America Makes

Driven by… NCDMM NATIONAL CENTER FOR DEFENSE MANUFACTURING AND MACHINING

OASIS Challenge Webinar

(Open-source Additive Scanning Implementation Strategy)

OASIS
Open-source Additive Scanning
Implementation Strategy

Name	Shawn Nesmith
Title	Senior Project Engineer
Organization	NCDMM
Email Address	shawn.nesmith@ncdmm.org

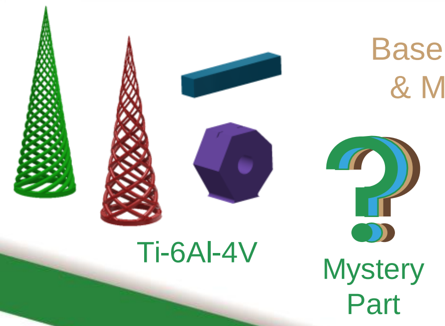
CHALLENGE

Aug 25th 2020

A direct challenge to the entire AM industry to advance laser-based powder bed additive manufacturing through the submission of open-source computer codes that output laser scan paths with innovative strategies, algorithms, and methods.

SCORING

Apr 1st 2021



Baseline code in C++ & Models provided

Criteria of Interest

Porosity
Tensile Strength
Ductility
Surface Finish

Geometric Fidelity
Code Quality
Code Run-Speed

OASIS

Open-source Additive Scanning Implementation Strategy

Are You Up for the Challenge?

SUBMISSIONS

Submissions accepted:
October 1 - December 1, 2020

Code language accepted:
Python, Java, C#, and C++

Dec 1st 2020

Submissions must be open-source

Leveraging previously developed open-source software along with the current Acceleration of Large-Scale AM (ALSAM) project.

AWARDS

$68,000 in Total Awards

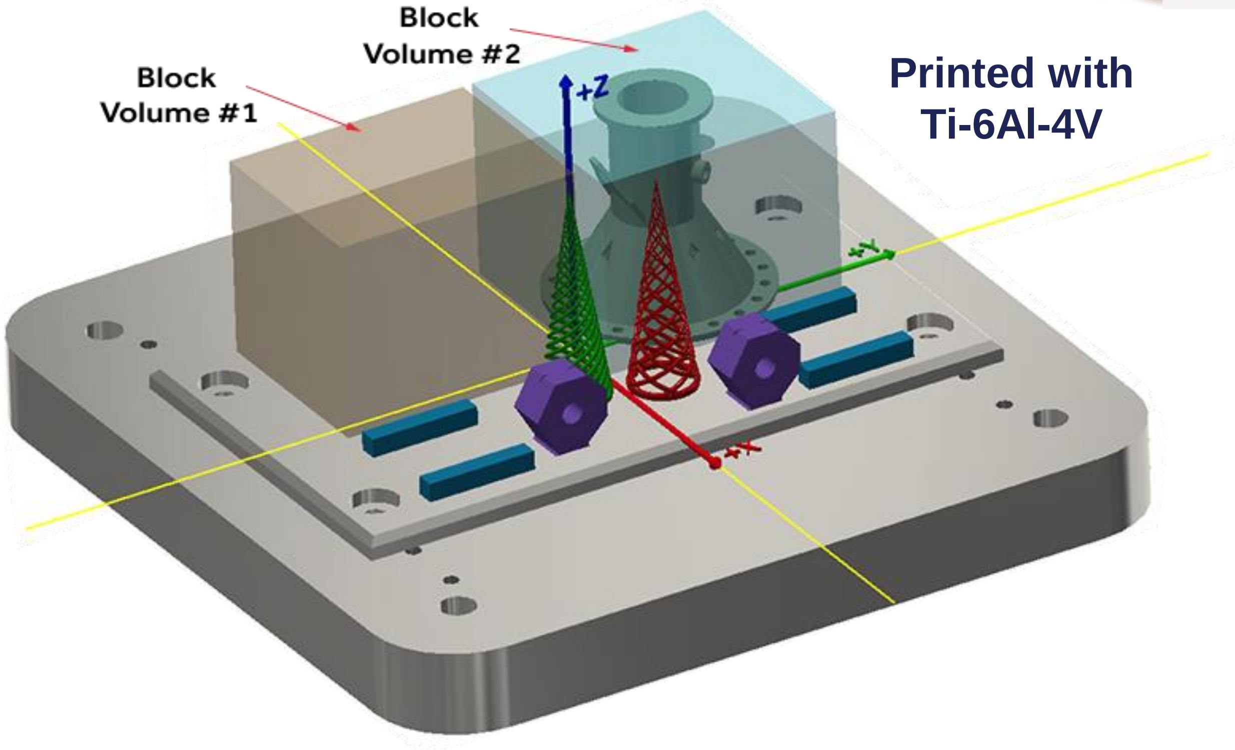
1st Place:	$25,500
2nd Place:	$15,500
3rd Place:	$10,500

3 Honorable Mentions:
$5,500 each

Awards Announced **June 4th 2021**

OASIS
Open-source Additive Scanning Implementation Strategy

Are You Up for the Challenge?



Printed with Ti-6Al-4V

Block Volume #1: Participant designed trophy / medallion

- Block volume bounding size: 60mm x 75mm x 60mm (X,Y,Z)
- Consolidated volume (including supports) is limited to 30,000mm^3 (30cm^3)
- The part must have features (i.e. holes) to allow unfused powder to drain out
- There will be no post-processing beyond cutting the part from the plate
  - no support removal, no polishing, no heat treatment, etc.

Block Volume #2: "Mystery Part"

- Block volume bounding size: 70mm x 75mm x 60mm (X,Y,Z)
- "Mystery Part" is a representative aerospace component composed of several geometric features including:
  - Thin walls
  - Thick to thin transitions
  - Various radii
  - Overhanging features
  - Channels/holes of various geometries and in multiple orientations

How the Challenge Works

The information contained in these slides are only a highlight of the rules. Please refer to the "**OASIS Challenge Documentation**" for the complete rules for the Challenge.

1. Download files and file folders from GitHub
2. Review all the documentation
3. Develop your code
4. Complete the build parameter files
5. Design your own trophy/medallion
6. Enter the contest
7. Upload submissions to private GitHub repository

Required by Contestant

8. Code evaluation
9. Top selected codes will be sent for print out of Ti-6Al-4V
10. Print evaluation
11. Announcement of winners

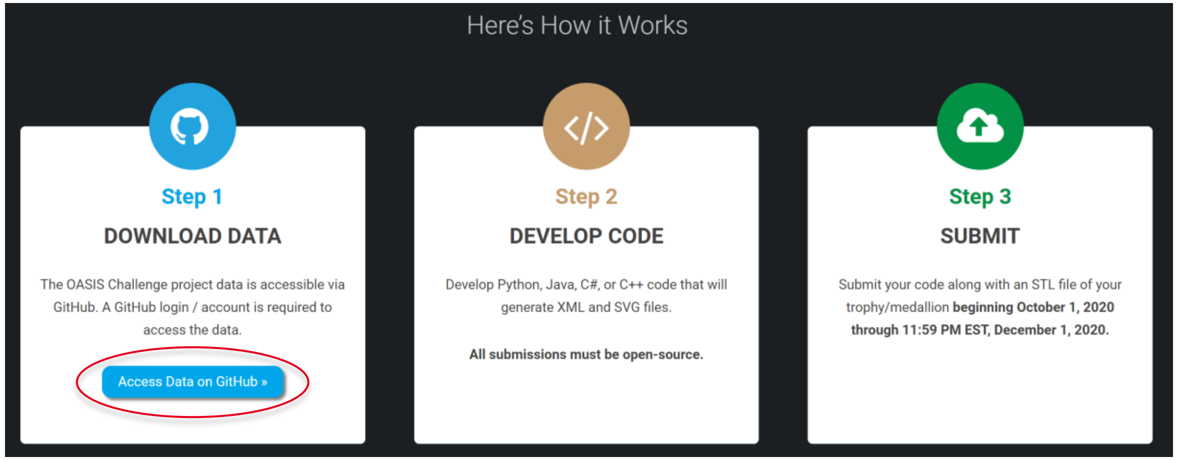
OASIS Team

OASIS Open-source Additive Scanning Implementation Strategy

1. Download files and file folders from GitHub

- Go to the OASIS website at: **https://americamakes.us/oasis**
- Click "**Access Data on GitHub**" (*a GitHub account is required*).
- Review the "**README.md**" file.
- Clone or download the files from the GitHub repository.

