

# Open Source Simulation Leveraging Open Source Technologies for Rapid Prototyping of Simulation Tools

*Presented by:* Eduardo Fonseca

October 2016

SGT-INC.COM





- Soyuz Docking to ISS Simulator
  - Built with Python and Blender
- Features
  - Open Source Tools
  - Rapid Prototyping





- Soyuz Docking is one of the most dynamic phases of flight in ISS operations
  - Training Flight Controllers is critical for Mission Success and Crew Safety
  - Human Space Flight involves high risk
- Currently Soyuz Docking is not a phase of flight that is trained in an integrated environment due to lack of adequate simulation tools







- International Space Station (ISS) EVA Suit Water Intrusion High Visibility Close Call IRIS Case Number: S-2013-199-00005 Recommendations
  - <u>R17:</u> The ISS Program should ensure that FMEA/CILs related to fastpaced failure scenarios (visiting vehicles, on-board emergency response, software transition issues, and serious system hardware failures) are regularly updated, studied, and used in training for flight controllers as well as engineering and safety personnel.
  - <u>R18:</u> As the success of the ISS Program continues, the ISS Program must institute requirements and behaviors that combat the tendency towards complacency by requiring regular training by all teams in the safety critical aspects of failures related to fastpaced scenarios (visiting vehicles, on-board emergency response, software transition issues, and serious system hardware failures).





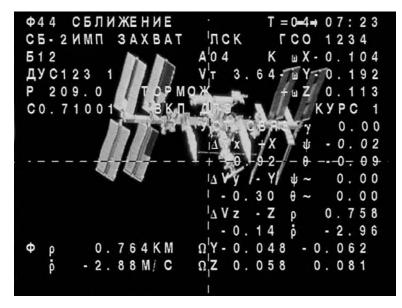
- Dynamic phases of flight that are trained in integrated environment
  - Reboost/Debris Avoidance Maneuver
  - HTV/Dragon/Cygnus Rendezvous, Capture and Berthing
  - EVA
  - Major Failures
  - Emergencies

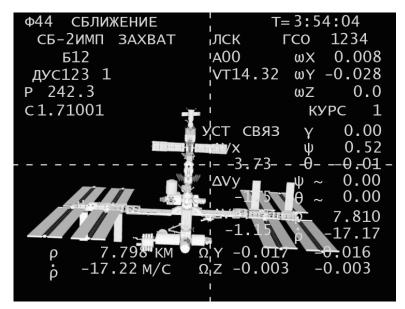
- Soyuz/Progress Docking/Undocking





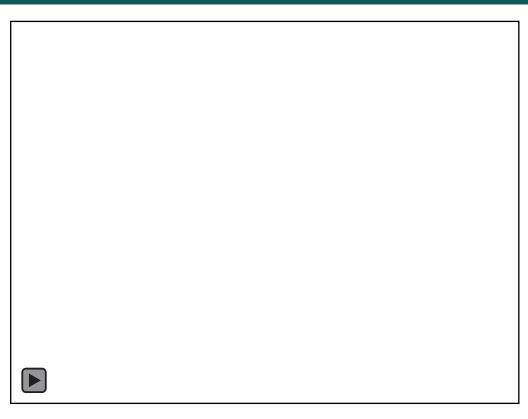
• A simulation tool to bridge the gap between current simulation capabilities/limitations













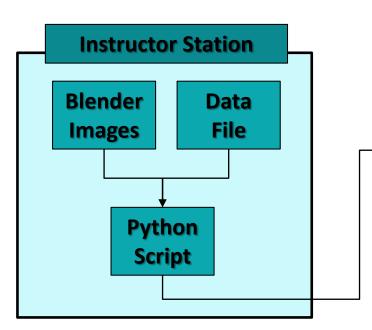


- Tools Used
  - Blender: 3D modeling
  - Excel: Data Manipulation
  - Python: Integration of data and 3D models





