This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

SAND2022-9373C

Sandia National Laboratories

Exceptional service in the national interest

# Database Driven Scenario Development

## Automating Application Creation for AR and PC

Adam Ewert, Isaac Davis, James K. Daniels

Sandia XR Conference, July 2022

Unclassified Unlimited Release

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

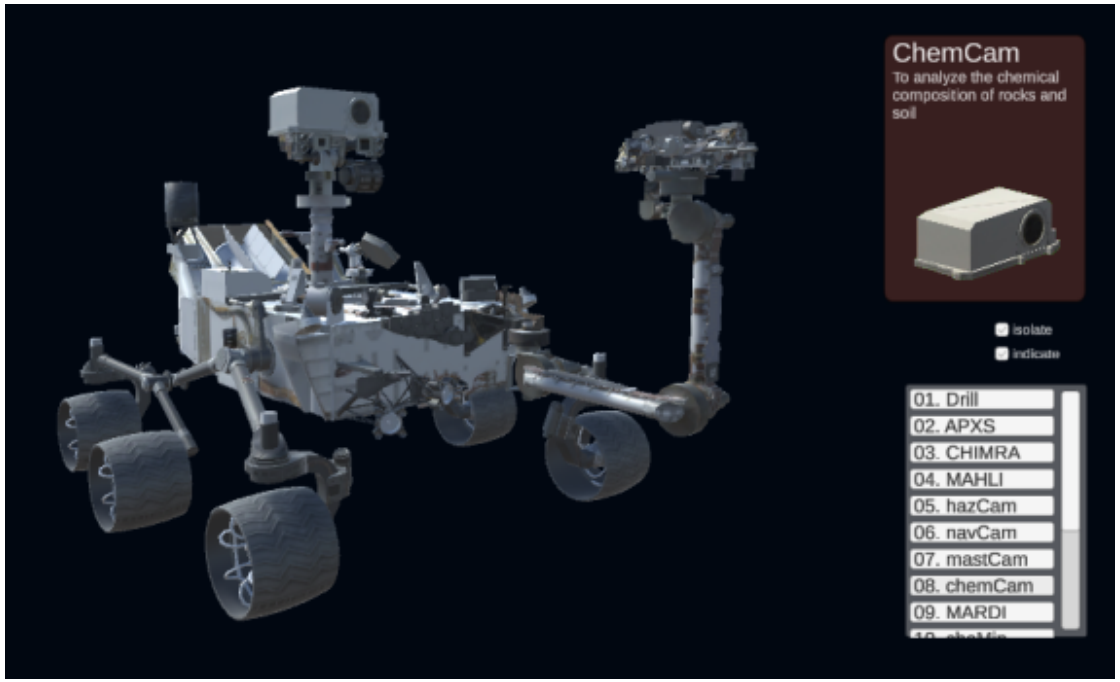
Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

# Presentation Overview

- Description of our Database Driven Design philosophy
- Why use a database driven pipeline
- Overview of our process
- Organization of the model set in Maya
- Creating the corresponding CSV files
- Demonstration of Curiosity Rover PC and AR applications
- Future development goals
- Questions

# Our Database Driven Design philosophy

- Creating applications that are intended as a teaching aid to convey information beyond standard CAD metadata
- Creating CSV file along with an organized 3d model set to drive actions and information display in the applications
- Consistent outcomes between models, customers, projects
- Ability to make changes seamlessly and maintain over time



# Why use a database driven pipeline?

Faster Development & Deployment

Reduce the Need for Expert Labor

Reduce Human Error and Labor

Make Applications Modifiable & Maintainable

# How it changes the pipeline

The Optimization stage now includes organizing the geometry into a readable format as well as spawning the CSV files that correspond to it.

During development the C# scripts in Unity are used to read the CSV files, generate appropriate UI elements, and perform manipulations.