1. **Download files and file folders from GitHub**
2. Review all the documentation
3. Develop your code
4. Complete the build parameter files
5. Design your own trophy/medallion
6. Enter the contest
7. Upload submissions to private GitHub repository
8. Code evaluation
9. Top selected codes sent for print
10. Print evaluation
11. Announcement of winners



Here's How it Works

Step 1

DOWNLOAD DATA

The OASIS Challenge project data is accessible via GitHub. A GitHub login / account is required to access the data.

Access Data on GitHub »

Step 2

DEVELOP CODE

Develop Python, Java, C#, or C++ code that will generate XML and SVG files.

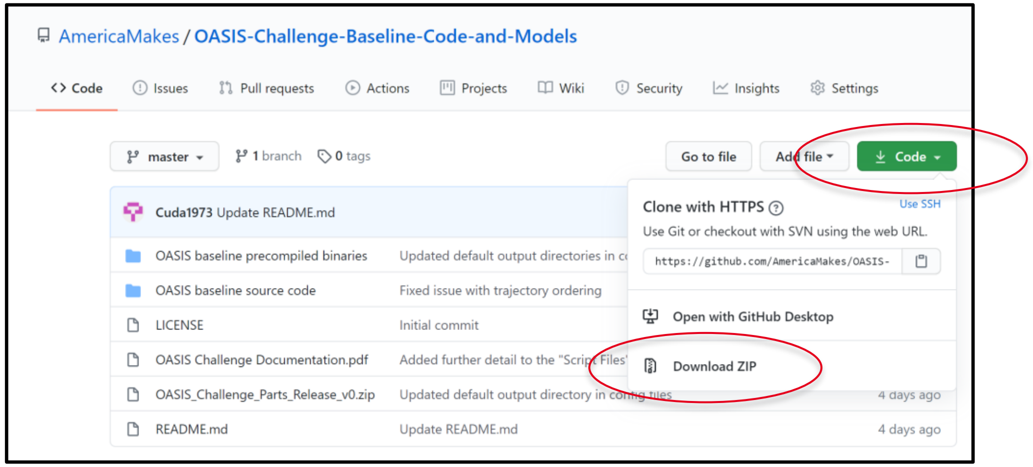
All submissions must be open-source.

Step 3

SUBMIT

Submit your code along with an STL file of your trophy/medallion **beginning October 1, 2020 through 11:59 PM EST, December 1, 2020.**

Access data on the GitHub site



AmericaMakes / **OASIS-Challenge-Baseline-Code-and-Models**

Cuda1973 Update README.md

OASIS baseline precompiled binaries	Updated default output directories in c	
OASIS baseline source code	Fixed issue with trajectory ordering	
LICENSE	Initial commit	
OASIS Challenge Documentation.pdf	Added further detail to the "Script Files'	
OASIS_Challenge_Parts_Release_v0.zip	Updated default output directory in config files	4 days ago
README.md	Update README.md	4 days ago

Clone with HTTPS

Use SSH

Use Git or checkout with SVN using the web URL.

```
https://github.com/AmericaMakes/OASIS-
```

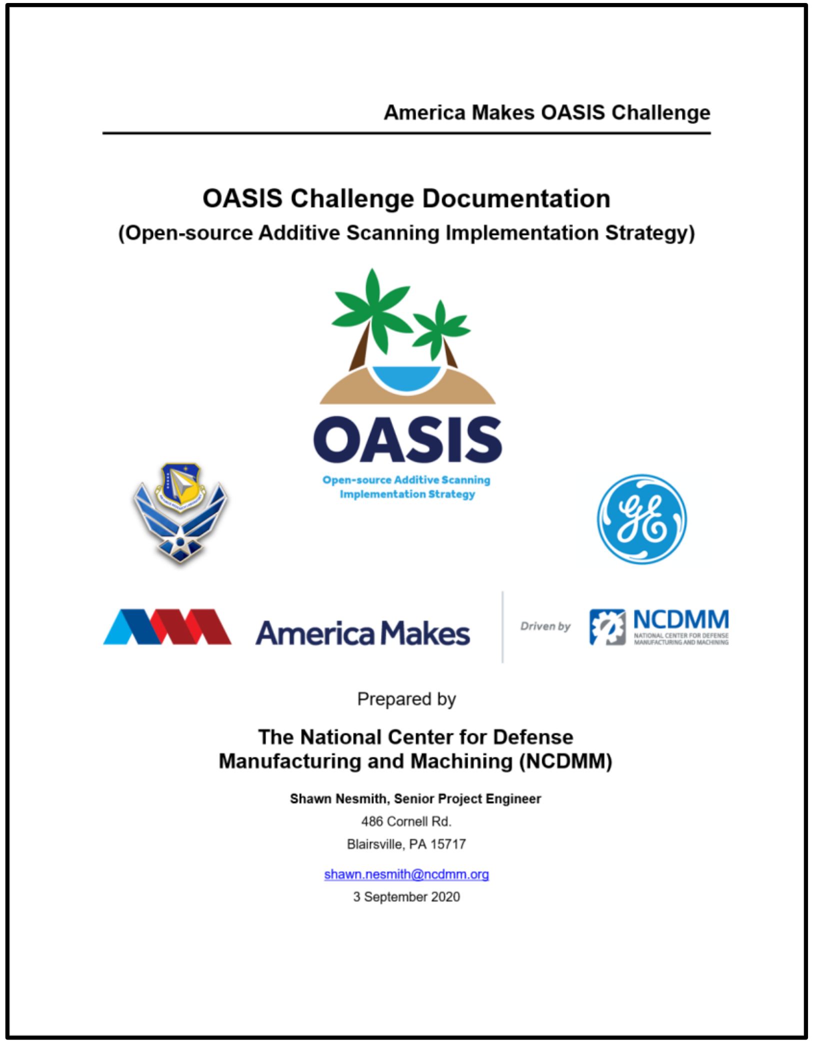
Open with GitHub Desktop

Download ZIP

Clone or download all files

2. Review all the documentation

1. Download files and file folders from GitHub
2. **Review all the documentation**
3. Develop your code
4. Complete the build parameter files
5. Design your own trophy/medallion
6. Enter the contest
7. Upload submissions to private GitHub repository
8. Code evaluation
9. Top selected codes sent for print
10. Print evaluation
11. Announcement of winners



America Makes OASIS Challenge

OASIS Challenge Documentation
(Open-source Additive Scanning Implementation Strategy)

OASIS
Open-source Additive Scanning Implementation Strategy

America Makes | Driven by NCDMM

Prepared by

The National Center for Defense
Manufacturing and Machining (NCDMM)

Shawn Nesmith, Senior Project Engineer
486 Cornell Rd.
Blairsville, PA 15717
shawn.nesmith@ncdmm.org
3 September 2020

- Thoroughly review the "**OASIS Challenge Documentation.pdf**" file.
  - This document contains all the information associated with the OASIS Challenge.
- Thoroughly review all the documentation contained within the file folders.
  - README files
  - Doc files
  - Excel files
  - Etc.
- Submit all questions to **challenge@americamakes.us**
  - Located at the bottom of the **OASIS Challenge website**.
  - Deadline for questions is November 1st, 2020.

3. Develop your code (Submitted code)

- Must be open-source (*public*) code. All private code will be disqualified.
- All submitted code must be one (1) of the following four (4) code languages:
  - **C#** / **C++** / **Java** / **Python**
- Submitted code that relies on custom libraries for performing the function that is the intent of the OASIS Challenge will be disqualified. Third-party libraries that provide general-purpose utilities are allowed. Examples of third-party libraries that would disqualify an entry are slicer and scan generation.
  - If library code is added to the entrant GitHub repository so that it can be addressed, then it is allowed.

1. Download files and file folders from GitHub
2. Review all the documentation
3. **Develop your code**
   - **Submitted code**
   - Code license
   - Script files
   - README file
   - Virtual target machine
4. Complete the build parameter files
5. Design your own trophy/medallion
6. Enter the contest
7. Upload submissions to private GitHub repository
8. Code evaluation
9. Top selected codes sent for print
10. Print evaluation
11. Announcement of winners

OASIS Open-source Additive Scanning Implementation Strategy

3. Develop your code (Code license)

- All submitted code must be clearly labeled and licensed under one (1) of the following open-source licenses:
  - **BSD 2-Clause**
  - **Apache 2.0**
  - **MIT**
- All libraries required to compile and run the code must also be licensed under one (1) of the above licenses.
- If required, subdirectories of the code may use a different license selected from the three (3) above. Include the license file in that directory. Code directories without a license file inherit the license file from the parent directory.
- System libraries that are part of the Microsoft Windows/10 or Microsoft Windows Server normal installation are allowed for code that is built on Microsoft Windows.

