National Aeronautics and Space Administration

APP DEVELOPMENT CHALLENGE

Ascent Abort 2

EXPLORE MOON to MARS

NEXT GEN STEM

www.nasa.gov

Contents

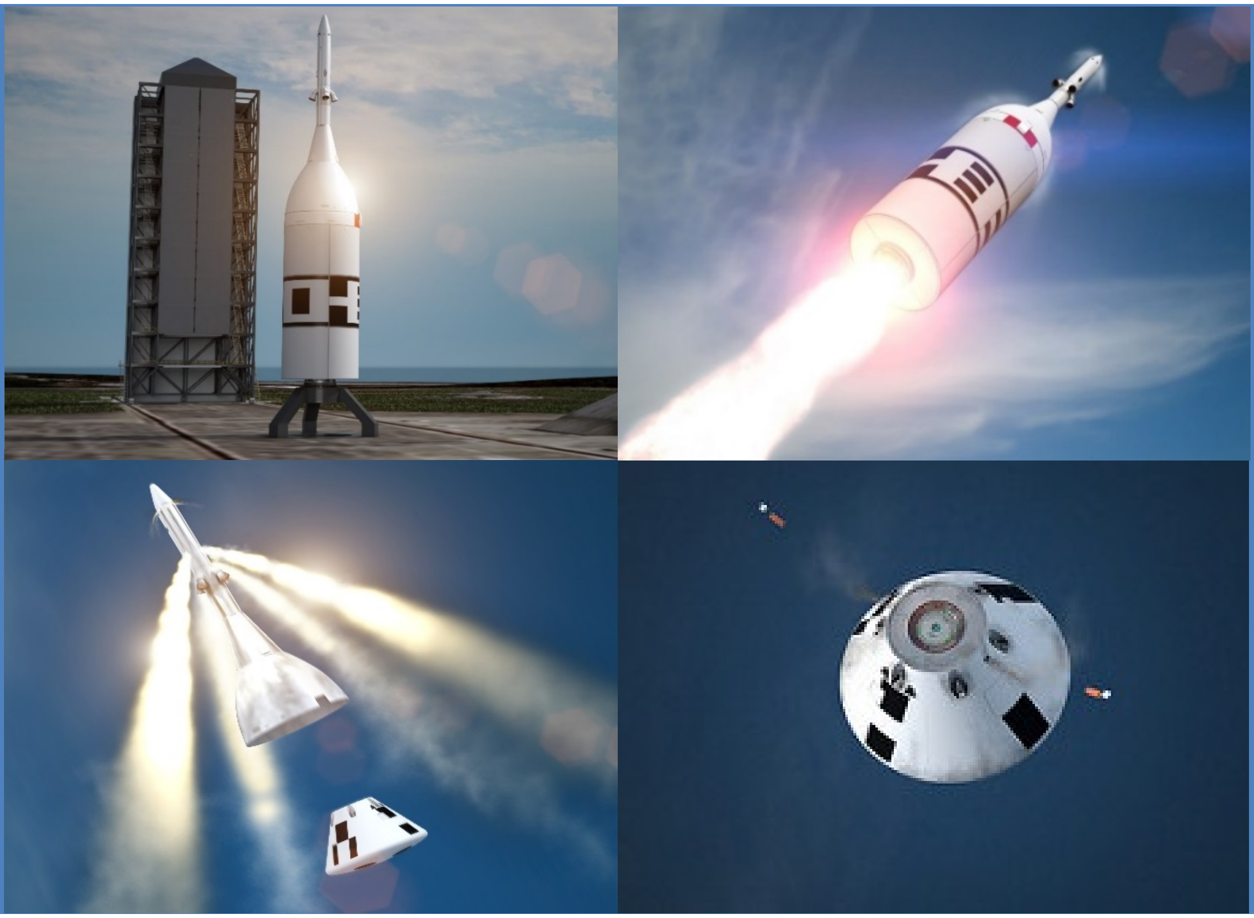
Introduction

Attention middle or high school students: it's time to form a team to compete in this unique opportunity.

NASA needs YOU! Are you someone who likes to code, create apps, or develop amazing visuals and graphics? Are you someone who is interested in coding or computer science, but just haven't had a chance to learn more? If so, join our App Development Challenge (ADC).

The App Development Challenge Guide provides all the details needed for each team to participate in developing an app in support of NASA's Ascent Abort 2 (AA-2) launch currently scheduled for no earlier than June, 2019.

Note: Information about the challenge will remain posted for use in educational settings beyond the challenge end date.



Getting Started

This guide provides challenge components, directions, timelines, handouts, and resources to help teams create an app, including extensive details for each aspect of the challenge, as well as links to important resources. The content found in the links are supplemental materials to this ADC Guide.

LET'S GET STARTED!

Lead teachers and team members should read the entire guide to understand what handouts, directions, coding components, and expectations are required as part of the challenge. Also, look through the "Mission Resources" section near the end of this guide. These resources link to important fact sheets, program overviews, and videos. New to programming? Check out NASA Computer Science Educational Resources here.

Computer Science Teacher Association (CSTA) K-12 Standards Addressed by the ADC

- 2-DA-08. Collect data using computational tools and transform the data to make it more useful and reliable.
- 2-AP-10. Use flowcharts and/or pseudocode to address complex problems as algorithms.
- 2-AP-11. Create clearly named variables that represent different data types and perform operations on their values.
- 2-AP-13. Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.
- 2-AP-15. Seek and incorporate feedback from team members and users to refine a solution that meets user needs.
- 2-AP-18. Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.
- 3A-CS-02. Compare levels of abstraction and interactions between application software, system software, and hardware layers.
- 3A-DA-11. Create interactive data visualizations using software tools to help others better understand real-world phenomena.
- 3A-DA-12. Create computational models that represent the relationships among different elements of data collected from a phenomenon or process.
- 3A-AP-14. Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.
- 3B-AP-24. Compare multiple programming languages and discuss how their features make them suitable for solving different types of problems.