• Tool Structure



### Mission Control Flight Control Room







#### Delivery Excellence Program SGT Proprietary Information

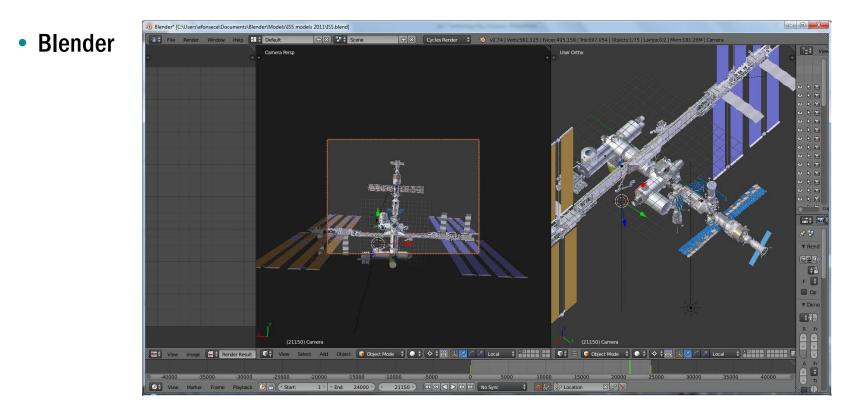




- Blender
  - Models from NASA 3D Resources (public): <u>https://nasa3d.arc.nasa.gov/</u>
  - Instructors determine the camera views
  - Model / camera / lighting/ flight path created in Blender
  - Rendezvous profile scene rendered
    - Each image requires 30 seconds of render time
    - 40 minutes requires 2400 frames at 1 frame / second
    - Total rendering time approximately 20 hours per machine
  - A text file with location of images is generated by user







#### Delivery Excellence Program SGT Proprietary Information



• Excel Data Manipulation

Trajectory

Status

3:54:00 СБ-2ИМП

3:54:01 СБ-2ИМП

3:54:02 СБ-2ИМП

3:54:03 СБ-2ИМП

0

20

20

20

20

20

20

20

20

Frame

Kurs

Status

**3AXBAT** 

**3AXBAT** 

**3AXBAT** 

**3AXBAT** 

7.423

7.172

6.925

6.681

6.44

6.202

5.966

5.732

5.499

DIST

0

20

40

60

80

100

120

140

160

- VVO provides data that drives the display
- Data manipulated to be Blender and Python readable

ACS

thruster

Status manifold

1234 512

1234 512

1234 512

1234 512

-4.31

-4.326

-4.328

-4.317

-4.292

-4.254

-4.204

-4.142

-4.068

2 002

Y ISS

-0.515

-0.481

-0.459

-0.434

-0.406

-0.374

-0.34

-0.303

0.064

-0.5

ZIS

X ISS

Attitude

лск

лск

ЛСК

лск

-12.66

-12.465

-12.291

-12.138

-12.003

-11.885

-11.782

-11.694

-11.617

11 661

VIPV

	the der a	•				4 ( rajectory S (6) (10) (10) (12) Resourc (14)	e Fuel	(3) Ku (7) Ku	rs Status Irs Mode (13) Syst		(4) Attitu 3) Accel Sta (11) euver		= h h F C O ( w X w Y w Z	9) Ange ± ±		Statu	
					C	,					17			<u>к</u> у (18) Ku	P ( Irs-A A		
										( (	т		g			,	
7	8	9a	9b	9c	10	11	12	1		14		(19)					
urs Mo		Angular	Angular	Angular			Resou			Nav C-	v x ± x		У			,	
	status	Rates (X)	Rates (Y)	Rates (Z)			Fue	I Sta		word status	20) Burn De						λ:
	A00	0.008	-0.017	-0.008	ДУС123 1	VT14.32	24	2.3		1.71001	,	,	đ			,	
	A00	0.008	-0.017		ДУС123 1	VT14.32	24	2.3		1.71001		(19)	(2	21) Kurs	-P Ang	gles	
	A00	0.008	-0.017	-0.008	ДУС123 1	VT14.32	24	2.3		1.71001	V Y	±Υ	y ±			,	
	A00	0.008	-0.028	0.000	ДУС123 1	VT14.32	24	2.3		1.71001	(20) Rum	Nalta Vi	•				
SS	SA	HLA	AZ					dINRT		X blendery	blenderz					,	
6.02			352.06			63.02	117.41	-7.55						Ra	inge &	Rang	ge Ra
5.699		-32.8	351.83	-52.76		63.44	116.99	-7.57				Bl	enderize		,		
5.384			351.58		27.17	63.84	116.56	-7.55									
5.079		-29.65	351.31	-49.62		64.22	116.14	-7.5									Ť
4.782	2 109.45	-28.11	351.01	-48.1	26.74	64.57	115.72	-7.42	-1230.						28)	<u></u>	ii
4.493		-26.6	350.69	-46.59	26.48	64.89	115.3	-7.3	-1219.4	48 -116.38	7 1288.567						
4.216		-25.12				65.18	114.88	-7.14		15 -107.21	3 1208.587						
3.94						65.44	114.45	-6.94	-1187.	37 -97.466	7 1131.473						
3.688		-22.24	349.51	-42.27	25.47	65.68	114.03	-6.68									
2 / 29	107.24	20.86	2/10/02	10.90		65.99	112.6	6.27	11/11	70 75 6							

