	Yes	Yes	Yes	No
Illuminated the target	Yes	Yes	No	-
Report "Mission Complete"	Yes	No	-	-

A：All missions are achieved. Score range is from 0 point to 100 point and it is calculated from accuracy of laser pointing and elapsed time. The laser accuracy and elapsed time are weighted, respectively. The laser accuracy is more valued than elapsed time.

B：Found QR code and irradiate the target but the report to astronaut is failed. It is ranked in the order of accuracy of the irradiation.

C：Found QR code. It is ranked in the order of time until reading QR code.

D：Never done anything. It is no score.

Table 2.3.2-2 Ranking and Class relation

Rank	Team Name	Class	Score
1	Team C	A	90
2	Team E	A	60
3	Team A	B	70
4	Team D	B	60
5	Team G	C	2:10
6	Team F	C	3:45
7	Team B	C	4:10

※In the Preliminary Round, all teams are ranked and classified by the worst run results out of 8 runs. Therefore, even if 7 runs have completed the mission to the end, it is ranked by the worst run result when do not reach to the end within 5 minutes even once. In other words, the result of the runs that did not reach the end is the team's evaluation.

It is important to create a program that can accomplish missions under any random conditions.

※In the Preliminary Trial, only one run is simulated at once.

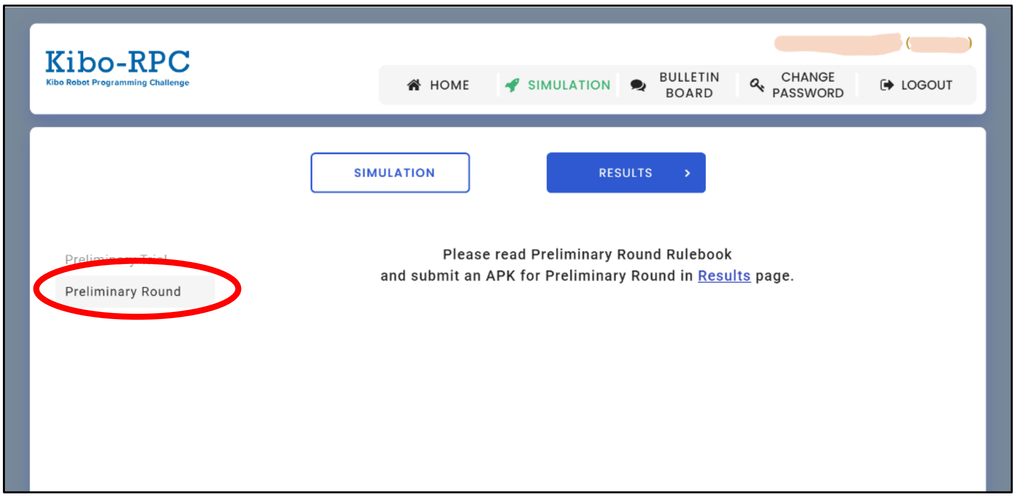
2.4. Joining Preliminary Round

2.4.1. How to participate in Preliminary Round

Participants need to submit the APK for the Preliminary Round by the submission deadline. (All figures below are screen shots in 1st Kibo-RPC.)

2.4.2. Submission of Preliminary Round APK

Participants create and submit a program for Preliminary Round by the submission deadline to participate in the Preliminary Round.

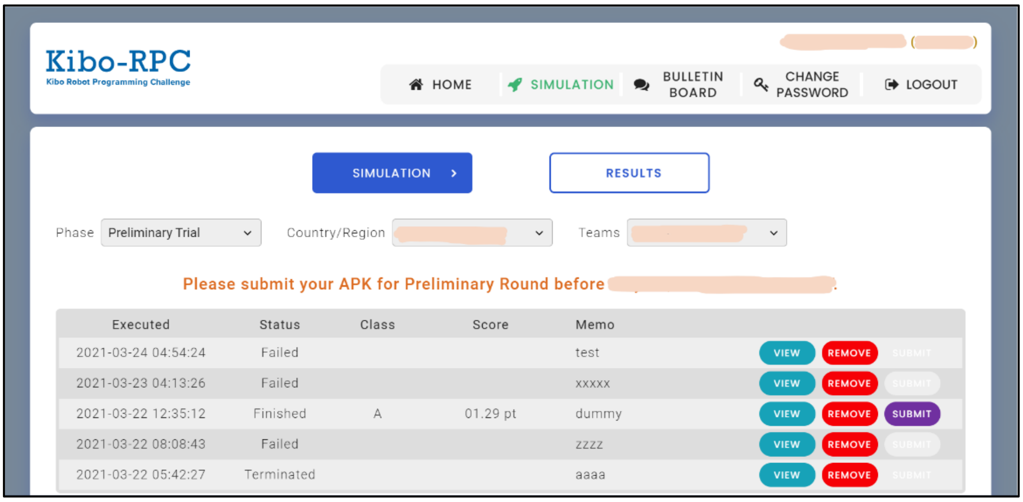


Please read Preliminary Round Rulebook
and submit an APK for Preliminary Round in Results page.