Curriculum Standards Alignment:

6^{th} – 12^{th} Grade Next Generation Science Standards

Next Generation Science Standards: (https://www.nextgenscience.org/search-standards)

Code	NGSS Performance Expectation:	Common Core State Standard Connections:
MS-ETS 1-1	Define the criteria and constraints of a design problem with sufficient precision to ensure a successful solution, taking into account relevant scientific principles and potential impacts on people and the natural environment that may limit possible solutions.	**ELA/Literacy –** RST.6-8.1 Cite specific textual evidence to support analysis of science and technical texts. (MS-ETS1-1),(MS-ETS1-2),(MS-ETS1-3) RST.6-8.7 Integrate quantitative or technical information expressed in words in a text with a version of that information expressed visually (e.g., in a flowchart, diagram, model, graph, or table). (MS-ETS1-3) RST.6-8.9 Compare and contrast the information gained from experiments, simulations, video, or multimedia sources with that gained from reading a text on the same topic. (MS-ETS1-2),(MS-ETS1-3) WHST.6-8.8 Gather relevant information from multiple print and digital sources, using search terms effectively; assess the credibility and accuracy of each source; and quote or paraphrase the data and conclusions of others while avoiding plagiarism and following a standard format for citation. (MS-ETS1-1) WHST.6-8.9 Draw evidence from informational texts to support analysis, reflection, and research. (MS-ETS1-2) SL.8.5 Integrate multimedia and visual displays into presentations to clarify information, strengthen claims and evidence, and add interest. (MS-ETS1-4) **Mathematics –** MP.2 Reason abstractly and quantitatively. (MS-ETS1-1),(MS-ETS1-2),(MS-ETS1-3),(MS-ETS1-4) 7.EE.3 Solve multi-step real-life and mathematical problems posed with positive and negative rational numbers in any form (whole numbers, fractions, and decimals), using tools strategically. Apply properties of operations to calculate with numbers in any form; convert between forms as appropriate; and assess the reasonableness of answers using mental computation and estimation strategies. (MS-ETS1-1),(MS-ETS1-2),(MS-ETS1-3)
MS-ETS 1-2	Evaluate competing design solutions using a systematic process to determine how well they meet the criteria and constraints of the problem.	
MS-ETS 1-3	Analyze data from tests to determine similarities and differences among several design solutions to identify the best characteristics of each that can be combined into a new solution to better meet the criteria for success.	
MS-ETS 1-4	Develop a model to generate data for iterative testing and modification of a proposed object, tool, or process such that an optimal design can be achieved.	
HS ETS 1-1	Analyze a major global challenge to specify qualitative and quantitative criteria and constraints for solutions that account for societal needs and wants.	**ELA/Literacy –** RST.11-12.7 Integrate and evaluate multiple sources of information presented in diverse formats and media (e.g., quantitative data, video, multimedia) in order to address a question or solve a problem. (HS-ETS1-1),(HS-ETS1-3) RST.11-12.8 Evaluate the hypotheses, data, analysis, and conclusions in a science or technical text, verifying the data when possible and corroborating or challenging conclusions with other sources of information. (HS-ETS1-1),(HS-ETS1-3) RST.11-12.9 Synthesize information from a range of sources (e.g., texts, experiments, simulations) into a coherent understanding of a process, phenomenon, or concept, resolving conflicting information when possible. (HS-ETS1-1),(HS-ETS1-3) **Mathematics –** MP.2 Reason abstractly and quantitatively. (HS-ETS1-1),(HS-ETS1-3),(HS-ETS1-4) MP.4 Model with mathematics. (HS-ETS1-1),(HS-ETS1-2),(HS-ETS1-3),(HS-ETS1-4)
HS-ETS 1-2	Design a solution to a complex real-world problem by breaking it down into smaller, more manageable problems that can be solved through engineering.	
HS-ETS 1-3	Evaluate a solution to a complex real-world problem based on prioritized criteria and trade-offs that account for a range of constraints, including cost, safety, reliability, and aesthetics, as well as possible social, cultural, and environmental impacts.	
HS-ETS 1-4	Use a computer simulation to model the impact of proposed solutions to a complex real-world problem with numerous criteria and constraints on interactions within and between systems relevant to the problem.	

6th – 8th Grade Computer Science Standards

K-12 CS Framework: (https://k12cs.org/framework-statements-by-grade-band/)
CSTA Standards: (https://www.csteachers.org/page/standards)

Subconcept	K-12 CS Framework Statements:	CSTA Standards:
Hardware and Software	Hardware and software determine a computing system's capability to store and process information. The design or selection of a computing system involves multiple considerations and potential tradeoffs, such as functionality, cost, size, speed, accessibility, and aesthetics.	Design projects that combine hardware and software components to collect and exchange data.
Troubleshooting	Comprehensive troubleshooting requires knowledge of how computing devices and components work and interact. A systematic process will identify the source of a problem, whether within a device or in a larger system of connected devices.	Systematically identify and fix problems with computing devices and their components.
Storage	Applications store data as a representation. Representations occur at multiple levels, from the arrangement of information into organized formats (such as tables in software) to the physical storage of bits. The software tools used to access information translate the low-level representation of bits into a form understandable by people.	Represent data using multiple encoding schemes.
Inference and Models	Computer models can be used to simulate events, examine theories and inferences, or make predictions with either few or millions of data points. Computer models are abstractions that represent phenomena and use data and algorithms to emphasize key features and relationships within a system. As more data is automatically collected, models can be refined.	Refine computational models based on the data they have generated.
Collection, Visualization and Transformation	People design algorithms and tools to automate the collection of data by computers. When data collection is automated, data is sampled and converted into a form that a computer can process. For example, data from an analog sensor must be converted into a digital form. The method used to automate data collection is influenced by the availability of tools and the intended use of the data. Data can be transformed to remove errors, highlight or expose relationships, and/or make it easier for computers to process.	Collect data using computational tools and transform the data to make it more useful and reliable.
Algorithms	Algorithms affect how people interact with computers and the way computers respond. People design algorithms that are generalizable to many situations. Algorithms that are readable are easier to follow, test, and debug.	Use flowcharts and/or pseudocode to address complex problems as algorithms.
Variables	Programmers create variables to store data values of selected types. A meaningful identifier is assigned to each variable to access and perform operations on the value by name. Variables enable the flexibility to represent different situations, process different sets of data, and produce varying outputs.	Create clearly named variables that represent different data types and perform operations on their values.
Control	Programmers select and combine control structures, such as loops, event handlers, and conditionals, to create more complex program behavior.	Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.
Modularity	Programs use procedures to organize code, hide implementation details, and make code easier to reuse. Procedures can be repurposed in new programs. Defining parameters for procedures can generalize behavior and increase reusability.	Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. Create procedures with parameters to organize code and make it easier to reuse.
Program Development	People design meaningful solutions for others by defining a problem's criteria and constraints, carefully considering the diverse needs and wants of the community, and testing whether criteria and constraints were met.	Seek and incorporate feedback from team members and users to refine a solution that meets user needs. Incorporate existing code, media, and libraries into original programs, and give attribution. Systematically test and refine programs using a range of test cases. Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts. Document programs in order to make them easier to follow, test, and debug.