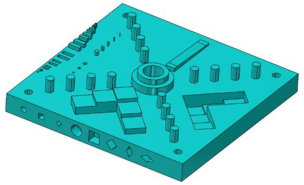
1. Download files and file folders from GitHub
2. Review all the documentation
3. **Develop your code**
   - Submitted code
   - **Code license**
   - Script files
   - README file
   - Virtual target machine
4. Complete the build parameter files
5. Design your own trophy/medallion
6. Enter the contest
7. Upload submissions to private GitHub repository
8. Code evaluation
9. Top selected codes sent for print
10. Print evaluation
11. Announcement of winners

OASIS
Open-source Additive Scanning Implementation Strategy

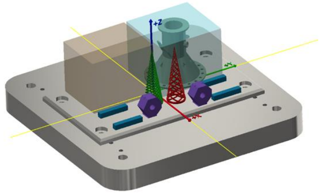
3. Develop your code (Script files)

- The submitted code must be able to be compiled via a batch file, named "***build.bat***" or "***build.sh***", which must be submitted with the source code. The resulting executable should go into the top-level directory. The executable shall reference the input configuration file and geometry(-ies) from the same directory.
- Two (2) additional batch files must be included to execute the compiled code on the two (2) different sets of geometries: "***run_NIST.bat***" for the NIST artifact (i.e. specified by the input configuration file "**OASIS_Input_Config_NIST_Plate_Baseline.xls**") and "***run_Build_Plate.bat***" for the challenge geometry (i.e. specified by the input configuration file "**OASIS_Input_Config_Build_Plate_Baseline.xls**"). The code should execute as a Windows command-line program, not as an interactive UI. The argument to the executable shall be the input configuration file.

1. Download files and file folders from GitHub
2. Review all the documentation
3. **Develop your code**
   - Submitted code
   - Code license
   - **Script files**
   - README file
   - Virtual target machine
4. Complete the build parameter files
5. Design your own trophy/medallion
6. Enter the contest
7. Upload submissions to private GitHub repository
8. Code evaluation
9. Top selected codes sent for print
10. Print evaluation
11. Announcement of winners



NIST Test Artifact



Build Plate

OASIS Open-source Additive Scanning Implementation Strategy

3. Develop your code (Script files)

- The output from the program shall be placed into a subdirectory as specified by the input configuration file (e.g. "***OASIS_NIST_Plate_Baseline***") and shall contain two (2) subdirectories: one (1) for layer files and one (1) for scan path files. The layer files subdirectory, at a minimum, will contain the SVG file output from the program. These will be used for visual inspection of the code output and shall be output at a rate of 1 per 10 layers (fixed for OASIS Challenge). The scan files subdirectory will contain two folders: one for the SVG output (fixed at 1 per 5 layers), and one (1) for XML output (1 per layer).
- The submitted code must create valid XML and SVG files as defined in the "**ALSAM3024 multiLaser XML schema 2020323.docx**" file located in the **https://github.com/AmericaMakes/OASIS-Challenge-Baseline-Code-and-Models/OASIS baseline source code/Documentation** directory.

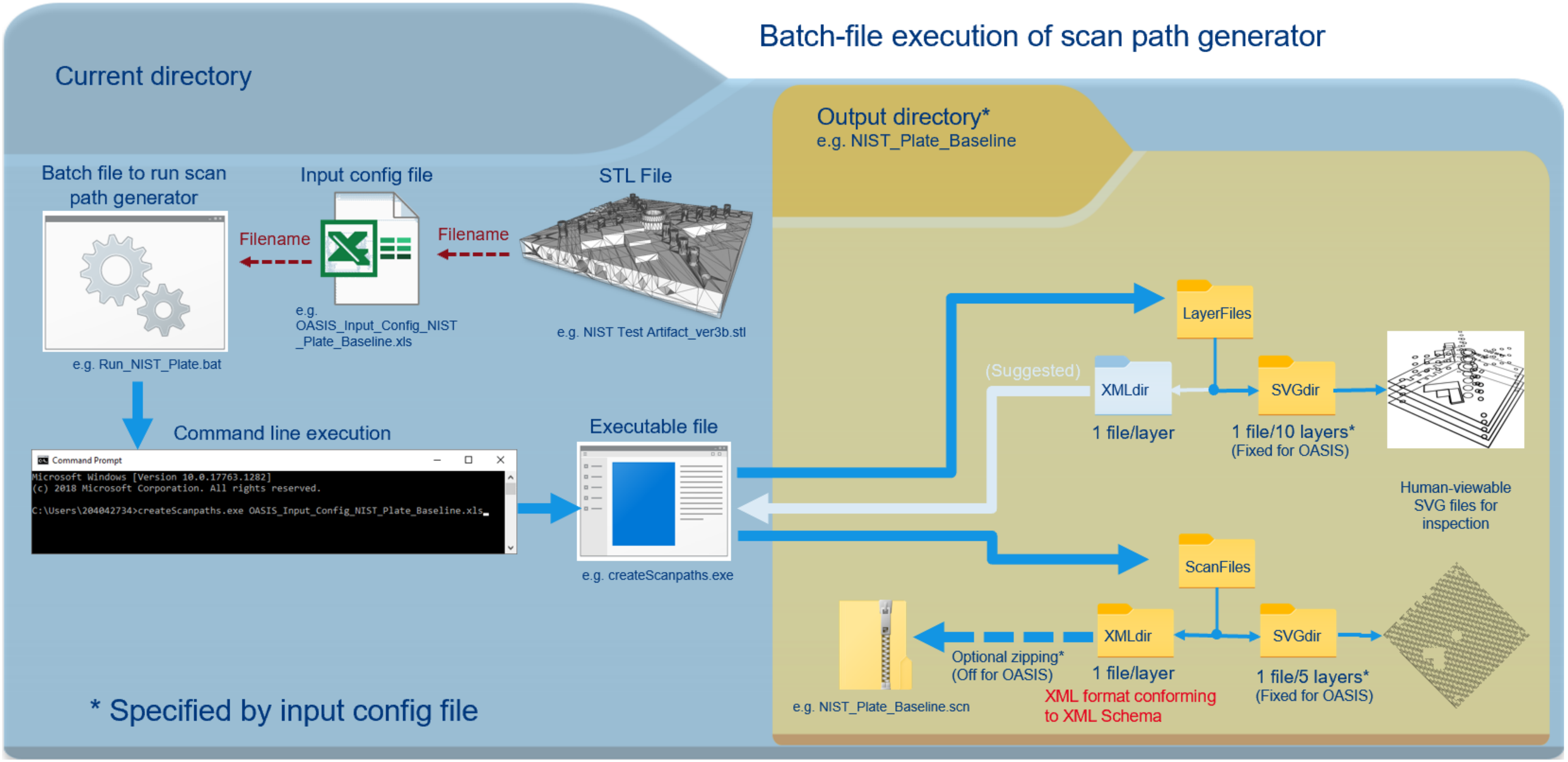
1. Download files and file folders from GitHub
2. Review all the documentation
3. **Develop your code**
   - Submitted code
   - Code license
   - **Script files**
   - README file
   - Virtual target machine
4. Complete the build parameter files
5. Design your own trophy/medallion
6. Enter the contest
7. Upload submissions to private GitHub repository
8. Code evaluation
9. Top selected codes sent for print
10. Print evaluation
11. Announcement of winners

OASIS Open-source Additive Scanning Implementation Strategy

3. Develop your code (Script files)

1. Download files and file folders from GitHub
2. Review all the documentation
3. **Develop your code**
   - Submitted code
   - Code license
   - **Script files**
   - README file
   - Virtual target machine
4. Complete the build parameter files
5. Design your own trophy/medallion
6. Enter the contest
7. Upload submissions to private GitHub repository
8. Code evaluation
9. Top selected codes sent for print
10. Print evaluation
11. Announcement of winners

Batch-file execution of scan path generator



```
Microsoft Windows [Version 10.0.17763.1282]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\204042734>createScanpaths.exe OASIS_Input_Config_NIST_Plate_Baseline.xls
```

* Specified by input config file

OASIS Open-source Additive Scanning Implementation Strategy

3. Develop your code (Script files)

- Code will be compiled using Docker containers. Entrants may provide a Dockerfile to use for building their executable with the Docker build and run commands in their script. If a Dockerfile is not provided, then the following images in Table #1 will be used to build the entries:

Docker Base Image
Table #1

Build File	Language	Docker base image
build.bat	C/C++	Probably based on mcr.microsoft.com/dotnet/core/sdk: 3.1.401-nanoserver-1809 with mingw and gcc installed.
build.bat	Python	python:3.7-windowsservercore-ltsc2016
build.bat	Java	openjdk:16-windowsservercore-ltsc2016
build.bat	C#	mcr.microsoft.com/dotnet/core/sdk: 3.1.401-nanoserver-1809
build.sh	C/C++	gcc:10.2
build.sh	Python	python:3.8.5-alpine3.12
build.sh	Java	openjdk:11
build.sh	C#	mcr.microsoft.com/dotnet/core/sdk: 3.1.401-alpine3.12

1. Download files and file folders from GitHub
2. Review all the documentation
3. **Develop your code**
   - Submitted code
   - Code license
   - **Script files**
   - README file
   - Virtual target machine
4. Complete the build parameter files
5. Design your own trophy/medallion
6. Enter the contest
7. Upload submissions to private GitHub repository
8. Code evaluation
9. Top selected codes sent for print
10. Print evaluation
11. Announcement of winners

3. Develop your code (README file)

- Code submission must include a “**README**” file describing how to build and run the submitted code.
- Must also state any custom libraries that the submitted code uses.

