Flex Robotics is a company that has been writing software on Linux. It has a history of creating robotic models that can provide key functionality to different work environments. In 2019 it foreshadowed in the DARPA robotics cave missions. It learned about providing key information to responders in difficult environments by identifying objects. The mission in this proposal is to make deep sea environments more accessible to scientists by using autonomous robots.

Flex Robotics is proposing to employ several methods to accomplish this theme. The first goal in the mission is navigation. Establishing odometry, IMU, and a CNN during the first project month for one submersible. The submersible will be able to use the parameters for teleoperation and build a map of the environment for geolocational future planning The software developer will work on odometry and set up three course mappings that will simulate a location game using AMCL localization with the result of being able to navigate the three test environments. The second objective is setting up the IMU sensor which will aid in the telemetry of the robot. The third objective is the CNN neural net which will be trained on way points for navigation.

Goal two will deal with Knowledge based long-term autonomy by employing an Algorithm during the second project month. The result will focus on how the robot will function long-term. Not pre-programmed to handle unexpected situations. The objective of this goal will be to develop an algorithm that will guide the robot through the course. It will employee the first goal's map and pick waypoints to navigate it.

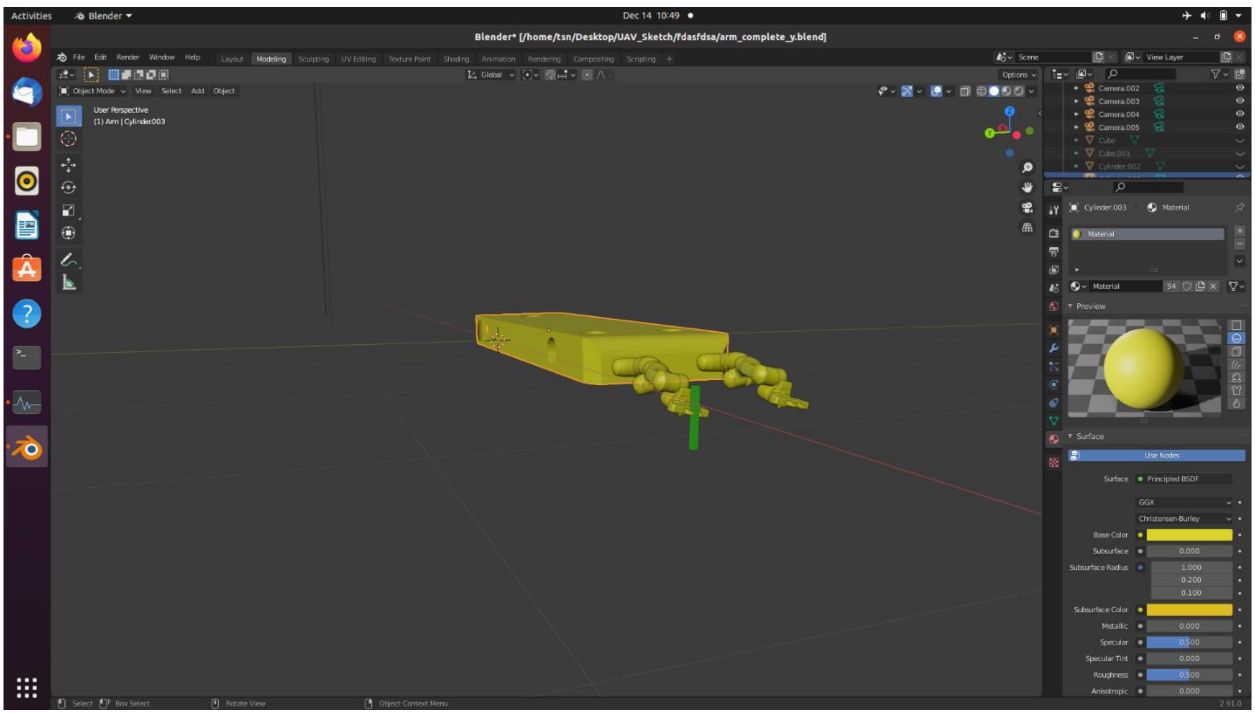
The third goal will deal with behavioral control and simulation. The approach is to create a course using Blender for object simulation and Gazebo for the physics environment. It will take place on month three. The result will be one submersible, two robot arms, three objects for arm simulation, three objects for coordinate location, and an underwater environment. The design engineer will develop 3D simulation for one submersible, course terrain, water ocean, 3 grabbing objects for robot arm, and underwater docking port. The objects will be created in Blender and exported as urdf, xacro, and sdf format. The result will be an environment as close to real ocean terrain as possible.