9^{th} – 12^{th} Grade Computer Science Standards

K-12 CS Framework: (https://k12cs.org/framework-statements-by-grade-band/)

CSTA Standards: (https://www.csteachers.org/page/standards)

Subconcept	K-12 CS Framework Statements:	CSTA Standards: 9th-12th grade (combined)
Hardware and Software	Levels of interaction exist between the hardware, software, and user of a computing system. The most common levels of software that a user interacts with include system software and applications. System software controls the flow of information between hardware components used for input, output, storage, and processing.	Compare levels of abstraction and interactions between application software, system software, and hardware layers.
Troubleshooting	Troubleshooting complex problems involves the use of multiple sources when researching, evaluating, and implementing potential solutions. Troubleshooting also relies on experience, such as when people recognize that a problem is similar to one they have seen before or adapt solutions that have worked in the past.	Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors. Illustrate ways computing systems implement logic, input, and output through hardware components.
Storage	Data can be composed of multiple data elements that relate to one another. For example, population data may contain information about age, gender, and height. People make choices about how data elements are organized and where data is stored. These choices affect cost, speed, reliability, accessibility, privacy, and integrity.	Translate between different bit representations of real-world phenomena, such as characters, numbers, and images. Evaluate the tradeoffs in how data elements are organized and where data is stored.
Inference and Models	The accuracy of predictions or inferences depends upon the limitations of the computer model and the data the model is built upon. The amount, quality, and diversity of data and the features chosen can affect the quality of a model and ability to understand a system. Predictions or inferences are tested to validate models.	Create computational models that represent the relationships among different elements of data collected from a phenomenon or process. Evaluate the ability of models and simulations to test and support the refinement of hypotheses.
Collection, Visualization and Transformation	Data is constantly collected or generated through automated processes that are not always evident, raising privacy concerns. The different collection methods and tools that are used influence the amount and quality of the data that is observed and recorded. People transform, generalize, simplify, and present large data sets in different ways to influence how other people interpret and understand the underlying information. Examples include visualization, aggregation, rearrangement, and application of mathematical operations.	Create interactive data visualizations using software tools to help others better understand real-world phenomena. Use data analysis tools and techniques to identify patterns in data representing complex systems. Select data collection tools and techniques to generate data sets that support a claim or communicate information.
Algorithms	People evaluate and select algorithms based on performance, reusability, and ease of implementation. Knowledge of common algorithms improves how people develop software, secure data, and store information.	Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests. Use and adapt classic algorithms to solve computational problems.
Variables	Data structures are used to manage program complexity. Programmers choose data structures based on functionality, storage, and performance tradeoffs.	Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables. Compare and contrast fundamental data structures and their uses.
Control	Programmers consider tradeoffs related to implementation, readability, and program performance when selecting and combining control structures.	Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance, and explain the benefits and drawbacks of choices made.
Modularity	Complex programs are designed as systems of interacting modules, each with a specific role, coordinating for a common overall purpose. These modules can be procedures within a program; combinations of data and procedures; or independent, but interrelated, programs. Modules allow for better management of complex tasks.	Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects. Construct solutions to problems using student-created components, such as procedures, modules and/or objects. Analyze a large-scale computational problem and identify generalizable patterns that can be applied to a solution.
Program Development	Diverse teams can develop programs with a broad impact through careful review and by drawing on the strengths of members in different roles. Design decisions often involve tradeoffs. The development of complex programs is aided by resources such as libraries and tools to edit and manage parts of the program. Systematic analysis is critical for identifying the effects of lingering bugs.	Evaluate and refine computational artifacts to make them more usable and accessible. Design and develop computational artifacts working in team roles using collaborative tools. Develop programs for multiple computing platforms. Use version control systems, integrated development environments (IDEs), and collaborative tools and practices (code documentation) in a group software project. Develop and use a series of test cases to verify that a program performs according to its design specifications. Compare multiple programming languages and discuss how their features make them suitable for solving different types of problems.

Moon to Mars

NASA will lead an innovative and sustainable program of exploration with commercial and international partners to enable human expansion across the solar system and to bring back to Earth new knowledge and opportunities. Working with U.S. companies and international partners, NASA will push the boundaries of human exploration forward to the Moon and on to Mars. NASA is working to establish a permanent human presence on the Moon within the next decade to uncover new scientific discoveries and lay the foundation for private companies to build a lunar economy.

Right now, NASA is taking steps to begin this next era of exploration.

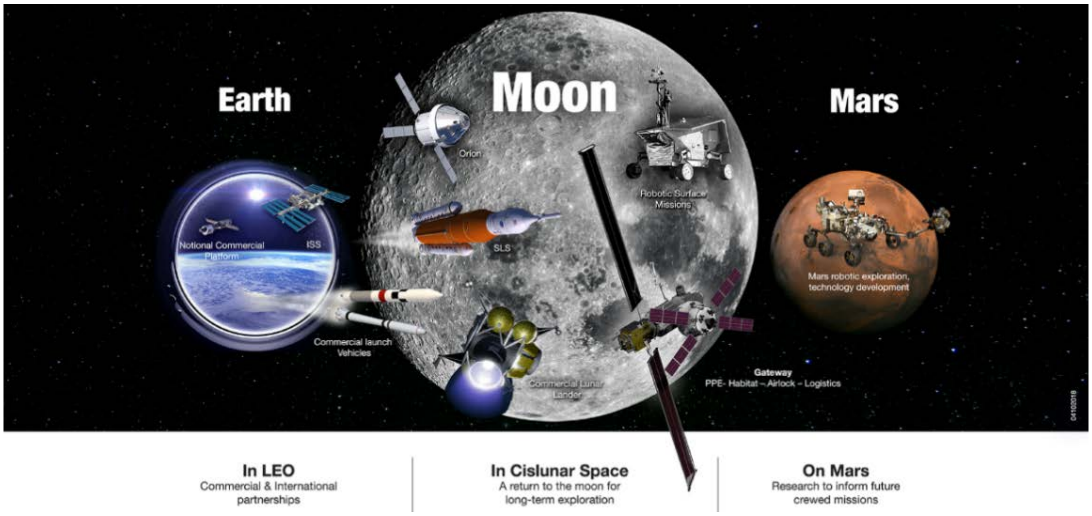
It all starts with delivery services to the lunar surface from U.S. companies for scientific instruments and technology demonstrations as well as a spaceship, called the Gateway, in orbit around the Moon that will support human missions to the surface with reusable lander elements for decades to come. The Gateway will, for the first time, give NASA and its partner's access to more of the lunar surface than ever before, supporting both human and robotic missions. The Agency's powerful Space Launch System (SLS) rocket and Orion spacecraft will be the backbone to build the Gateway and transport astronauts to and from Earth.