1. Download files and file folders from GitHub
2. Review all the documentation
3. **Develop your code**
   - Submitted code
   - Code license
   - Script files
   - **README file**
   - Virtual target machine
4. Complete the build parameter files
5. Design your own trophy/medallion
6. Enter the contest
7. Upload submissions to private GitHub repository
8. Code evaluation
9. Top selected codes sent for print
10. Print evaluation
11. Announcement of winners

3. Develop your code (Virtual target machine)

- Microsoft Server 2019 Operating System using Docker containers
- Two (2) virtual CPU's
  - Intel® Xeon® CPU E5-2673 v4

    Note: Only one (1) CPU will be dedicated to the program execution, single thread
  - 2.30 GHz
  - 64-bit OS
  - X64-based processor
  - 4GB RAM

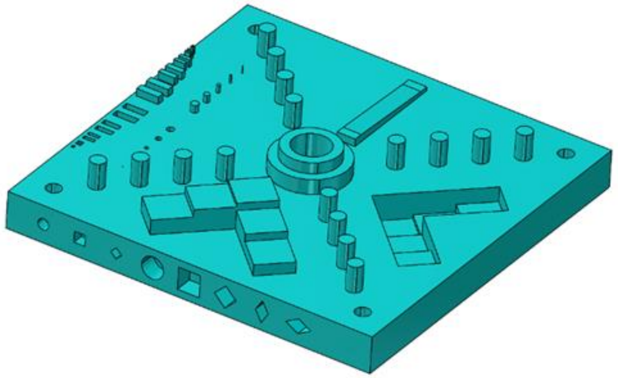
1. Download files and file folders from GitHub
2. Review all the documentation
3. **Develop your code**
   - Submitted code
   - Code license
   - Script files
   - README file
   - **Virtual target machine**
4. Complete the build parameter files
5. Design your own trophy/medallion
6. Enter the contest
7. Upload submissions to private GitHub repository
8. Code evaluation
9. Top selected codes sent for print
10. Print evaluation
11. Announcement of winners

OASIS Open-source Additive Scanning Implementation Strategy

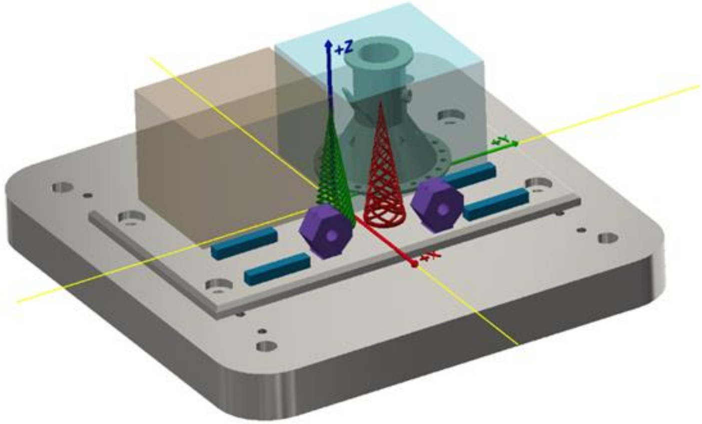
4. Complete the build parameter files

1. Download files and file folders from GitHub
2. Review all the documentation
3. Develop your code
4. **Complete the build parameter files**
5. Design your own trophy/medallion
6. Enter the contest
7. Upload submissions to private GitHub repository
8. Code evaluation
9. Top selected codes sent for print
10. Print evaluation
11. Announcement of winners

- Two (2) pre-filled configuration files are provided as part of the baseline source code for the two (2) code evaluations.
  - **"OASIS_Input_Config_NIST_Plate_Baseline.xls"**
    - Configured to run the scan path generator on the NIST test artifact geometry.
  - **"OASIS_Input_Config_Build_Plate_Baseline.xls"**
    - Configured to run the rest of the Challenge geometries on the build plate.
    - Contains placeholders for the "**Mystery Part**" and the user's designed trophy/medallion part.



NIST Test Artifact

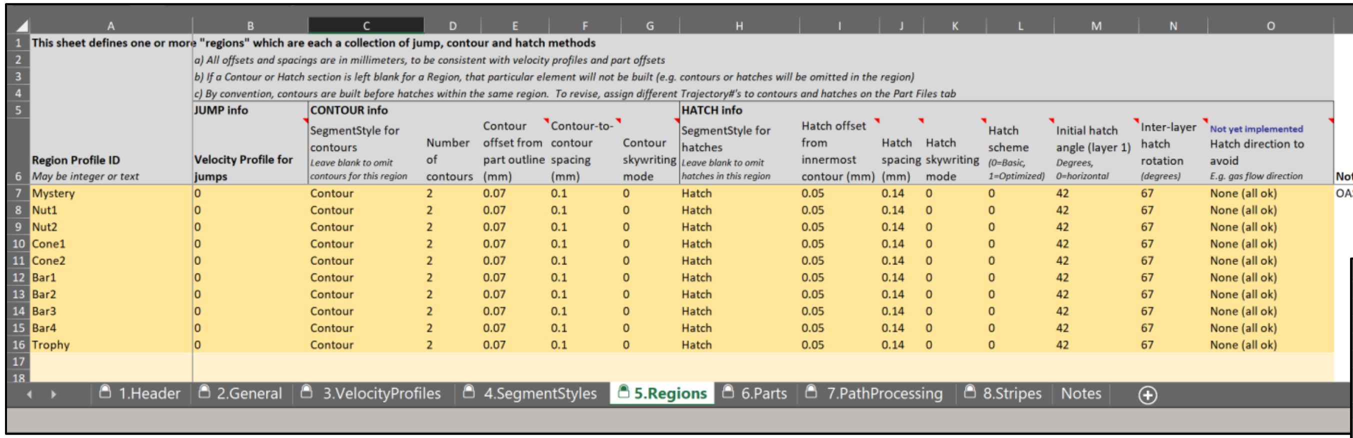


Build Plate

4. Complete the build parameter files

1. Download files and file folders from GitHub
2. Review all the documentation
3. Develop your code
4. **Complete the build parameter files**
5. Design your own trophy/medallion
6. Enter the contest
7. Upload submissions to private GitHub repository
8. Code evaluation
9. Top selected codes sent for print
10. Print evaluation
11. Announcement of winners

- Thoroughly review the "OASIS allowable build parameter ranges.pdf" file.
  - Adjustable scan strategy parameters: ***can*** be changed by participant.
  - Fixed process parameters: ***cannot*** be changed by the participant.
    - Cells in the spreadsheets with greyed out text are fixed for the purpose of this Challenge.
  - Your scan strategy may require more parameters than are provided by the baseline configuration file. You may add additional parameters to the configuration file, but do not remove any of the existing parameters.