Figure 2.4.2-1 Preliminary Round

Before the Preliminary Round period, the web simulation result screen will be changed as shown in Figure 2.4.2-1. Before submitting your APK, please evaluate the performance of your program using Preliminary Trial simulator. After that, select your best program from the result list of “Preliminary Trial” on “RESULTS” screen, and press the "SUBMIT" button that is newly added and then the APK is submitted. Once press the “SUBMIT” button, it is changed to the “CANCEL" button. Therefore, you can resend the other APK by clicking the “CANCEL” button before the deadline.

Note: “SUBMIT” button can only be pressed when you subumit the APK that is “Finished” status in the “Preliminary trial”. If you have only “Failed” APK, you cannot participate in the Preliminary Round. Please make sure to create an APK that is “Finished” in the “Preliminary trial”.



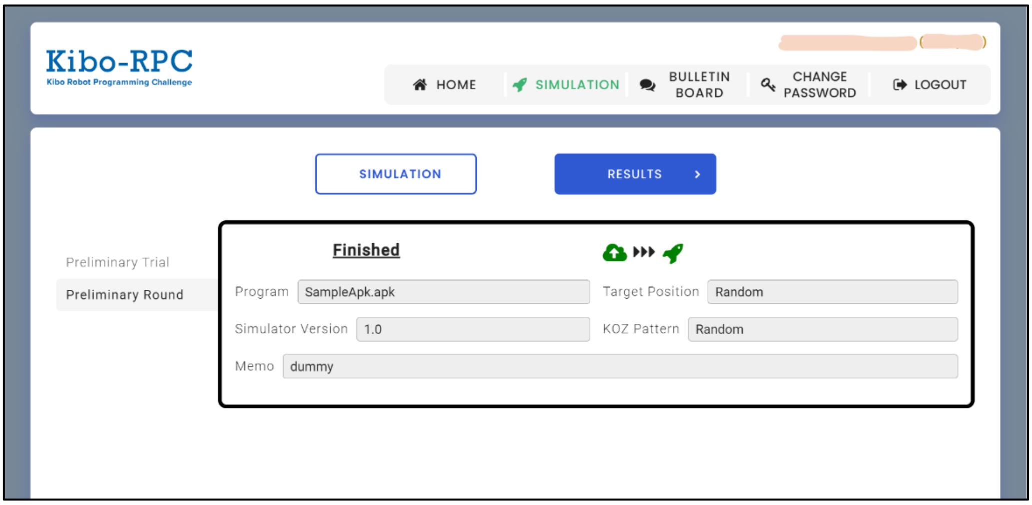
Phase: Preliminary Trial
Country/Region: [redacted]
Teams: [redacted]

Please submit your APK for Preliminary Round before [redacted].

Executed	Status	Class	Score	Memo	
2021-03-24 04:54:24	Failed			test	VIEW REMOVE SUBMIT
2021-03-23 04:13:26	Failed			xxxxx	VIEW REMOVE SUBMIT
2021-03-22 12:35:12	Finished	A	01.29 pt	dummy	VIEW REMOVE SUBMIT
2021-03-22 08:08:43	Failed			zzzz	VIEW REMOVE SUBMIT
2021-03-22 05:42:27	Terminated			aaaa	VIEW REMOVE SUBMIT

Figure 2.4.2-2 Preliminary Trial after change

The submitted APK can be confirmed from the "Preliminary Round" tab as shown in Figure 2.4.2-3. You can cancel the APK on RESULTS screen, but be careful about the submission deadline because you cannot press the “SUBMIT” and “CANCEL” button over the deadline



Finished

Program: SampleApk.apk
Target Position: Random
Simulator Version: 1.0
KOZ Pattern: Random
Memo: dummy

Figure 2.4.2-3 Preliminary Round

2.5. Event methodology

This section explains three typical Preliminary Round styles which are carried out in each country/region. Please confirm how Preliminary Round event will be held at the ACTIVITY tab on the Kibo-RPC official web site or contact your country's/region's POC.

