

catalogs the information into a streamlined rating infrastructure.

This system consists of four distinguishing components: image repository, database, server-side logic, and Android mobile application. The image repository contains images from various JPL flight projects. The database stores the image information as well as the user rating. The server-side logic retrieves the image information from the database and categorizes each image for display. The Android mobile application is an interfacing delivery system that retrieves the image information from the server for each Android mobile device user. Also created is

a reporting and tracking system for charting and monitoring usage.

Unlike other Android mobile image applications, this system uses the latest emerging technologies to produce image listings based directly on user input. This allows for countless combinations of images returned. The back-end infrastructure uses industry-standard coding and database methods, enabling future software improvement and technology updates. The flexibility of the system design framework permits multiple levels of display possibilities and provides integration capabilities. Unique features of the software include

image/video retrieval from a selected set of categories, image Web links that can be shared among e-mail users, sharing to Facebook/Twitter, marking as user's favorites, and image metadata searchable for instant results.

This work was done by Jon D. Nelson, Sandy C. Gutheinz, Joshua R. Strom, Jeremy M. Arca, Martin Perez, Karen Boggs, and Alice Stanboli of Caltech for NASA's Jet Propulsion Laboratory. For more information, see <http://www.jpl.nasa.gov/apps/spaceimages/>.

This software is available for commercial licensing. Please contact Dan Broderick at Daniel.F.Broderick@jpl.nasa.gov. Refer to NPO-47961.

➤ Kinect Engineering with Learning (KEWL)

Lyndon B. Johnson Space Center, Houston, Texas

According to a Nielsen survey at the time of this reporting, 41% of all households have a game console. This is one market in which NASA has been absent from education and outreach efforts. Kinect Engineering with Learning (KEWL) is made to enter into that market and bring NASA education and outreach to a very familiar venue. KEWL creates an education and outreach experience that is more participatory, both in a school and museum environment.

KEWL is a set of applications that runs on an Xbox 360 (see Figure 1) using the Kinect controller used for education and outreach. These applications currently include: Train R2 (see Figure 2), a visual simulation of Robonaut 2 that allows students to control a virtual R2 in a game environment; Drive R2, an interface using the Xbox 360 and Kinect controller that allows students to control the real R2 using the methods they learned playing Train R2; ISS experience, a virtual tour of the interior of the International Space Station where students use their body to fly through the virtual ISS; Gravity Ball, a simulation of throwing balls in the gravity of different planets; Solar Array repair, a simulation of the simplified STS-121 solar array repair mission; and PlaySpace, a Mars/Moon application that allows students to experience different aspects of Mars/Moon.

Users can "fly through" the ISS using their body, allowing an experience similar to what an astronaut would have on orbit. In PlaySpace, users can fly over the surface of Mars and view surface data obtained by Mars rovers. Users of Train R2



Figure 1. The Xbox 360 is one of the most recognized gaming consoles.



Figure 2. In Train R2, the student learns to control a simulated R2 using simple poses.

and Drive R2 can experience what it is like to control a robot over a distance with a time delay, simulating the time delay that would occur between ground control and an on-orbit robot. The initial ISS experiences were built using parts of code from the NASA Enigma software. The models used in these ex-

periences were also from the Integrated Graphics Operations and Analysis Lab model database. The PlaySpace experience incorporates surface data obtained from NASA rovers and satellites and was built by NASA JPL.

This work was done by Sharon Goza and David Shores of Johnson Space Center;

William Leu, Raymond Kraesig, Eric Riche-son, Clinton Wallace, Moses Hernandez, and Cheyenne McKeegan of Tietronix Software Inc.; and Jeffrey Norris, Victor Luo, Alexander Menzies, Dara Kong, and Matt Clausen of JPL. Further information is contained in a TSP (see page 1). MSC-25110-1

▶ **Spacecraft 3D Augmented Reality Mobile App**

NASA's Jet Propulsion Laboratory, Pasadena, California

The Spacecraft 3D application allows users to learn about and interact with iconic NASA missions in a new and immersive way using common mobile devices (see figure). Using Augmented Reality (AR) techniques to project 3D renditions of the mission spacecraft into real-world surroundings, users can interact with and learn about Curiosity, GRAIL, Cassini, and Voyager. Additional updates on future missions, animations, and information will be ongoing.

Using a printed AR Target and camera on a mobile device, users can get up close with these robotic explorers, see how some move, and learn about these engineering feats, which are used to expand knowledge and understanding about space.

The software receives input from the mobile device's camera to recognize the presence of an AR marker in the camera's field of view. It then displays a 3D rendition of the selected spacecraft



Spacecraft 3D Application allows one to interact and learn about different missions.

in the user's physical surroundings, on the mobile device's screen, while it tracks the device's movement in relation to the physical position of the spacecraft's 3D image on the AR marker.

This work was done by Kevin J. Hussey, Paul R. Doronila, Brian E. Kumanchik, Evan G. Chan, and Douglas J. Ellison of Caltech; and Andrea Boeck and Justin M. Moore of Mooreboeck Inc. for NASA's Jet Propulsion Laboratory.. For more information access:

<https://play.google.com/store/apps/details?id=gov.nasa.jpl.spacecraft3D>

<http://www.space.com/16569-nasa-app-spacecraft-hand.html>

http://www.nasa.gov/mission_pages/msl/news/app20120711.html

[http://www.engadget.com/2012/07/11/nasa-spacecraft-3d-ios-app/.](http://www.engadget.com/2012/07/11/nasa-spacecraft-3d-ios-app/)

This software is available for commercial licensing. Please contact Dan Broderick at Daniel.F.Broderick@jpl.nasa.gov. Refer to NPO-48763.

▶ **MPST Software: grl_pef_check**

NASA's Jet Propulsion Laboratory, Pasadena, California

This innovation is a tool used to verify and validate spacecraft sequences at the predicted events file (PEF) level for the GRAIL (Gravity Recovery and Interior Laboratory, see http://www.nasa.gov/mission_pages/grail/main/index.html) mission as part of the Multi-Mission Planning and Sequencing Team (MPST) operations process to reduce the possibility for errors. This tool is used to catch any sequence related errors or issues immediately after the seqgen modeling to streamline downstream processes.

This script verifies and validates the seqgen modeling for the GRAIL MPST

process. A PEF is provided as input, and dozens of checks are performed on it to verify and validate the command products including command content, command ordering, flight-rule violations, modeling boundary consistency, resource limits, and ground commanding consistency. By performing as many checks as early in the process as possible, grl_pef_check streamlines the MPST task of generating GRAIL command and modeled products on an aggressive schedule.

By enumerating each check being performed, and clearly stating the criteria and assumptions made at each step,

grl_pef_check can be used as a manual checklist as well as an automated tool. This helper script was written with a focus on enabling the user with the information they need in order to evaluate a sequence quickly and efficiently, while still keeping them informed and active in the overall sequencing process. grl_pef_check verifies and validates the modeling and sequence content prior to investing any more effort into the build. There are dozens of various items in the modeling run that need to be checked, which is a time-consuming and error-prone task. Currently, no software exists that provides this functionality. Com-