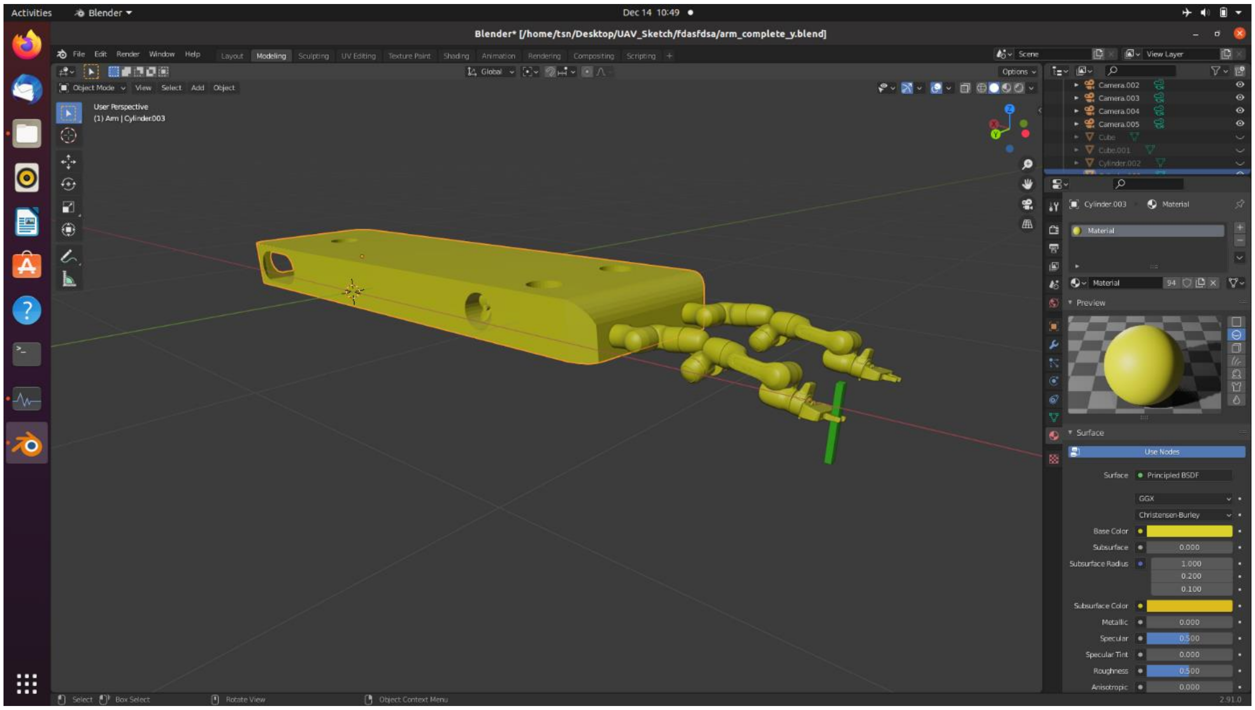
The fourth goal is hardware. Create custom circuit board, frame, motor mounting brackets, battery, tray, propellers, and arms. It will take place on month four and last one month. It is for one submersible. The result will be a submersible with components to fulfill the mission. The Design Engineer will create a PCB board for four motors, camera, steering, IMU sensor. Design custom board with Arm processor using KiCad software. The board will be quoted via gerber file and printed through jlpcb or pcway. The next objective is designing the stl model mesh files for one submersible. The files can be created in either Solid Works or Blender. The stl files will be sent to LexCent for printing the plastic mold which can be shipped. The result will be a model ready for motors, board, and battery installation.