This sheet defines one or more "regions" which are each a collection of jump, contour and hatch methods

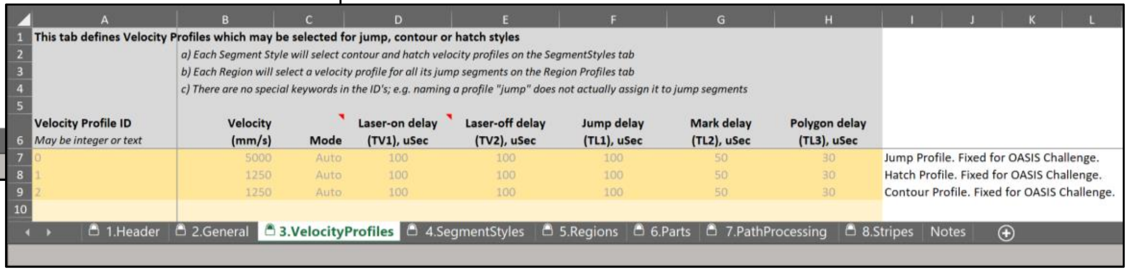
a) All offsets and spacings are in millimeters, to be consistent with velocity profiles and part offsets

b) If a Contour or Hatch section is left blank for a Region, that particular element will not be built (e.g. contours or hatches will be omitted in the region)

c) By convention, contours are built before hatches within the same region. To revise, assign different Trajectory#'s to contours and hatches on the Part Files tab

	JUMP info	CONTOUR info					HATCH info								
Region Profile ID *May be integer or text*	Velocity Profile for jumps	SegmentStyle for contours *Leave blank to omit contours for this region*	Number of contours	Contour offset from part outline (mm)	Contour-to-contour spacing (mm)	Contour skywriting mode	SegmentStyle for hatches *Leave blank to omit hatches in this region*	Hatch offset from innermost contour (mm)	Hatch spacing (mm)	Hatch skywriting mode	Hatch scheme *(0=Basic, 1=Optimized)*	Initial hatch angle (layer 1) *Degrees, 0=horizontal*	Inter-layer hatch rotation *(degrees)*	Not yet implemented Hatch direction to avoid *E.g. gas flow direction*	**Note** (for your documentation; not processed or read)
Mystery	0	Contour	2	0.07	0.1	0	Hatch	0.05	0.14	0	0	42	67	None (all ok)	OASIS Challenge: Freely modify and add to these parameters!
Nut1	0	Contour	2	0.07	0.1	0	Hatch	0.05	0.14	0	0	42	67	None (all ok)	
Nut2	0	Contour	2	0.07	0.1	0	Hatch	0.05	0.14	0	0	42	67	None (all ok)	
Cone1	0	Contour	2	0.07	0.1	0	Hatch	0.05	0.14	0	0	42	67	None (all ok)	
Cone2	0	Contour	2	0.07	0.1	0	Hatch	0.05	0.14	0	0	42	67	None (all ok)	
Bar1	0	Contour	2	0.07	0.1	0	Hatch	0.05	0.14	0	0	42	67	None (all ok)	
Bar2	0	Contour	2	0.07	0.1	0	Hatch	0.05	0.14	0	0	42	67	None (all ok)	
Bar3	0	Contour	2	0.07	0.1	0	Hatch	0.05	0.14	0	0	42	67	None (all ok)	
Bar4	0	Contour	2	0.07	0.1	0	Hatch	0.05	0.14	0	0	42	67	None (all ok)	
Trophy	0	Contour	2	0.07	0.1	0	Hatch	0.05	0.14	0	0	42	67	None (all ok)	

Adjustable scan strategy parameters example



This tab defines Velocity Profiles which may be selected for jump, contour or hatch styles

a) Each Segment Style will select contour and hatch velocity profiles on the SegmentStyles tab

b) Each Region will select a velocity profile for all its jump segments on the Region Profiles tab

c) There are no special keywords in the ID's; e.g. naming a profile "jump" does not actually assign it to jump segments

Velocity Profile ID *May be integer or text*	Velocity (mm/s)	Mode	Laser-on delay (TV1), uSec	Laser-off delay (TV2), uSec	Jump delay (TL1), uSec	Mark delay (TL2), uSec	Polygon delay (TL3), uSec	
0	5000	Auto	100	100	100	50	30	Jump Profile. Fixed for OASIS Challenge.
1	1250	Auto	100	100	100	50	30	Hatch Profile. Fixed for OASIS Challenge.
2	1250	Auto	100	100	100	50	30	Contour Profile. Fixed for OASIS Challenge.

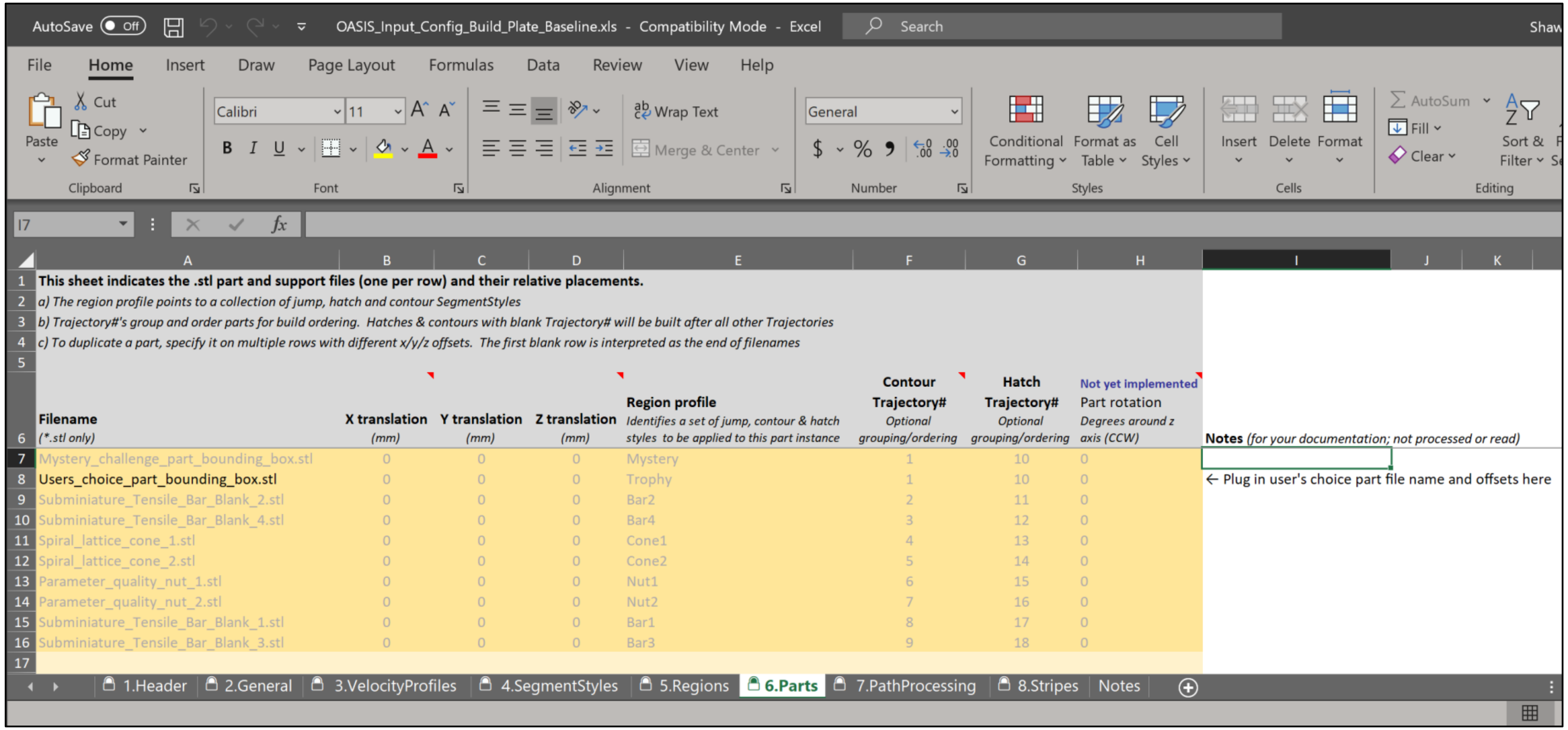
Fixed process parameters example

OASIS Open-source Additive Scanning Implementation Strategy

4. Complete the build parameter files

- Be sure to replace the “**Users_choice_part_bounding_box.stl**” (on Tab **6.Parts**) with your own trophy/medallion stl file (*including position offsets: X, Y, and Z*) in your “**OASIS_Input_Config_Build_Plate_<Organization Name>.xls**” file.