2.5.1. Real Event

Real Event means the Preliminary Round event is held at a venue in your country /region. Participants must submit the APK on the web by the deadline and go to the venue on the day. If it is difficult to go to the venue, each team's leader should talk to your country's/region's POC and participate in an alternative way (typically through Social Media). Figure 2.5.1-1 shows the general flow.

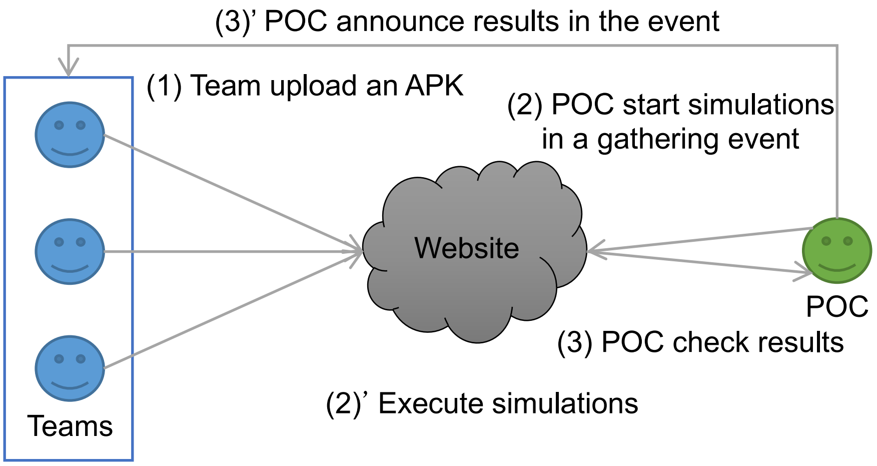

(3)' POC announce results in the event
(1) Team upload an APK
(2) POC start simulations in a gathering event
Website
POC
(3) POC check results
(2)' Execute simulations
Teams

Figure 2.5.1-1 Real Event flow

2.5.2. Virtual Event

Virtual Event means the participants take part in the Preliminary Round through Social Media. Therefore, participants can participate from your school and home. The basic Preliminary Round procedure is the same as in section 2.5.1. Participants must submit the APK on the web by the deadline.

2.5.3. E-mail Notification

E-mail Notification is a method that done without gathering with other participants, and the result is notified by e-mail from your country's/region's POC later. Therefore, participants do not need to participate in the Preliminary Round on a particular day. However, as in 2.5.1 and 2.5.2, participants must submit the APK on the web by the deadline.

*Please confirm with the POC in each country/region which style they will carry out.

3. Programming Skills Round

3.1. Submission for Programming Skills Round

Only the winner of the Preliminary Round in each country/region can participate in the Programming Skills Round hosted by JAXA. The winner of the Programming Skills Round can advance to the Final Round. Since the program submitted for the Preliminary Round will be used also for the Programming Skills Round, the deadline also will be on the same day with the Preliminary Round.

3.2. Game Rules

Game rules and Scoring factor are the same as the Preliminary Round. However, gaming factor like randomness of the airflow and a position of target might be changed.

3.3. Result

The Programming Skills Round will be hosted by JAXA. The event methodology and date will be announced.

4. Final Round

4.1. Preparation for Final Round

Only the winner of the Programming Skills Round can participate in the Final Round. In the Final Round, the participants will be able to refine your program from the Preliminary/ Programming Skills Round. Please create a program for the finals and submit APK and source code by the submission deadline. Refer to "Submission of Final Round APK" in section 4.4.

Draft Program Submission Deadline: TBD

4.2. Game Rules

4.2.1. Game Flow

In the Final Round, each team creates a program that moves Astrobee on ISS from the Dock Station to the target point and to Astronaut with avoiding KOZ and illuminates the target with the laser. Basically, the same game flow with the Preliminary Round, but please note that “Mission Complete” report is a little different from the Preliminary Round. “Mission Complete” report is done by playing a pre-recorded audio file submitted by the finalist teams. When the “Mission Complete” API prepared by JAXA is issued by APK, it will automatically play the audio file. See the Programming Manual for detailed information on audio files.

Figure 4.2.1-2 The 1st Kibo-RPC Final Round

4.2.2. Preconditions

Coming soon

Since the environmental conditions are different from in simulation and on-orbit, it is important to create a program that can show high performance even in the real environment.

4.2.3. Mission Complete Report

The supported formats of the audio files are described at following URL;
https://developer.android.com/guide/topics/media/media-formats

You need to record your voice and implement it into your APK to report mission completion on the ISS. The detail of how to implement is written in Section XXX on the Programming Manual.