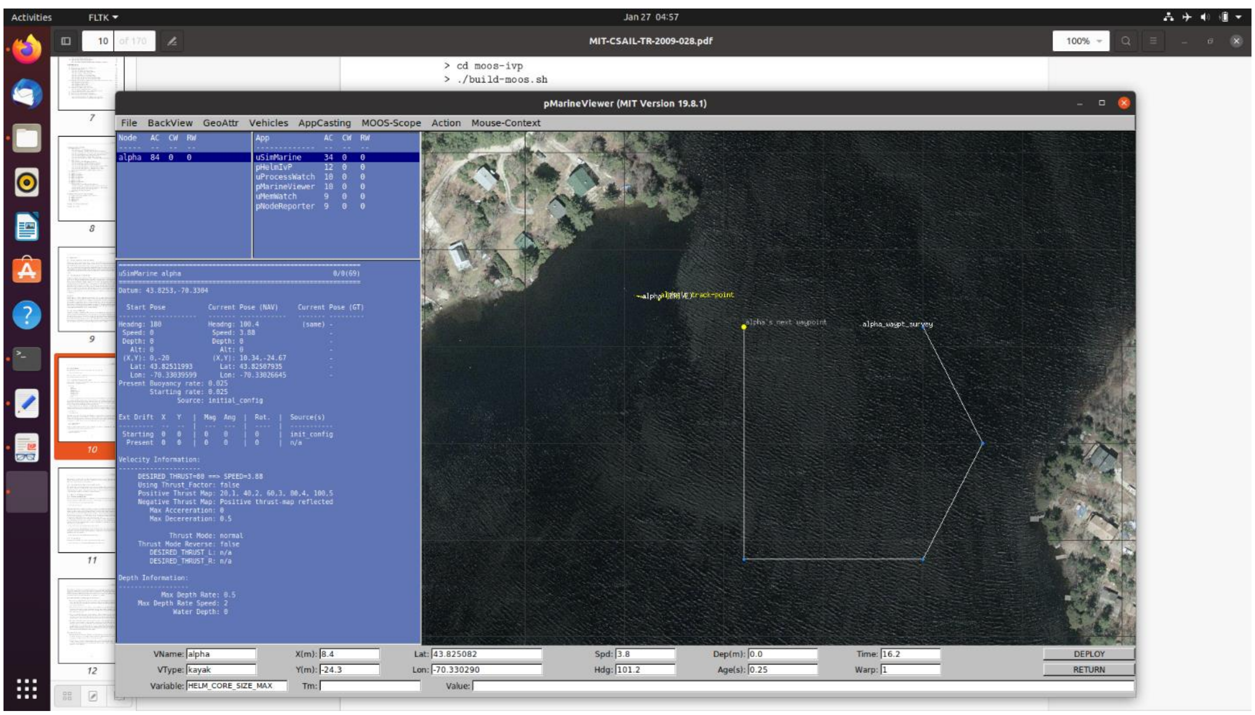
Goal five is robot control. The approach is an off the shelf controller with tether wire for preprogramming and system testing. It will take place on month five and last one month. The result is a submersible that can be dropped into a pool testbed for real world simulation. The Software Developer will create two robot arms. The kinematics for the arms will be tested on real objects. Moveit will be used to program the joints in the physics simulator. Scripts will be tested for automatic object detection and procedures. The benefit is that scientific analysis can be done on different objects. Command and control center can further pin point specific instructions for each mission. Joystick commands will be programmed for a Logitect or Xbox controller. The robot will be first tested in Gazebo and then later tested on the submersible in the testbed pool.

Goal six is system design. The approach is to create a docking system for the robot to upload data and recharge. It will take place on month six and last for one month. It is for one submersible and a team of scientists. The result is the submersible will be able to recharge, upload/download data through antenna connected to the docking station and communicate to scientists that are stationed in a lab desktop computer environment through a Wi-Fi or satellite communication depending on the vender's preference and scaling approach. The first objective will be done by the Software Developer and it will be to connect Gazebo to a live robot using bag data for the submersible. The robot will get instructions from the simulation environment and do the commands in live test pool. The second objective is to create a docking station to download/upload data and charge. The robot will be programmed to do the docking maneuver with the result of being able to find its way back to the docking port.

Goal seven is sustained interaction and learning. It will be done in a testbed environment where something is built like a structure and then sent to robot to do a task. It will last for one month and be for one submersible. The robot will get a simulated task and then carry out that procedure on a real object. The first objective to this goal will be carried out by a Software Developer using a Ontological database. The submersible will have Java like objects in the code. Each one will be an instruction. The result is the robot will learn to do the instructions. Picture building a house in the physics simulator then giving those commands to the robot. Bag data recorded would not work because of small changes in the actual environment. The advantage of doing things this way instead of a live wire type build is that this can be done repeatedly.




Activities
Blender
Dec 14 10:49
Blender* [/home/tsn/Desktop/UAV_Sketch/fdasfdsa/arm_complete_y.blend]
User Perspective
(1) Arm | Cylinder.003


Activities
FLTK
Jan 27 04:57
MIT-CSAIL-TR-2009-028.pdf
> cd moos-ivp
> ./build-moos.sh
pMarineViewer (MIT Version 19.8.1)
File BackView GeoAttr Vehicles AppCasting MOOS-Scope Action Mouse-Context
DEPLOY
RETURN