1. Download files and file folders from GitHub
2. Review all the documentation
3. Develop your code
4. **Complete the build parameter files**
5. Design your own trophy/medallion
6. Enter the contest
7. Upload submissions to private GitHub repository
8. Code evaluation
9. Top selected codes sent for print
10. Print evaluation
11. Announcement of winners



This sheet indicates the .stl part and support files (one per row) and their relative placements.

a) The region profile points to a collection of jump, hatch and contour SegmentStyles

b) Trajectory#'s group and order parts for build ordering. Hatches & contours with blank Trajectory# will be built after all other Trajectories

c) To duplicate a part, specify it on multiple rows with different x/y/z offsets. The first blank row is interpreted as the end of filenames

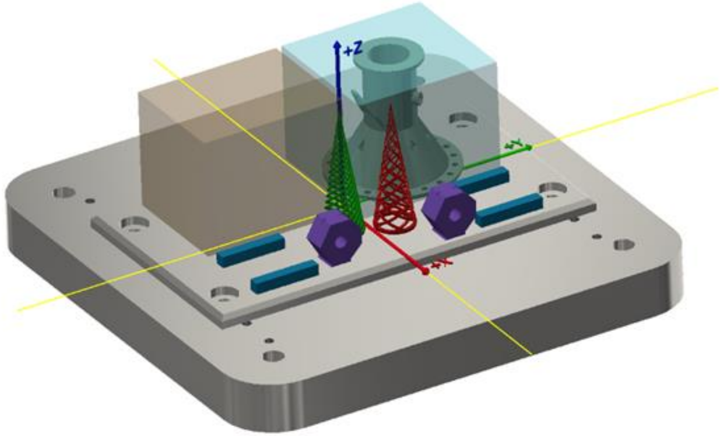
Filename (**.stl only*)	**X translation** (*mm*)	**Y translation** (*mm*)	**Z translation** (*mm*)	**Region profile** *Identifies a set of jump, contour & hatch styles to be applied to this part instance*	**Contour Trajectory#** *Optional grouping/ordering*	**Hatch Trajectory#** *Optional grouping/ordering*	Not yet implemented Part rotation *Degrees around z axis (CCW)*	**Notes** *(for your documentation; not processed or read)*
Mystery_challenge_part_bounding_box.stl	0	0	0	Mystery	1	10	0	
Users_choice_part_bounding_box.stl	0	0	0	Trophy	1	10	0	← Plug in user's choice part file name and offsets here
Subminiature_Tensile_Bar_Blank_2.stl	0	0	0	Bar2	2	11	0	
Subminiature_Tensile_Bar_Blank_4.stl	0	0	0	Bar4	3	12	0	
Spiral_lattice_cone_1.stl	0	0	0	Cone1	4	13	0	
Spiral_lattice_cone_2.stl	0	0	0	Cone2	5	14	0	
Parameter_quality_nut_1.stl	0	0	0	Nut1	6	15	0	
Parameter_quality_nut_2.stl	0	0	0	Nut2	7	16	0	
Subminiature_Tensile_Bar_Blank_1.stl	0	0	0	Bar1	8	17	0	
Subminiature_Tensile_Bar_Blank_3.stl	0	0	0	Bar3	9	18	0	

OASIS Open-source Additive Scanning Implementation Strategy

1. Download files and file folders from GitHub
2. Review all the documentation
3. Develop your code
4. Complete the build parameter files
5. **Design your own trophy/medallion**
6. Enter the contest
7. Upload submissions to private GitHub repository
8. Code evaluation
9. Top selected codes sent for print
10. Print evaluation
11. Announcement of winners

5. Design your own trophy/medallion

- Designed trophy/medallion submitted by the participant that will be printed, if the submitted code is selected for printing.
- File type: Binary or ASCII *.stl (*Please dimension your part in millimeters*)
- Max file size: 100 MB
- Volume bounding size: 60mm x 75mm x 60mm (X, Y, Z).
  - The part must be located inside of the highlighted volume shown in the drawing "**Users_choice_part_bounding_box.pdf**" (*note the center point*), which is included with the challenge geometry. This can be achieved by:
    1. exporting the STL file with the same coordinates as the build plate, or
    2. applying the X, Y and Z offsets to the STL file in the input configuration file.



Build Plate

5. Design your own trophy/medallion

1. Download files and file folders from GitHub
2. Review all the documentation
3. Develop your code
4. Complete the build parameter files
5. **Design your own trophy/medallion**
6. Enter the contest
7. Upload submissions to private GitHub repository
8. Code evaluation
9. Top selected codes sent for print
10. Print evaluation
11. Announcement of winners

- Consolidated volume (*including supports*) is limited to 30,000mm^3 (30cm^3).

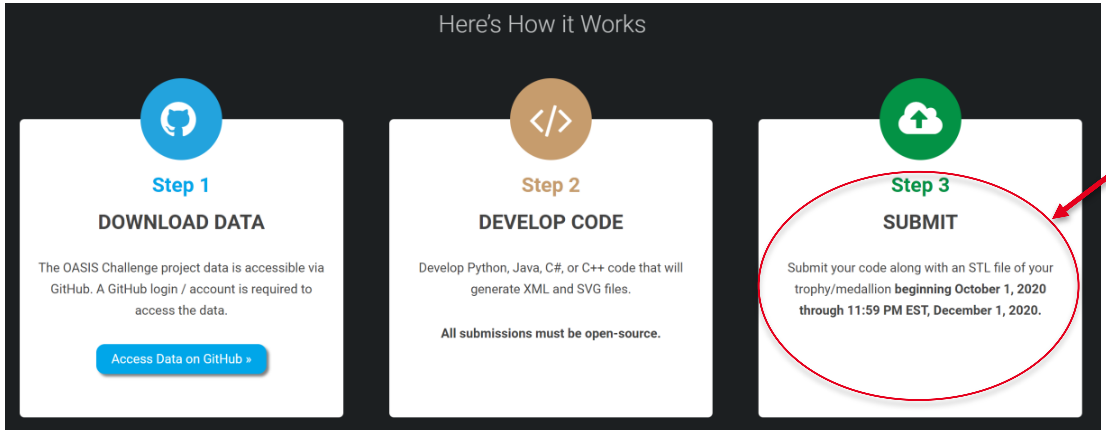
  Note: This is just over 10% of the bounding volume
- The designed trophy/medallion must have features (*i.e. holes*) to allow unfused powder to drain out, and supports, if required.
- There will be no post processing beyond stress relieving and cutting the part from the plate (*no support removal, no polishing, no heat treatment, etc.*)
- No additional modifications (*e.g. scaling, positioning, support geometries, repair, etc.*) to the stl file will be performed by the OASIS team. It is strongly recommended that participants confirm their stl files are well formatted (*water-tight, with no inverted normals*), and that their code will produce valid outputs for the submitted part.

  Note: If the designed trophy/medallion does not fit within the volume bounding size, or the consolidated volume is larger than specified above, it will not be printed

6. Enter the contest

1. Download files and file folders from GitHub
2. Review all the documentation
3. Develop your code
4. Complete the build parameter files
5. Design your own trophy/medallion
6. **Enter the contest**
7. Upload submissions to private GitHub repository
8. Code evaluation
9. Top selected codes sent for print
10. Print evaluation
11. Announcement of winners

- Go to the OASIS website at: https://americamakes.us/oasis
- Participants can enter the contest beginning **October 1st, 2020**.
- Deadline for submissions is **Tuesday, December 1st, 2020 at midnight**.
  - Submissions received after the deadline will not be considered.
- Participants will answer a few questions and then receive a link to a private GitHub repository where participants will submit your content.
- After the deadline, the repository will be switched from private to public and all participant access will be removed until judging is complete.



Here's How it Works

Step 1

DOWNLOAD DATA

The OASIS Challenge project data is accessible via GitHub. A GitHub login / account is required to access the data.

Step 2

DEVELOP CODE

Develop Python, Java, C#, or C++ code that will generate XML and SVG files.

All submissions must be open-source.

Step 3

SUBMIT

Submit your code along with an STL file of your trophy/medallion **beginning October 1, 2020 through 11:59 PM EST, December 1, 2020.**

Submit button will be added on October 1st, 2020.

Submit code and content beginning October 1, 2020

OASIS Open-source Additive Scanning Implementation Strategy

7. Upload submissions to private GitHub repository

- Update your GitHub repository with your code, trophy/medallion design, and completed build parameters files.
- One (1) submission per entry, must include:
  - **Submitted code**
  - **Code license**
  - **Script files** (*there are three (3) script files required per entry*)
  - **README file**
  - **Build parameter files** (*there are two (2) build files required per entry*)
  - **Trophy/medallion file**

1. Download files and file folders from GitHub
2. Review all the documentation
3. Develop your code
4. Complete the build parameter files
5. Design your own trophy/medallion
6. Enter the contest
7. **Upload submissions to private GitHub repository**
8. Code evaluation
9. Top selected codes sent for print
10. Print evaluation
11. Announcement of winners

Deadline is Tuesday, December 1st, 2020 at midnight

OASIS Open-source Additive Scanning Implementation Strategy

8. Code evaluation

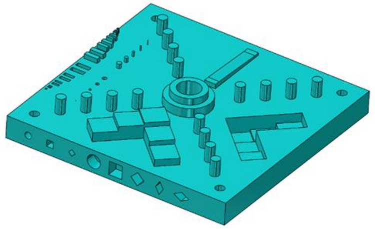
- Code evaluation conducted by the OASIS team.
- All ties will be decided by the “**Code Run Speed**”.