When the "reportMissionCompletion" API is called, the audio file will be played following sequence (Figure 4.2.3) after some action of Astrobee.

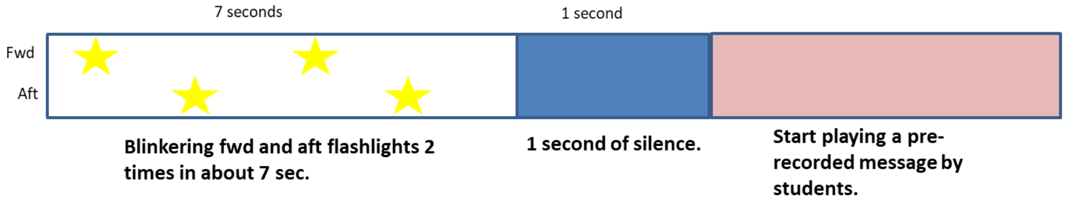

7 seconds
1 second
Fwd
Aft
Blinkering fwd and aft flashlights 2 times in about 7 sec.
1 second of silence.
Start playing a pre-recorded message by students.

Figure 4.2.3 Audio play sequence

4.2.4. Keep-In-Zone (KIZ) and Keep-Out-Zone (KOZ)

The same as the Preliminary Round.

4.2.5. Only one APK, only one run

Each team submits one APK like the preliminary round. However, it only runs one-time on ISS. Since the Final Round uses Astrobee on ISS, it cannot be redone or interrupted. It is a one-time shot.

4.2.6. The Time limit 5 minutes per team

If it exceeds 5 minutes, APK will automatically shut down. Please make sure that it completes the mission within 5 minutes. If Astrobee gets stuck or its self-position is lost, it automatically judges as a game-over.

4.2.7. Operation of the APK on the day of the Final Round

Participants may NOT operate the APK on the day of the Final Round.

The submitted APKs are code-reviewed by the technical team of JAXA / NASA and uplinked to Astrobee on ISS in advance. One team can run only once. The APKs are started with the execution command sent from ground operators.

4.2.8. Judging method

On the ISS, the speed and accuracy of the mission are judged by the following methods. Details of the scoring criteria are given in Sections 2.3.

Speed: Time from APK execution start (Undock from Dock Station) to send Finish command (Mission Complete Report) is recorded as the time stamp in Astrobee, which is available to the ground as telemetry. Finish time is the time called API of "Mission Complete".

Accuracy: Eleven snapshots (Ten are used for the judging and one is spare.) at the laser irradiation is recorded in Astrobee on ISS. The accuracy is judged by image analysis.

Besides, the level of mission achievement is judged in the same way as the Preliminary Round, using APIs (startMission, sendDiscovoredQR, reportMissionCompletion). See also the Programming Manual.

4.3. Scoring

4.3.1. Factors

The scoring factors are the same as the Preliminary Round. However, in the final round, your APK will be run only once onboard.

4.3.2. Ranking Method

The same as the Preliminary Round. However, your voice message of "Mission Complete" will be evaluated for Special award as well.

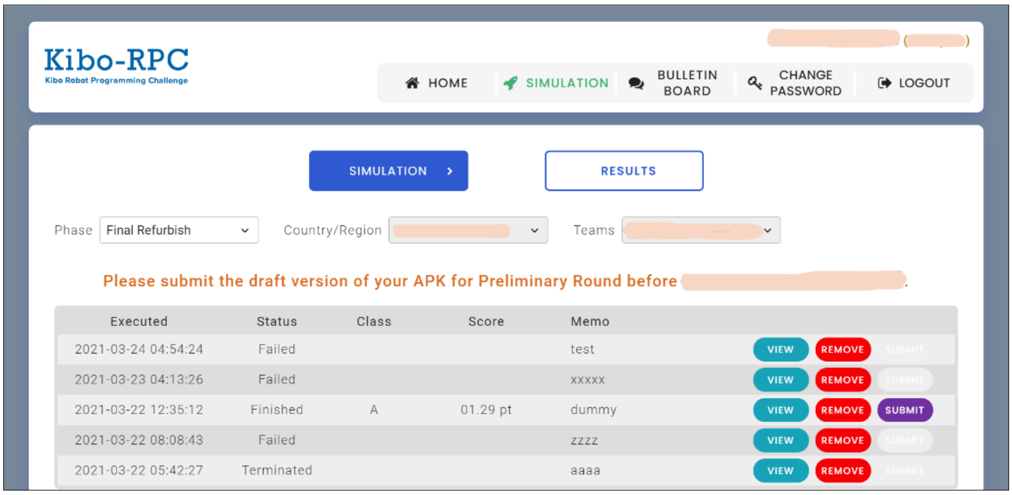
4.4. Joining the Final Round

4.4.1. Submission of APK and Source Code

The participants need to create and submit a program for Final Round in advance by the deadline. In addition, the source code shall be confirmed with JAXA and NASA in advance for safety reasons since we use the Astrobee on-orbit for the Final Round. Therefore, you must submit (1) APK and (2) source code.