Getting There: SLS and Orion

SLS and Orion are critical to the NASA's exploration plans at the Moon and beyond. NASA designed the Space Launch System as the world's most powerful rocket for safely sending humans on missions to deep space, and Orion is specifically designed to keep humans alive hundreds of thousands of miles from home, where getting back to Earth takes days rather than hours. Together, the spacecraft and rocket will provide an entirely new capability for human exploration and be flexible for launching crew and cargo missions to enable exploration missions.

The first mission will test the new spacecraft systems flying together for the first time, without crew, and the second flight will take astronauts on a flight test around the Moon. Subsequent flights of Orion are planned send crew at least once a year to the Gateway and also deliver components to the lunar outpost. NASA is going back to the Moon to stay, in a measured, sustainable fashion while we keep our eyes on Mars. A sustainable crew presence in deep space will require the best of NASA, international partners and the private sector.

For more information please visit: https://www.nasa.gov/topics/moon-to-mars/overview



EXPLORE MOON to MARS

APP DEVELOPMENT CHALLENGE: ASCENT ABORT 2

NASA's App Development Challenge (ADC)

Orion will be NASA's safest spacecraft ever built to take humans to explore beyond the Moon and on to Mars, and that's by design. An integral part of ensuring safe spaceflight is Orion's Launch Abort System (LAS). This state-of-the-art crew escape system is attached to the top of the spacecraft and can propel the crew module away from the rocket within milliseconds should a life-threatening event arise during ascent.

In June 2019, Orion is scheduled to undergo a full-stress test of the LAS, called Ascent Abort Test 2 (AA-2), where a booster will launch from Cape Canaveral Air Force Station in Florida, carrying a fully functional LAS and a 22,000-pound Orion test vehicle to an altitude of 31,000 feet at Mach 1.3 (over 1,000 miles an hour). At that point, the LAS' powerful reverse-flow abort motor will fire 400,000 pounds of thrust, propelling the Orion test vehicle to a safe distance away from the rocket.

The test will provide engineers with critical abort test data to validate computer models of the spacecraft's LAS performance and system functions in advance of the first crewed flight of Orion and the SLS rocket. AA-2 provides the only opportunity to test a fully active launch abort system during ascent before flying crew, so verifying that it works as predicted in the event of an emergency is a critical step for deep space exploration.

Challenge Overview

During NASA's AA-2 Test of Orion, a closed Flight Control Room (FCR) will be established to manage the test flight from the point of launch until landing. Only individuals present in the FCR will be able to receive test flight data real-time.

The ADC will provide an opportunity for middle school and/or high school students to demonstrate the practice of coding and engineering design principles. In this ADC, students work in teams to develop an app that displays data from NASA's Ascent Abort-2 (AA-2) Test of Orion in a creative and meaningful way.

Student-generated solutions will provide the AA-2 team with a suite of options to choose from to visualize data and will help NASA personnel in remote locations collaborate with colleagues in the FCR at Johnson Space Center during the AA-2 test or for future missions.

Challenge Summary

In Round 1, teams will post their app designs online for consideration by NASA for use in future missions. In Round 2, teams with favorable submissions will advance to present their app in an interview with NASA engineers working on the AA-2 Launch Test. After this round, NASA will select student team(s) for an all-expenses paid trip to a NASA field center in early summer, 2019. The challenge will begin on March 13, 2019 and Round 1 participation will end with video submissions on May 1, 2019.

Note: Information about the challenge will remain posted for use in educational settings beyond the challenge end date.

Review the Challenge
Student teams must

- Be able to process approximately three minutes' worth of data smoothly.
- Use any programming language (e.g. Java, Scratch, etc.) and/or operating systems (Windows, Android, etc.) to complete development of an app.
- Read 21 double-precision floating point numbers (doubles) that represent position and rotation of the rocket's three main parts from an external source.
- Display all 21 doubles (not necessarily simultaneously) in some meaningful form other than text.
- Submit a video of original student-led work on the completed app.
- Adhere to each school district's policies regarding participation in the challenge.
- Complete program requirements as identified by the ADC team.

Review the Timeline

- March 13 Live Virtual Kick-Off Event and Presentation
- March 27 Live Virtual Connection-Game Engine Scripting for Object Motion
- April 3 Live Virtual Connection-Pseudo Coding for Quaternion
- April 10 Last Day to Enter the Challenge
- April 17 Live Virtual Connection-Human Factors for Visualization Application
- May 1 ADC Video Submission Deadline

Form a Team
All participants must be

- U.S. Citizens
- On a team of five to six middle or high school students and a lead teacher
- At least 13 years of age
- Led by a certified and practicing K-12 educator (i.e., lead teacher)

Notify NASA
After reviewing the above thoroughly, email any questions or team's intent to participate to NASA's ADC team at JSC-M2MSTEM@mail.nasa.gov.
Lead teachers must email a notification of intent to participate in the challenge that includes a letter of support from the school's administrator. Notifications of intent to participate must be received and confirmed no later than April 10, 2019.

Challenge Timeline

The following timeline will provide details on all major activities for the App Development Challenge.

The start date is Wednesday, March 13, 2019 and Video Submissions are due on Wednesday, May 1st, by 2pm, central time. This will conclude team's participation in the App Development Challenge unless selected to participate in Round 2. Teams that are not selected for Round 2 will be notified by email.

Teams are encouraged to participate in all live virtual connections to receive information about coding and app development. If there are any questions about this timeline, please contact the team at JSC-M2MSTEM@mail.nasa.gov.

Reminder, lead teachers must send an email to the email address listed above to confirm team's participation in the challenge.

