



Milestone 2

INTELLIGENT GROUND VEHICLE COMPETITION

Milestone Two Task Matrix

#	Task	Percentage	Will	Adam	Chris	Brent	To do
1	Finished Navigation GUI	100	0	0	0	100	none
2	Prototype Navigation	100	0	30	0	70	none
3	RabbitMQ C++ Clients	80	80	0	20	0	debug
4	Line following	66	0	0	100	0	final implementation
5	Prototype Control and IOP	20	20	80	0	0	get document
6	Performance Testing ZED	100	0	0	100	0	none
7	Evaluate INS Driver	100	50	0	50	0	none

GUI & Navigation

- ▶ SBMPC already implemented
- ▶ GUI for building, running tests created
 - ▶ Current: place rectangular obstacles, vehicle, goal. Plots found path
 - ▶ Future: Metrics on time and space usage
- ▶ Next steps:
 - ▶ Continued Testing
 - ▶ Optimize SBMPC
 - ▶ Extend GUI for entire software system

Performance Testing & Line Following

- ▶ Edge detection using OpenCV
- ▶ Frame rate and performance of ZED tested
- ▶ GPU based algorithms avoiding CPU bottleneck
- ▶ Next Steps:
 - ▶ Converting edges to line obstacles
 - ▶ Overlaying color and depth data to deal with inclinations

Communication Framework

- ▶ Subscriber event loop with callbacks
- ▶ Independent publisher
- ▶ Ready to start coding subscriber and publisher clients for components of software
- ▶ Working on API of publishable requests and responses in C++ and Java
- ▶ Need to work on Unicode in C++

IOP Challenge

- ▶ Emailed Matthew Skalny (Primary IOP Judge)
- ▶ Considering the following documents:
 - ▶ AS5669A
 - ▶ AS5710A
 - ▶ AS6009
 - ▶ AS5684B
- ▶ Starting work on what we know is needed
 - ▶ Basic UDP client
 - ▶ Custom UDP packet headers

Milestone Three Task Matrix

#	Task	Will	Adam	Chris	Brent
1	Finished GUI	15	15	0	70
2	Optimized Navigation Algorithm	0	30	0	70
3	RabbitMQ Clients for each software component	100	0	0	0
4	Finished Line Following	0	0	100	0
5	LIDAR Integration	25	0	75	0
6	IOP Test Client	0	100	0	0
7	IOP Nav Platform	0	100	0	0
8	Control Component	50	50	0	0
9	Integration Testing MQ Clients and IOP	50	50	0	0
10	Integration Testing Components	25	25	25	25