(1). How to Submit APK (All figures are screen shot in 1st Kibo-RPC.)

You can select one APK and press submit button on “RESULTS” display. Only class D or better, i.e., those that are guaranteed to have been created in accordance with appropriate procedures, can be submitted. (If you have properly adjusted the parameters and API for Astrobee actual machine as section 5.1(5), the simulation result may not be class A.)



Phase: Final Refurbish
Country/Region: [redacted]
Teams: [redacted]

Please submit the draft version of your APK for Preliminary Round before [redacted].

Executed	Status	Class	Score	Memo
2021-03-24 04:54:24	Failed			test
2021-03-23 04:13:26	Failed			xxxxx
2021-03-22 12:35:12	Finished	A	01.29 pt	dummy
2021-03-22 08:08:43	Failed			zzzz
2021-03-22 05:42:27	Terminated			aaaa

Fig.4.4.1-1 How to Submit APK

(2). How to Submit Source Code

You must submit your source code via e-mail. Follow the instructions below.

1. **Generate MD5 of APK file**

   Kibo-RPC secretariat confirms APK submitted via website and this md5.

   A) on Windows

   Execute following commands in 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) on Ubuntu

   Execute following commands in 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 file/directories**

   Delete following file. (Be sure not to delete md5)

   - [root dir]/app/build/outputs/apk/app-debug.apk

   Delete 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.) root directory and send the compressed file to secretariat**

   Compressed file should be a few hundred KB or a few MB. Confirm that all files, especially Java source files and md5, are included in the compressed file.

   Please send it to Z-KRPC@ml.jaxa.jp

4.5. Event methodology

This section explains the plans of the Final Round.

4.5.1. Real Event

Real Event means the Final Round event is held at a venue in Japan. Participants must submit the APK on the web by the deadline and go to the venue on the day. The final round will be streamed live for participants who cannot travel to Japan. Figure 4.5.1-1 shows the general flow.

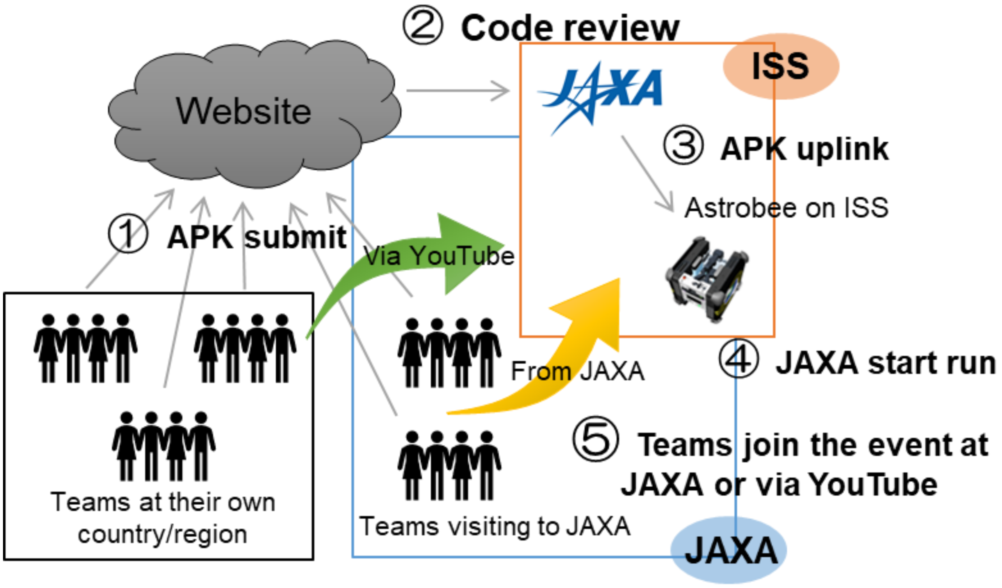


Figure 4.5.1-1 Real Event flow