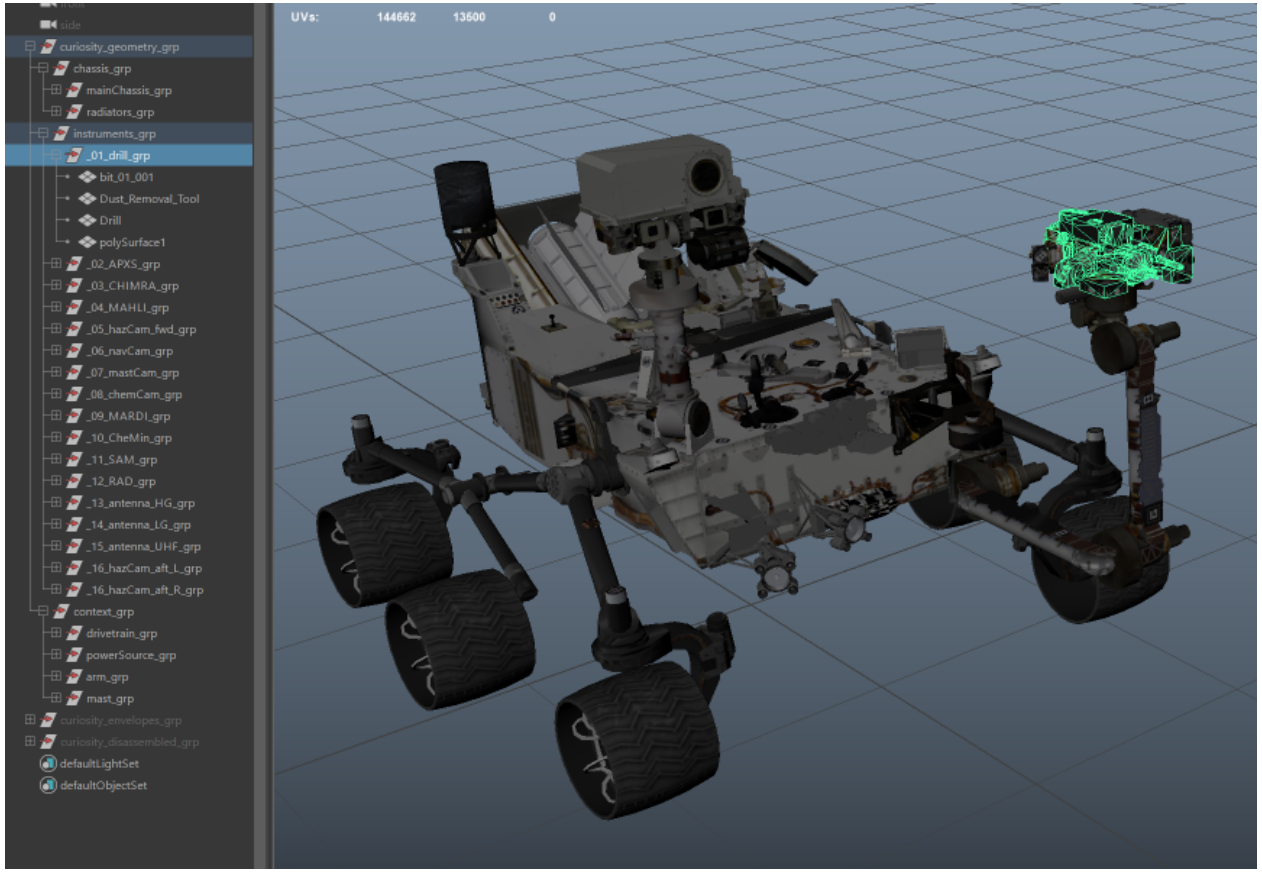
## Pipeline Summary

How we get useable applications to our customers:

- Ideation and Planning
- File Management
- Asset Acquisition
- Optimization
- Asset Generation
- Application Development
- Testing and Feedback
- Final Delivery