: H N

time (real)

0:54:00

0:54:01

0:54:02

0:54:03

87

87

87

87

87

87

87

87

87

07

IME

time (SGMT)

0:54:04

0:54:24

0:54:44

0:55:04

0:55:24

0:55:44

0:56:04

0:56:24

0:56:44

0.57.04

0:54:00

0:54:01

0:54:02

0:54:03

0:00:00

0:00:20

0:00:20

0:00:20

0:00:20

0:00:20

0:00:20

0:00:20

0:00:20

0.00.00

Time

#### Delivery Excellence Program SGT Proprietary Information

A :



Blender* [C:\Users\efonseca\Documents\Ble	ender\Models\JSS models 2011\JSS_081616.blend]	
👀 🕈 File Render Window Help 📘	🗘 Default 🕂 🛠 🔀 🚼 Scene 2 🕂 🗙 Cycles Render 🕴 🔌 v2.74   Verts:558,717   Faces:493,416   Tris:69	6,040   Objects:1/78   Lamps:0/2   Mem:155.62M   Sphere
view Transform User Ortho	import bpy import math as np	T_+ View Search All Scenes +
Translate Rotate Scale	<pre>scene = by.context.scene cube = scene.object5(0)</pre>	♥ Up RenderLayers   Ø) ● World
Mirror	<pre>scene.frame_set(0)</pre>	L Empty.001   ∀ ∀ ∀ ∀ ∀ ∀ ♥ ♥ № №
v Edit	<pre>cube.location = (-1235.5333333333.147.6333333333.1726.02) cube.keyframe_insert(data_path= "location" , index=-1)</pre>	● Ø Animation   ** ● - ₩ Camera ● ▶ 151
Duplicate Duplicate Linked	<pre>scene.frame_set(20) cube.location = (-1240.12,-143.33333333333333333333333333333333333</pre>	● Ø Animation   % @E Camera
Join	scene.frame_set(40)	∀ Sphere   4 <sup>°</sup> ∀ • № 101 9 Sun.002   ⊛ • № 101
Set Origin	<pre>cube.location = (-1240.6933333333.137.886666666667.1543.41333333333) cube.keyframe_insert(data_path= "location" , index=-1)</pre>	
Smooth Flat Data Transfer:	<pre>scene.frame_set(60) cube.location = (-1227.54,-131.58,1455.98) cube.keyframe_insert(data_path= "location" , index=-1)</pre>	
Data Data Layo ► History	<pre>scene.frame_set(80) cube.location = (-1230.3733333333124.41333333333.1370.84) cube.keyframe insert(data path= "location", index=-1)</pre>	
	scene.frame_set(100) cube.location = (-1219.48,-116.386666666667,1288.56666666667)	
	<pre>cube.keyframe_insert(data_path= "location" , index=-1) scene.frame_set(120)</pre>	x 3 Scene ▼ Render
	cube.location = (-1205.l4666666667,-107.21333333333,1208.586666666667) cube.keyframe_insert(data_path= "location" , index=-1)	🐻 Render 🚰 Animation 🕼 Audio
	<pre>scene.frame_set(140) cube.location = (-1187.37333333333.97.4666666666666667,1131.473333333333)</pre>	Display: Image Editor 💠 🛅 Feature Set: Supported 🔷
Operator	<pre>cube.keyframe_insert(data_path= "location" , index=-1) scene.frame_set(160)</pre>	Open Shading Language     Dimensions
	<pre>cube.location = (-1166.16,-86.86,1057.2266666667) cube.keyframe_insert(data_path= "location" , index=-1)</pre>	Render Presets + +
××	<pre>scene.frame_set(180) cube.location = (-1141.793333333375.68,985.56) 1) Schem</pre>	Resolution: Frame Range:
	I) spiele Object Mo, 플라 View Text: Edit: Format Templates 플라 Soyuz.py - 슈 맨 왕 프 램 왕 Run Script : Register T	
800 -500 -400 -200 0 - 200	400 600 800 1000 1200 1400 1600 1800 2000 2200 2400 2600 2800 3000 3200 3400 3600 3800 4000 4200 4400 4	Aspect Ratio: Frame Rate:
Chew Marker Frame Playback		3200         3200         3400         (Y:         1.000)         Time Remapping:           32         30         Border         Crop         <0:100 * N: 100 *)

#### Delivery Excellence Program SGT Proprietary Information





- Python
  - Reads text file with location of Blender images
  - Reads text file with VVO data
  - Loop:
    - Displays specified image from table
    - Displays specified telemetry from table
    - Iterate