1. Download files and file folders from GitHub
2. Review all the documentation
3. Develop your code
4. Complete the build parameter files
5. Design your own trophy/medallion
6. Enter the contest
7. Upload submissions to private GitHub repository
8. **Code evaluation**
9. Top selected codes sent for print
10. Print evaluation
11. Announcement of winners

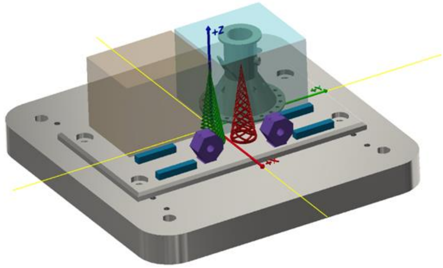
Submitted Code Evaluation (20% *of Total Score*)

Table #2

Review Elements	Description	Evaluation Tools	Score Percentage	Score Weighting		How Element is Scored	Comments
Code Quality	The following will be checked: - # of vulnerabilities - # of bugs - # of code smells - # of security hot spots	SonarQube	50%	Bugs	50%	Normalized, weighted score based on evaluation from SonarCloud.io using default rules and estimate of effort to remediate.	See SonarCloud.io and Table #3 below.
				Vulnerabilities	25%		
				Security Hot Spots	15%		
				Code Smell	10%		
Code Run Speed	Total time it takes to create valid XML and SVG files for: - NIST Artifact - Build Plate (w/o trophy/medallion)	Run time on a virtual machine utilizing Microsoft Server 2019 Operating System using Docker containers. Single core, single threaded.	50%	1		Normalized score based on all participant results.	No run times longer than 6 hours. Run times over 6 hours will be disqualified.



NIST Test Artifact



Build Plate

SonarCloud.io (Sonar)

Sonar cloud service, found at http://sonarcloud.io, will be used to assess the code quality of the submissions. This tool can be run on public GitHub repositories, so will not be performed until the deadline passes and the GitHub repository is switched from private to public.

OASIS Open-source Additive Scanning Implementation Strategy

8. Code evaluation

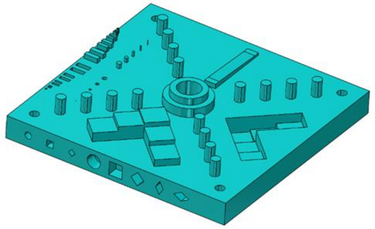
The rules for assessing the code can be found at https://rules.sonarsource.com/. Navigate to the appropriate language. **Note: Languages will have different rules which may impact the scoring of different projects**

Sonar's analysis includes an estimate of the amount of time required to remediate a given issue which is a proxy for the complexity of the remediation effort. These estimated remediation times will be used as a basis for calculating the score with a "**perfect**" entry having no issues. The classification of the issues will impact the weight applied to the time estimate. For instance, the time to remediate "**bugs**" will contribute to 50% of the code quality score while "**code smells**" will contribute 10%. Each category will be normalized by determining the worst case entered.

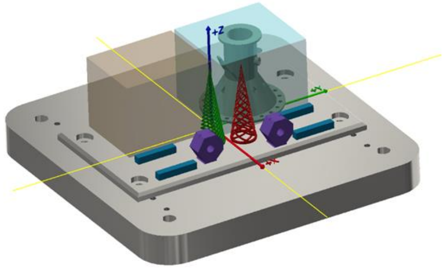
Code Quality – Example Scoring
Table #3

Project	Bugs (days)	Vulnerabilities (days)	Code Smell (days)	Security Hot Spots (days)	Total Score (%)
	50%	25%	10%	15%	
A	1.125	0.014	35.000	1.000	25.67%
B	1.208	0.001	2.083	0.001	49.12%
C	0.083	0.125	19.000	0.000	66.12%
D	0.013	0.000	9.000	0.125	95.01%
Perfection	0.000	0.000	0.000	0.000	100.00%

1. Download files and file folders from GitHub
2. Review all the documentation
3. Develop your code
4. Complete the build parameter files
5. Design your own trophy/medallion
6. Enter the contest
7. Upload submissions to private GitHub repository
8. **Code evaluation**
9. Top selected codes sent for print
10. Print evaluation
11. Announcement of winners



NIST Test Artifact



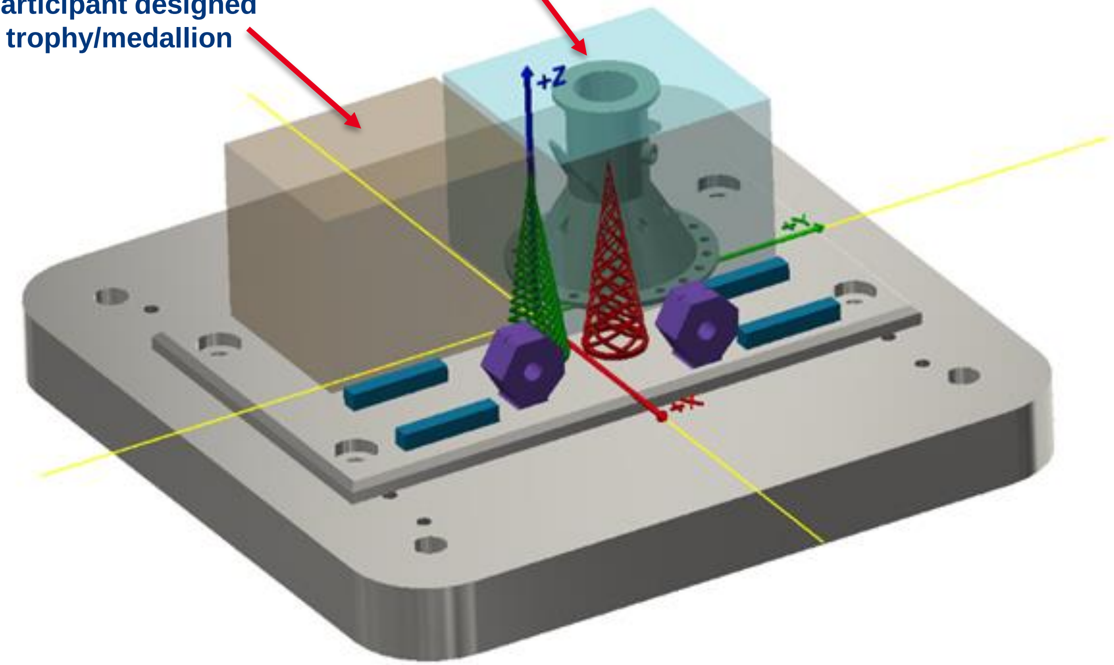
Build Plate

OASIS
Open-source Additive Scanning Implementation Strategy

9. Top selected codes sent for print

1. Download files and file folders from GitHub
2. Review all the documentation
3. Develop your code
4. Complete the build parameter files
5. Design your own trophy/medallion
6. Enter the contest
7. Upload submissions to private GitHub repository
8. Code evaluation
9. **Top selected codes sent for print**
10. Print evaluation
11. Announcement of winners

- Top selected codes will be sent to GEGR for the build plate to be printed using Ti-6Al-4V on the Concept Laser M2-Open Direct Metal Laser Melting (DMLM) machine.
- Print will include:
  - Four (4) tensile bars
  - Two (2) angled nuts
  - Two (2) lattice cones
  - Participant designed trophy/medallion
  - “**Mystery Part**”



“Mystery Part”

Participant designed trophy/medallion

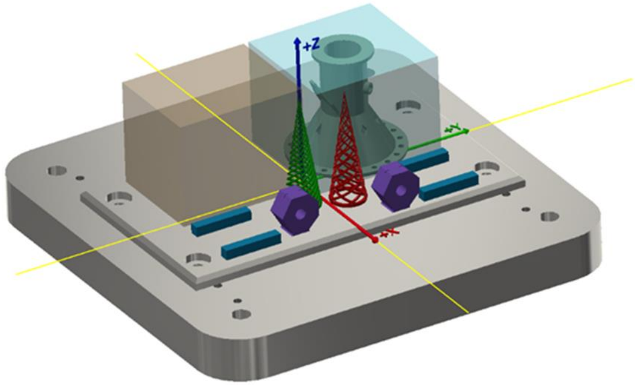
Build Plate

OASIS
Open-source Additive Scanning Implementation Strategy

9. Top selected codes sent for print

1. Download files and file folders from GitHub
2. Review all the documentation
3. Develop your code
4. Complete the build parameter files
5. Design your own trophy/medallion
6. Enter the contest
7. Upload submissions to private GitHub repository
8. Code evaluation
9. **Top selected codes sent for print**
10. Print evaluation
11. Announcement of winners

- The “**Mystery Part**” is a representative aerospace component, defined by the OASIS Team, composed of several geometric features including:
  - Thin walls
  - Thick to thin transitions
  - Various radii
  - Small overhanging features
  - Channels/holes of various geometries and in multiple orientations
- Volume bounding size: 70mm x 75mm x 60mm (X, Y, Z).
- “**Mystery Part**” information:
  - Height: 44.45 mm
  - Volume: 40,328 mm3
  - Surface Area: 60,878 mm2
- The performance of all code entries will be evaluated using the same “**Mystery Part**” along with the other geometries of the Build Plate. No further information regarding the “**Mystery Part**” will be provided until after judging is complete.