Start Date	Due Date	Event
3/13/2019	5/1/2019	**NASA'S APP DEVELOPMENT CHALLENGE BEGINS, ROUND 1**
3/13/2019		Virtual Connection 1 WELCOME, CHALLENGE GUIDE Q/A & PRESENTATION
3/27/2019		Virtual Connection 2: GAME ENGINE SCRIPTING FOR OBJECT MOTION
4/3/2019		Virtual Connection 3 PSEUDO CODING FOR QUATERNION
	4/10/2019	**LAST DAY TO ENTER THE CHALLENGE**
4/17/2019		Fourth Virtual Connection 4 HUMAN FACTORS FOR VISUALIZATION APPLICATIONS
	5/1/2019	**ADC VIDEO SUBMISSION DEADLINE (2PM CENTRAL)**
5/9/2019	5/20/2019	**ROUND 1 ENDS, ANNOUNCEMENT OF TEAMS FOR ROUND 2. ROUND 2 BEGINS**
5/10/2019		ROUND 2 TEAMS RECEIVE INTERVIEW QUESTIONS
5/15/2019	5/16/2019	VIRTUAL INTERVIEWS WITH AA-2 TEAM
	5/20/2019	**ROUND 2 ENDS, ANNOUNCMENT OF SELECTED TEAMS**
5/20/2019	6/10/2019	WORK WITH MENTORS FOR ONSITE EXPERIENCE
	6/10/2019	PRESENT COMPLETED APP TO NASA
06/10/2019	06/14/2019	**ONSITE EXPERIENCE AT LOCAL NASA CENTER**
6/12/19	**(Tentatively scheduled)**	**AA-2 LAUNCH**

(Timeline is subject to change)

Live Virtual Connections

Round 1 teams will be supported during the challenge with live virtual connections from subject matter experts at NASA's Johnson Space Center. Virtual connections will present information essential for completing the challenge. These events will include ADC updates, app development presentations, possible schedule changes and information on future events. Each virtual connection will include a question and answer session with the subject matter expert providing the presentation.

Virtual Connection Schedule

Teams are encouraged to gather at one location to participate in these events. Teachers should lead this effort, and encourage students not to connect individually from another location.

The ADC Virtual Connection schedule is as follows; the event times will be determined:

Live Connection #1
Date: Wednesday, March 13th 2019
Topic: Welcome, ADC Introduction, Challenge Guide Q/A, and Multicasting Presentation.

Live Connection #2
Date: Wednesday, March 27th 2019
Topic: Game Engine Scripting for Object Motion, General updates, and Challenge Q/A.

Live Connection #3
Date: Wednesday, April 3rd 2019
Topic: Pseudo Coding for Quaternion, General updates, and Challenge Q/A.

Live Connection #4
Date: Wednesday, April 17th 2019
Topic: Human Factors for Visualization Applications, General updates, and Challenge Q/A.

Round 1 Participation

Teams will be challenged to create an app that the Ascent Abort-2 team could use to visualize launch data. The ADC team will provide teams with the following three components to create their apps:

1.) Flight Data: An excel spreadsheet (CSV file) with seven data points for each of the three vehicles below.
   a. Command Module (Orion capsule)
   b. Launch Abort System (LAS)
   c. Launch Rocket
2.) Data Player: A java.jar file that will multicast the 'Flight Data' from (1.). The team's app must be able to receive the AA-2 information sent out by the Data Player.
3.) AA-2 Models: Individual Wavefront (.obj) models for each of the three main pieces of the AA-2 vehicle to use for visualizations.

Each team is responsible for processing this information and creating an app to visualize this data. Teams need to download the components first then run the Data Player. The following instructions describe the process for each team.

App Coding Components

1.) Access the flight data by going to the website and downloading the data player (i.e. jar file) and the Excel CSV Data file into ONE FOLDER. The Data file contains values for the Data Player to send out.

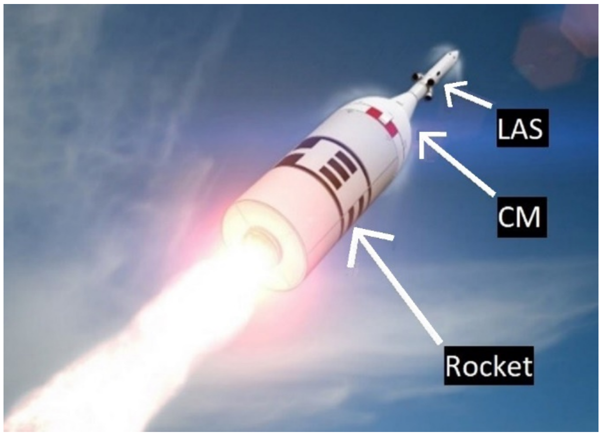
   Both items must be in the same folder for the test flight data to run.

2.) Run the Data Player by changing the directory to the folder containing the test.jar and Data.csv file and then type the following in the command prompt:
   java -jar test.jar

   The data will go out as multicast 237.7.7.7 on port 42055.

   To make sure it is working, run the data listener by typing the following in another command prompt in the same directory:
   java -cp test.jar receiver

   For help, type the following in the command prompt:
   java -jar test.jar –h



3.) Download the AA-2 models by copying the files directly from the website. The models are .obj files and can be used with a 3D rendering engine. The 3 pieces (LAS/CM/Rocket) fit together as shown in the image. The booster file refers to the rocket. Each piece has its own Wavefront (.obj) file.

Teams should contact JSC-M2MSTEM@mail.nasa.gov if they experience problems accessing these files.

Challenge Mentorship

Teams should reach out to universities, local organizations, businesses, and subject matter experts to seek guidance on coding and app development. Teams must include a short narrative, as part of the video presentation, on the connections made and how discussions helped in development of the app.

Teams needing additional assistance in developing an app can request a mentor from the ADC team. Mentors are not guaranteed to be available and will be provided on a first come, first serve basis.

Progress Report Handout

The Progress Report Handout provides a guide for teams in developing an app. Each team should begin their planning by completing an app development plan at the start of the challenge. Teams can work outside of this suggested path and use their own timeline, but it is important to create milestones for success to complete the App Development Challenge. Please note important details in the comment section including coding language, particular app details, unique coding aspects or visualizations, and concerns/challenges encountered that have altered the team's timeline.