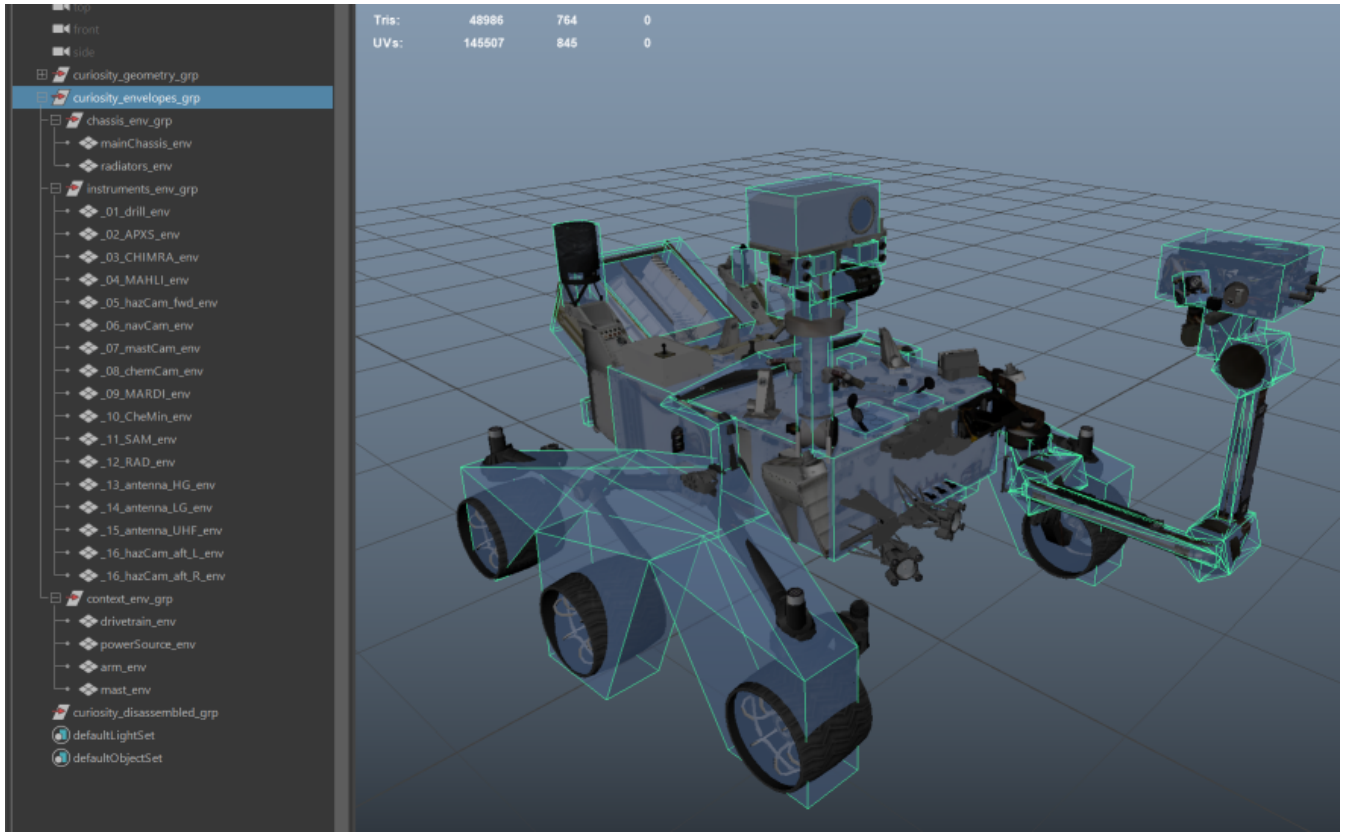
# Preparing and Organizing the Model

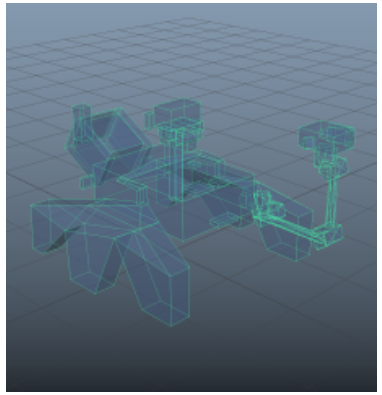
- Optimization
  - Models need to be clean
  - Textures created
- Organization
  - Sort components into groups based on the needs of the customer
  - 3 Main Groups
    - Outer Housings
    - Instruments
    - Contextual Parts
- Using groups allows us to flexibly change geometry when making updates

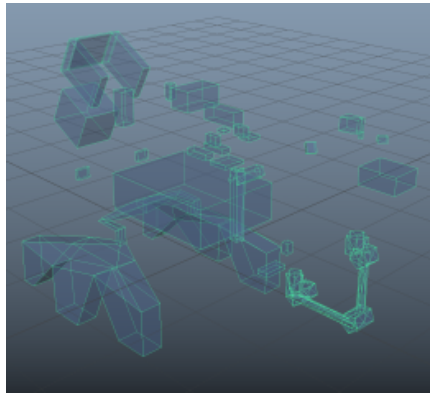


# Collision Envelopes

- Generate envelopes
  - Used as the collision mesh and parent in Unity
- More precise than generic capsule colliders
- Can be used to define alternate positions and orientations