- Python
  - Libraries:
    - NumPy
    - Pandas
    - Matplotlib











#### • Python

67	# For loop that runs for the duration equal to the size of the arrav.
	# For 40 minute video it should be about 2400 loops
	while True:
70	<pre>img = imread(cbook.get_sample_data(frame.FILE[i+j])) # Reads the respective image</pre>
71	plt.clf() # Clears the screen
72	plt.imshow(img, cmap = cm.Greys_r, zorder=0, extent=[0.5, 8.0, 1.0, 7.0]) # Show the image
73	plt.xlim(0.5.8) # Set the x-limits on the display
74	plt.ylim(1,7) # Set the y-limits on the display
75	plt.plot(xhudx,yhudx,"", color="#ffffff", dashes=(11, 14), linewidth=2) # Plot the horizontal guide
76	plt.plot(xhudy,yhudy,"", color="#ffffff", dashes=(11, 14), linewidth=2) # Plot the vertical guide
77	plt.text(0.75,6.70, $\mathbf{u}'\mathbf{\Phi}'$ + '44', fontdict=font) # Add text
78	plt.text(1.66.6.70.u'CD/UKEH/IE', fontdict=font) # Add text
79	plt.text(5.55,6.70,'T=', fontdict=font)
80	plt.text(6.00,6.70,str(rndz.a1[i]), fontdict=font) # Add text
81	plt.text(2.28,6.34,str(rndz.a2[i]), fontdict=font, horizontalalignment='right') # Add text
82	plt.text(2.58,6.34,str(rndz,a3[i]), fontdict=font) # Add text
83	plt.text(4.40,6.34,str(rndz.a4[i]), fontdict=font) # Add text
84	plt.text(5.78,6.34,u'FCO', fontdict=font) # Add text
85	plt.text(6.72,6.34,str(rndz.a5[i]), fontdict=font) # Add text
86	plt.text(2.00,5.97,str(rndz.a6[i]), fontdict=font, horizontalalignment='right') # Add text
87	if str(rndz.a7[i]) != '.':
88	plt.text(3.65,5.97,str(rndz.a7[i]), fontdict=font, horizontalalignment='right') # Add text
89	plt.text(4.40,5.97,str(rndz.a8[i]), fontdict=font, horizontalalignment='left') # Add text
90	<pre>plt.text(6.00, 5.97,u'w', fontdict=font) # Add text</pre>
91	plt.text(6.00, 5.61,u'w', fontdict=font) # Add text
92	plt.text(6.00, 5.25,u'w', fontdict=font) # Add text
93	plt.text(6.22, 5.97,'X', fontdict=font) # Add text
94	<pre>plt.text(6.22, 5.61,'Y', fontdict=font) # Add text</pre>
95	<pre>plt.text(6.22, 5.25,'Z', fontdict=font) # Add text</pre>
96	plt.text(7.69,5.97,str("%.3f" % rndz.a9a[i]), fontdict=font, horizontalalignment='right') # Add text
97	<pre>plt.text(7.69,5.61,str("%.3f" % rndz.a9b[i]), fontdict=font, horizontalalignment='right') # Add text</pre>
98	plt.text(7.69,5.25,str(rndz.a9c[i]), fontdict=font, horizontalalignment='right') # Add text
99	plt.text(2.00,5.61,str(rndz.a10[i]), fontdict=font, horizontalalignment='right') # Add text
100	<pre>plt.text(2.42,5.61,'1', fontdict=font, horizontalalignment='right') # Add text</pre>
101	plt.text(4.40,5.61,str(rndz.a11[i]), fontdict=font, horizontalalignment='left') # Add text
102	<pre>plt.text(0.75,5.25,'P', fontdict=font, horizontalalignment='left') # Add text</pre>
103	plt.text(1.21,5.25,str(rndz.a12[i]), fontdict=font, horizontalalignment='left') # Add text







- Logistics
  - Graphics and telemetry cannot be generated in real time (currently)
  - Number of rendering workstations reduce rendering time
- Cost
  - Open Source (no cost)
  - 10 hours of prototype development





- Allow the capability to render 3D model and data in real time
- Investigate integration to existing simulators (Space Station Training Facility / TS21 Simulator)
- Train users on tool operations and improvements
- Apply to other simulators / programs / customers
  - Axiom
  - Boeing
  - Sierra Nevada
  - SpaceX
  - ...
- This is one of many possible solutions to solve the problem





### *Eduardo Fonseca* Email: efonseca@sgt-inc.com Phone: 281-244-5507 (0) | 281-467-2219 (M)