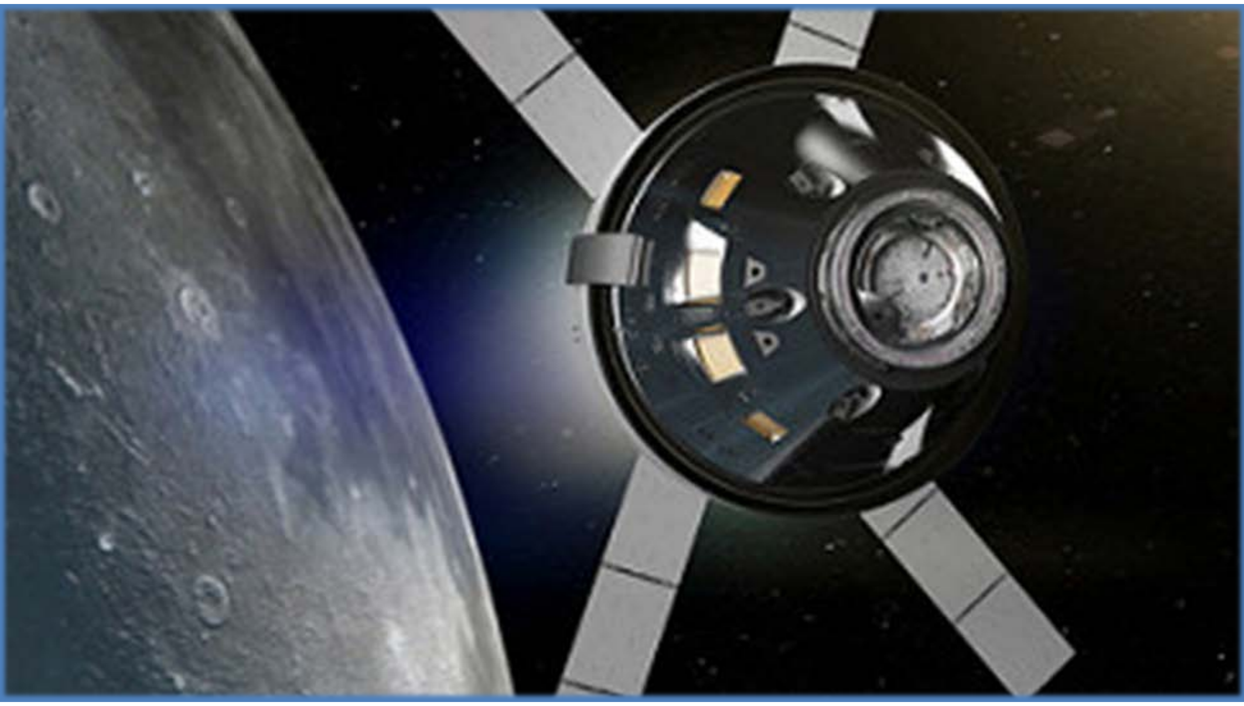
Note that the Progress Report Handout refers to a 3D rendering engine package and might not be applicable to each team. Teams can adjust this handout as needed for level of experience or specific app design concepts.

The Progress Report Handout can be found on page 24.

Story Board Handout

The Story Board Handout is for conceptualizing development of the app. A completed story board conveys what a team imagines as the final app design. During the ADC teams will encounter challenges that require them to define problems, brainstorm options, and choose the best possible solution for completing an app design. As part of the video presentation, teams must include a narrative on the challenges they encountered and how they were solved. Teams will also use the Story Board Handout to draw their main app in color, and then use the outer bubbles for additional details or descriptions of the app parts. The final app design is subject to change based on challenges encountered throughout the development process.

The Story Board Handout can be found on page 26.



Video Presentation Criteria and Scoring Rubric*

Each team will submit a five minute video that highlights student-led work on development of the AA-2 data app. All videos should be posted to YouTube in the unlisted setting. Video submissions should not be available to the public and must be in accordance with the media policies of the team's institution. The video presentation will showcase student-led work from the design phase through completion of the app. The Video Presentation Scoring Rubric, located on page 27, will be used to assess and score each team's video submission. Once the video is posted to YouTube, each lead teacher must email the ADC team a link to the video by 2pm central time on Wednesday, May 1, 2019 at JSC-M2MSTEM@mail.nasa.gov

Video Presentation Criteria

Each team must use the following script for their five-minute video submission.

1.) Introductory statement:
   a. "This is team (team name) and we worked on NASA's App Development Challenge…."
   b. Do not identify the name of any student, teacher, school, group, city, or region in the presentation
   c. Identify what coding language was utilized, what data points the app visualizes, and how successful the team was in creating a useful app
2.) Run the app, visually showing highlights of the three minutes of data
3.) Provide narration on each aspect of the app's visualizations
4.) Identify challenges the team encountered and how they were solved along the way
5.) Identify what skills the team learned and how these were acquired during the challenge
6.) Identify any guidance the team received from mentors and subject matter experts that assisted in the learning process
7.) Based on the results, explain what the team would do differently in the future to improve the app

Note these items are included in the Video Presentation Scoring Rubric on page 27.

Video Presentation Scoring Rubric*

At the end of Round 1, the Video Presentation Scoring Rubric will be used by members of the ADC team, JSC subject matter experts, mentors and NASA AA-2 personnel to evaluate video submissions. Remember to follow the script and provide details on the items requested above. Teams are encouraged to use the rubric to self-score their video in advance of the final submission. The video presentation score will be part of the selection process.

The Video Presentation Scoring Rubric can be found on page 27.

**Information about the challenge will remain posted for use in educational settings beyond the challenge end date.*

Round 2 Participation

Round 1 submissions are due by 2pm central time on Wednesday May 1, 2019. The ADC team will spend the next few days reviewing and evaluating submissions. Teams selected to participate in Round 2 can expect to be notified by close of business on May 9, 2019 (schedule subject to change).

Round 2 participants will receive questions and comments regarding their app and video presentation from NASA's AA-2 Team. When teams receive these questions, they should prepare written responses and prepare for a live virtual connection with NASA personnel and the ADC team. The live virtual connection will be 10-20 minutes long and give teams the opportunity to speak to both the AA-2 and ADC teams. Teams will provide answers to the questions and respond to additional comments. Lead teachers must email their written responses to these questions to JSC-M2MSTEM@mail.nasa.gov before May 15, 2019.

Live virtual connections will be facilitated through video conference and the date/time of the event along with questions regarding the app will be emailed to the lead teacher by the ADC team.

Each team's written submission and live connection will be evaluated to help determine those selected for the onsite Culminating Event. The selected team(s) must work with ADC mentors and team members to adjust any required items for app delivery to NASA personnel prior to attending the Culminating Event.

Teams selected to attend the Culminating Event at a NASA Field Center can expect to be notified by close of business on May 20, 2019.