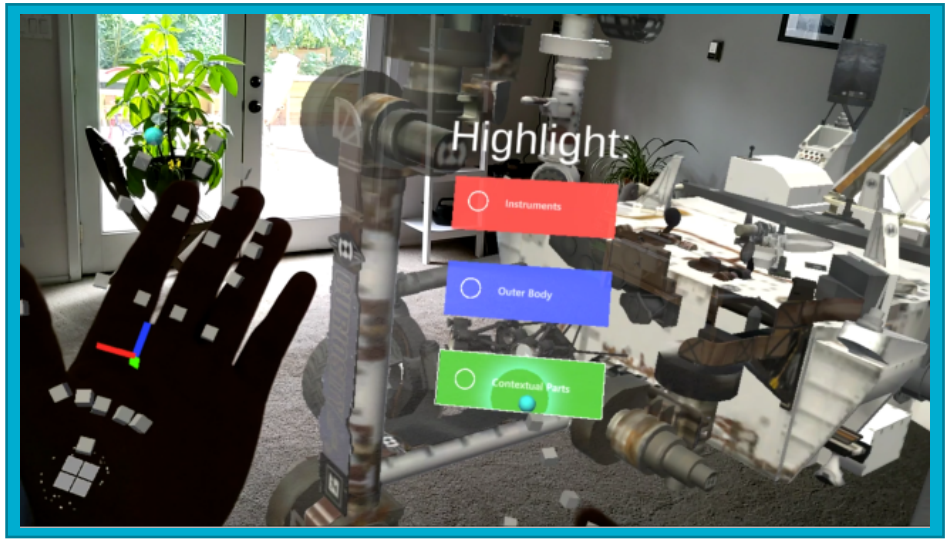
# Creating the corresponding CSV file

- CSV file contains information for object type
  - i.e. information, housing, context or more
- Contains hierarchical relationships for dynamically reparenting groups in the application
- Can contain additional information that will be displayed
  - Descriptions, names, part numbers etc.

| | A | B | C | D |
|---|---|---|---|---|
| 1 | geo name | section level | assembly level | component level |
| 2 | **mainChassis_grp** | **mainChassis_env** | **mainChassis_env** | **mainChassis_env** |
| 3 | radiators_grp | **mainChassis_env** | **mainChassis_env** | radiators_env |
| 4 | _01_drill_grp | arm_env | _01_drill_env | _01_drill_env |
| 5 | _02_APXS_grp | arm_env | _01_drill_env | _02_APXS_env |
| 6 | _03_CHIMRA_grp | arm_env | _01_drill_env | _03_CHIMRA_env |
| 7 | _04_MAHLI_grp | arm_env | _01_drill_env | _04_MAHLI_env |
| 8 | _05_hazCam_fwd_grp | **mainChassis_env** | **mainChassis_env** | _05_hazCam_fwd_env |
| 9 | _06_navCam_grp | mast_env | _08_chemCam_env | _06_navCam_env |
| 10 | _07_mastCam_grp | mast_env | _08_chemCam_env | _07_mastCam_env |
| 11 | _08_chemCam_grp | mast_env | _08_chemCam_env | _08_chemCam_env |
| 12 | _09_MARDI_grp | **mainChassis_env** | **mainChassis_env** | _09_MARDI_env |
| 13 | _10_CheMin_grp | **mainChassis_env** | **mainChassis_env** | _10_CheMin_env |
| 14 | _11_SAM_grp | **mainChassis_env** | **mainChassis_env** | _11_SAM_env |
| 15 | _12_RAD_grp | **mainChassis_env** | **mainChassis_env** | _12_RAD_env |
| 16 | _13_antenna_HG_grp | **mainChassis_env** | **mainChassis_env** | _13_antenna_HG_env |
| 17 | _14_antenna_LG_grp | **mainChassis_env** | **mainChassis_env** | _14_antenna_LG_env |
| 18 | _15_antenna_UHF_grp | **mainChassis_env** | **mainChassis_env** | _15_antenna_UHF_env |
| 19 | _16_hazCam_aft_L_grp | **mainChassis_env** | **mainChassis_env** | _16_hazCam_aft_L_env |
| 20 | _16_hazCam_aft_R_grp | **mainChassis_env** | **mainChassis_env** | _16_hazCam_aft_R_env |
| 21 | drivetrain_grp | **mainChassis_env** | drivetrain_env | drivetrain_env |
| 22 | powerSource_grp | radiators_env | powerSource_env | powerSource_env |
| 23 | arm_grp | arm_env | arm_env | arm_env |
| 24 | mast_grp | mast_env | mast_env | mast_env |
| 25 | | | | |

# Curiosity Rover Example

- AR and PC application displaying NASA's publicly available assets
  - Shows the form and location of various instruments on the Curiosity Rover
  - Sources information and categorization from NASA's educational websites
  - Demonstrates how the ARIA pipeline can be used to create useful and educational apps for PC and HoloLens for complex geometry.

# Future Goals and Improvements

- **How can we automate repetitive tasks?**
- In Maya (python and MEL)
  - Ensure that geometry and envelope names match correctly
  - Export geometry group names and envelope names directly into CSV
- In Excel
  - Updates between CSV files
- In Unity (C#)
  - Generation of UI from CSV
  - Information displayed as live text and populated from CSV
  - Descriptive error message throws
    - Reduce time to identify errors in database or geometry
  - Merge development of PC and AR applications into a single Unity project

# Questions