Build Plate

OASIS Open-source Additive Scanning Implementation Strategy

10. Print evaluation

1. Download files and file folders from GitHub
2. Review all the documentation
3. Develop your code
4. Complete the build parameter files
5. Design your own trophy/medallion
6. Enter the contest
7. Upload submissions to private GitHub repository
8. Code evaluation
9. Top selected codes sent for print
10. **Print evaluation**
11. Announcement of winners

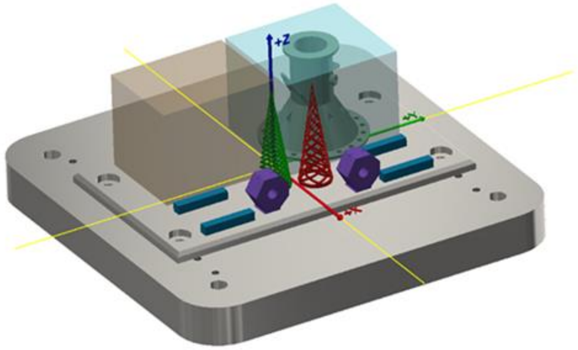
- Ti-6Al-4V print evaluation conducted by the OASIS team.

  Note: Build plate will be stress relieved and parts cut from plate using wire EDM. No further processing will be done except as specified in Table #4

Ti-6Al-4V Print Evaluation (*80% of Total Score*)
Table #4

Angled Nuts (30% *of Total Score*)
****Note: One (1) nut will be left whole / one (1) nut will be cross sectioned at notches, mounted, and polished****

Review Elements	Description	Units	Evaluation Tools	Score Percentage	Score Weighting	How Element is Scored	Comments
Surface Finish	24 Measurements - 1 un-sectioned nut - 8 total surfaces - 3 patches per surface - does not include the “cut” mounted surface - does not include the hole	microns	Alicona Profilometer (S_a)	25%	1	Score (%) = $100*(1-(avgS_a-minS_a)/minS_a)$	1 whole nut per build plate. Average S_a will be used in scoring.
Geometric Fidelity	Un-Sectioned Nut - All 8 surfaces - Hole size Sectioned Nut - Hole size	mm	Optical CMM (Nikon touchless point cloud scanner)	35%	1	Score (%) = $100*(1-(dev_{avg}-dev_{min})/dev_{min})$	Position of EDM cut to remove the nuts from the build plate will not affect the Geometric Fidelity. Average deviation will be used in scoring.
Porosity	Cross-section one of the nuts then mount, polish, and photograph the entire cross-section. Bulk porosity, near surface porosity (250 microns from the surface), and total porosity will be recorded. Any pores smaller than 5 microns will be ignored.	microns	Optical microscopy, ImageJ	40%	1	Score (%) = $100*(1-porosity_{avg}-0.9)/(1-porosity_{min}-0.9)$	1 cross-sectioned nut per build plate. Average porosity will be used in scoring. Upper porosity limit is 10%. More than 10% will receive a score of 0.



Build Plate

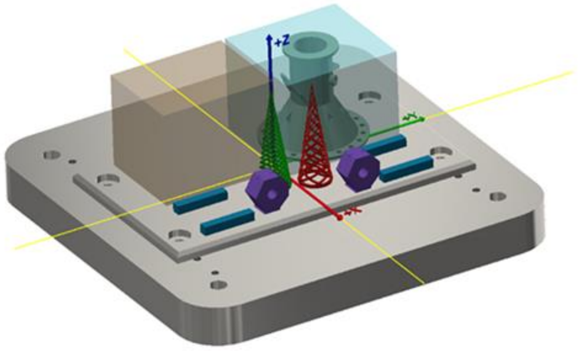
OASIS
Open-source Additive Scanning Implementation Strategy

10. Print evaluation

- Ti-6Al-4V print evaluation conducted by the OASIS team.

1. Download files and file folders from GitHub
2. Review all the documentation
3. Develop your code
4. Complete the build parameter files
5. Design your own trophy/medallion
6. Enter the contest
7. Upload submissions to private GitHub repository
8. Code evaluation
9. Top selected codes sent for print
10. **Print evaluation**
11. Announcement of winners

Ti-6Al-4V Print Evaluation (*80% of Total Score*) **Table #4 continued**							
Tensile Bars (*30% of Total Score*) ****Note: All 4 tensile bars will be machined into miniature tensile specimens and tested****							
Review Elements	**Description**	**Units**	**Evaluation Tools**	**Score Percentage**	**Score Weighting**	**How Element is Scored**	**Comments**
Ultimate Tensile Strength (UTS)	Miniature tensile specimens will be machined for this test. - surface finish = 32 µin	MPa	Instron Tensile Tester	50%	1	Score (%) = $100*(UTS_{avg}/UTS_{max})$	Average UTS of the best 3 out of 4 results will be used in scoring.
Strain at UTS	Miniature tensile specimens will be machined for this test. - surface finish = 32 µin	mm/ mm	Instron Tensile Tester	50%	1	Score (%) = $100*(EPS_{avg}/EPS_{max})$	Average EPS from the best 3 out of 4 results will be used in scoring.
Lattice Cones (*20% of Total Score*)							
Review Elements	**Description**	**Units**	**Evaluation Tools**	**Score Percentage**	**Score Weighting**	**How Element is Scored**	**Comments**
Height Lattice Cone 1	Cone height until print failure of first lattice.	mm	Calipers	50%	1	Score (%) = $100*(height/height_{max})$	Height will be measured ***prior*** to removal from build plate.
Height Lattice Cone 2	Cone height until print failure of first lattice.	mm	Calipers	50%	1	Score (%) = $100*(height/height_{max})$	Height will be measured ***prior*** to removal from build plate.



Build Plate

OASIS Open-source Additive Scanning Implementation Strategy

11. Announcement of winners

1. Download files and file folders from GitHub
2. Review all the documentation
3. Develop your code
4. Complete the build parameter files
5. Design your own trophy/medallion
6. Enter the contest
7. Upload submissions to private GitHub repository
8. Code evaluation
9. Top selected codes sent for print
10. Print evaluation
11. **Announcement of winners**

- OASIS Challenge award winners anticipated notification date is Friday, June 4th, 2021.
- A workshop will be held for the winners to showcase their code.

Note: The award checks will be written to the America Makes member organization, not to the individual. No award money under this challenge can be paid to non-U.S. citizens from China, Iran, Iraq, Syria, Sudan, North Korea, Cuba, or Venezuela.

Awards

1st Place: $25,500

2nd Place: $15,500

3rd Place: $10,500

Honorable Mention 1: $5,500

Honorable Mention 2: $5,500

Honorable Mention 3: $5,500

OASIS Open-source Additive Scanning Implementation Strategy

Eligibility

Domestic and foreign members who are in good standing with America Makes are eligible to participate. Only the lead proposer must be a member of America Makes. Participants of the lead proposer's team may be made up of America Makes members and non-members.

If the lead proposing organization is not currently an America Makes member, the organization must become one by December 1st, 2020. Information on how to join America Makes is available on the **membership page**. **Note: The foreign membership approval process requires a minimum of three (3) weeks to complete**

Members associated with the evaluation panel are not eligible. This includes participants employed by NCDMM, AFRL, and GE.

FAQ and Challenge Clarifications

A list of frequently asked questions and clarifications regarding the challenge will be compiled and updated periodically on the challenge website. Deadline for questions is November 1st, 2020.

Questions should be submitted to **challenge@americamakes.us**.

OASIS Open-source Additive Scanning Implementation Strategy

Reasons for disqualification

Below are reasons for disqualification from the OASIS Challenge…

1. Participant is not a member of America Makes by Tuesday, December 1st, 2020.
2. Not all the required files are submitted by the Tuesday, December 1st, 2020 deadline.
3. Submitted code is private and not open source.
4. Submitted code relies on external libraries to perform the key elements of the challenge.
5. Submitted code is not written in one (1) of the four (4) specified code languages.
6. Submitted code cannot be compiled.
7. Submitted code does not create valid XML and SVG files.
8. Submitted code takes longer than six (6) hours to run or crashes during evaluation (*with and without the trophy*).
9. Created files (*XML/SVG*) causes the print machine to “crash” (*including the trophy*)
10. Submitted code is identified as malicious software.
11. Proper license is not associated with the submitted code.

Questions??

Are You Up for the Challenge?