Start Date	Due Date	Event
5/9/2019	5/20/2019	**ROUND 1 ENDS, ANNOUNCEMENT OF TEAMS FOR ROUND 2. ROUND 2 BEGINS**
5/10/2019		ROUND 2 TEAMS RECEIVE INTERVIEW QUESTIONS
5/15/2019	5/16/2019	VIRTUAL INTERVIEWS WITH AA-2 TEAM
	5/20/2019	**ROUND 2 ENDS, ANNOUNCMENT OF SELECTED TEAMS**
5/20/2019	6/10/2019	WORK WITH MENTORS FOR ONSITE EXPERIENCE
	6/10/2019	PRESENT COMPLETED APP TO NASA
6/10/2019	6/14/2019	**ONSITE EXPERIENCE AT LOCAL NASA CENTER**
6/12/19	**(Tentatively scheduled)**	**AA-2 LAUNCH**

(Timeline is subject to change)

Conclusion and Contact Information:

The ADC Team hopes participation in NASA's App Development Challenge is a beneficial and rewarding learning experience for each team. The ADC team will provide timely updates to lead teachers via email. Questions can be directed to JSC-M2MSTEM@mail.nasa.gov.

Supplemental Information and Resources

Learning Objectives (Computational Thinking Model)

The following K-12 Computer Science Framework's 7 Core Practices model shows examples of what will be completed for each practice by students participating in the App Development Challenge.

K-12 Computer Science Framework's 7 Core Practices
(Source: https://k12cs.org/navigating-the-practices/)

1. Fostering an inclusive computing culture
    - Identify a variety of people who can provide ideas, mentorship, and feedback
    - Identify a variety of people who might be end-users for the product
2. Collaborating around computing
    - Select the programming language, operating system, and development environment to be used based upon the team's skill-set and experience

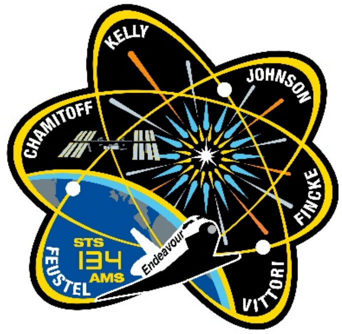
    - Identify roles & responsibilities for team members, as well as norms for ensuring everyone has a voice

3. Recognizing and defining computational problems
   - Explain the challenge
   - Break the challenge into smaller chunks
   - Develop storyboards for visualizing a final app
4. Developing and using abstractions
   - Identify and incorporate existing libraries, modules, images and 3D models that may provide useful features in visualizing an app
   - Code useful subroutines (i.e. recognizing keyboard and mouse input) that may be used multiple times in the final app
5. Creating computational artifacts
   - Develop some pseudo-code to guide development of the app
   - Code a multicast listener that receives data from the multicast sender application (provided by NASA)
   - Code a conversion from the quaternion data into more useful forms (e.g. pitch, roll, and yaw in rectangular coordinates)
   - Code ways to visualize the location and rotation of one (or more) of the three vehicle parts
   - Code other useful features
6. Testing and refining computational artifacts
   - Run the app with a small selection of the NASA-provided data either manually or via the multicast player
   - Run an entire mission from the multicast player with the app receiving and displaying the information in real time
   - Let other people “beta test” the app and provide feedback
   - Add “junk data” to the mission data, and ensure the app can handle the errors
7. Communicating about computing
   - Script and deliver a presentation on how the team’s app works, what they learned making it, and what ideas they have for future improvements
   - Record a video of team presentation and submit it to NASA

Digital Badges for Students and Educators

The NASA STEM Educator Professional Development Collaborative (NASA EPDC) provides professional learning services via online activities for educators and students. These lessons are directly aligned to NASA mission objectives. An electronic badge, i.e. completion certificate, is awarded to any student or teacher upon completion of the course. Teachers can also use this badge for professional development hours with their school districts. The activities are typically three to six hours long. Teachers are encouraged to visit the website and utilize these lessons in their classroom. Please note that the team(s) selected for the Culminating Event at a NASA Field Center might complete one of these activities onsite at NASA.

Team Patch Design



NASA mission patches are iconic images developed by astronauts and program teams for their missions over the past 50 years. ADC teams are encouraged to develop a mission patch for inclusion in the video presentation at the end of Round 1. NASA personnel may request the team's patch. Various online articles discuss mission patches. Click here to see the complete history of NASA mission patches from the NASA History Program Office. Click here to view an activity on mission patch design.

Orion's Ascent Abort 2 Mission Patch

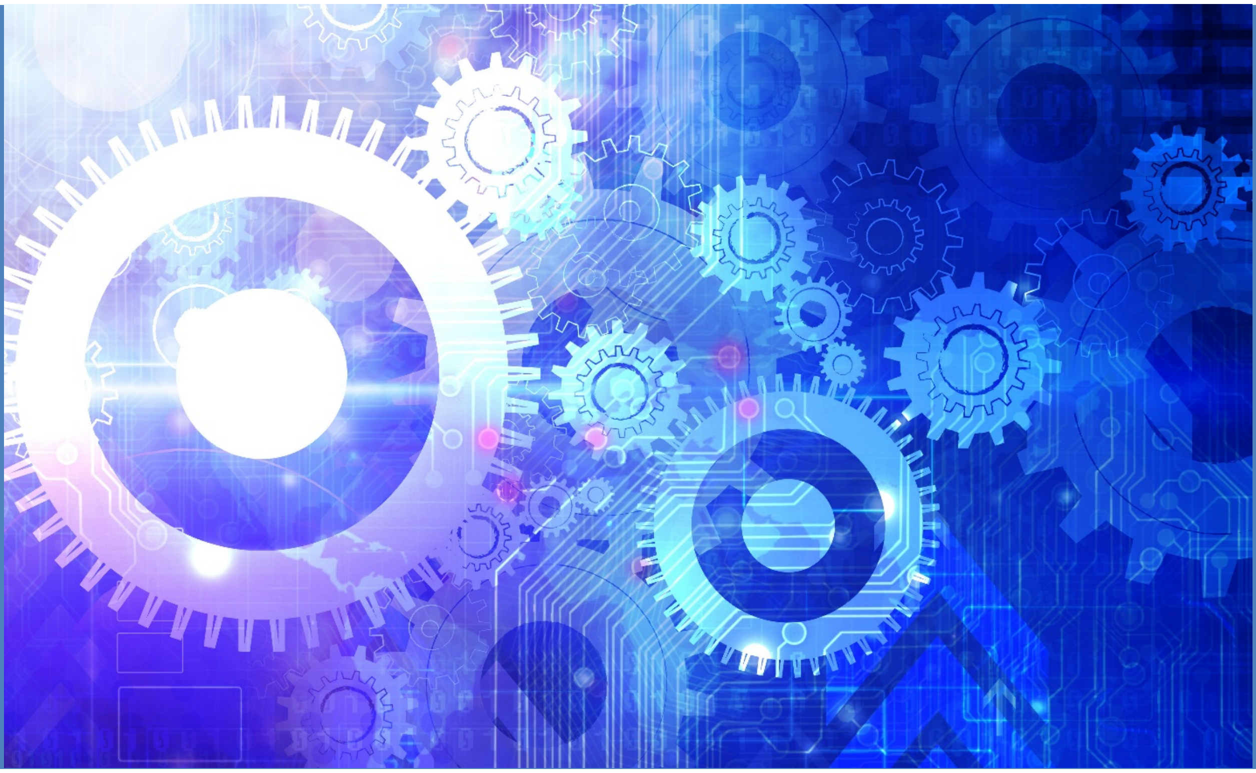
Mission Resources

- We are NASA – NASA's next great leap video. The next chapter in space exploration
- Orion Soars on First Flight – NASA's Orion Spacecraft successfully launched December 5th 2014 on its first flight
- Moon to Mars: SLS and Orion – Sustainable program of exploration
- AA-2 Test Overview Animation – AA-2 Launch Video Animation, launch currently scheduled no earlier than June 2019
- AA-2 Fact Sheet – AA-2 Flight Test Information Sheet
- AA-2 Everything You Needed To Know – NASA update on status of AA-2 and links to program information
- Orion AA-2 – Orion AA-2 Information
- Launch Abort System Fact Sheet – Orion Launch Abort System Information Sheet
- Orion Overview Fact Sheet – Orion Spacecraft Overview
- Orion Quick Facts – NASA Facts, Orion Spacecraft
- Orion Recovery – Orion Recovery Operations Information Sheet
- Exploration Mission-1 – NASA Exploration Mission 1 (EM1) website
- NASA Orion Program – NASA Orion website
- Space Launch System – NASA Space Launch System (SLS) website
- NASA Moon to Mars – News report, sustainable campaign to return to the Moon

STEM Engagement Resources

- NASA Computer Science Educational Resources
- Orion Classroom Combo Activity
- NASA Space App Challenge
- Exploration Design Challenge: The formal challenge ended in 2014, but the activities are still available
- Parachuting on Mars Activity Guide
- Design a Crew Vehicle Activity Challenge: 6-8

Appendix



Team Progress Report

Team EXPLORE MOON to MARS APP DEVELOPMENT CHALLENGE: ASCENT ABORT 2 Name: ______________________________

	Action	Anticipated Completion Date	Actual Completion Date	Comments
A	Select 3D rendering engine package and the programming language that will be used to interface with the telemetry multicast program.			
B	Create a Multicast listener (in the language selected in step (A)). This program receives data from the multicast sender application and prints out the value of each of the 21 parameters (just like the sample java "receiver" program provided).			
C	Import the 3 vehicle modes into the 3D rendering engine selected in step (A).			
D	Move each of the 3D models to a (specified) X/Y/Z position (in whatever coordinate frame the 3D engine uses).			
E	Rotate each of the 3D models to (specified) orientations (in whatever orientation coordinates the 3D display engine uses).			

EXPLORE MOONtoMARS

APP DEVELOPMENT CHALLENGE: ASCENT ABORT 2

	Action	Anticipated Completion Date	Actual Completion Date	Comments
F	Modify program in step (B) so it can convert the position/quaternion information into whatever coordinate frames/angles/units are needed by the rendering engine selected in step (A).			
G	Use 1 data sample from the telemetry player to drive the position of 1 of the models in the 3D display engine (i.e. have one model placed at the position/orientation of one sample point from sample data file).			
H	Drive all 3 models using a single sample position. (like step (G), but for all 3 models).			
I	Drive all 3 models using live data (like step (H), using the positions/orientation provided by step (F)).			
J	Add text/gauge readouts into the app (using specified data values).			
K	Tie live telemetry from step (F) into the gauges/text from step (J).			
L	Run an entire mission from the multicast player with the app receiving and displaying the information in real time.			

EXPLORE MOON to MARS

APP DEVELOPMENT CHALLENGE: ASCENT ABORT 2

Story Board Handout

Video Presentation Scoring Rubric

Category	Best = 3 points	Better = 2 points	Good = 1 point	Missing = 0 points	TOTAL POINTS
App Creation - Graphics and Visualizations	App visualizes both position and rotation using one or more of the provided models/graphics.	App visualizes both position and rotation using graphs or basic shapes.	App shows visualization for either position or rotation, but not both.	App fails to visualize data in some useful form beyond text.	Raw Score Multiplier x 3
App Creation - Data Processing	App runs complete set of data uninterrupted and smoothly without glitches.	App runs complete set of data with some delays, stutters or other glitches.	App runs two minutes or less of the data.	App fails to run any data.	Raw Score Multiplier x 3
Communicating Computing	Fully describe each app component with detailed information into the work required to complete each component and how the team worked collaboratively to complete.	Describe more then one app component OR describe your only app component with detailed information but not completely identifying the component or how the team worked to complete.	Describe at least one app component but fail to fully describe its function and how the team worked to complete.	Identification of app components and how the team worked to complete is not included.	Raw Score
Troubleshooting	Challenges encountered are clearly identified with explanations on how the team worked to evaluate and solve these items and the result of these actions.	Challenges encountered are identified with limited explanations on how the team worked to evaluate and solve these items and results of these actions.	Challenges are implied or not fully stated. It is not identified if any action is taken to solve these items.	Narrative of challenges encountered is not included or incomplete	Raw Score
Student Acquired Skills	Key skills learned and acquired through participation in the challenge are identified with detailed information and supported with visible app components.	Key skills learned are stated but with limited details of how these were acquired or incorporated into the app development.	Key skills learned are stated but no additional narrative on how these were acquired or incorporated into the development of the app is provided.	Narrative on student acquired skills is not included or incomplete.	Raw Score
Subject Matter Experts and Mentors	Interactions with scientists, engineers, or other relevant SME's is fully discussed, including how their guidance and feedback is incorporated into the app design.	Interactions with scientists, engineers, or other relevant SME's is discussed and details are included on guidance they provided.	Interactions with scientists, engineers, or other relevant SME's is acknowledged in general terms, but no specific details are included.	Scientists, engineers or SME interactions are not acknowledged.	Raw Score
			FINAL SCORE	(out of 30)	
			FINAL % SCORE	%	

Team Name: ____