1. N/A

2. Algorithms:

I plan to use Long Short Term Memory neural networks in my project. They seem to be the best choice for composing music. They can learn from real music instead of musical rules like with plain reinforcement learning, and this is important because music varies far beyond the rules we have created for it. Also, LSTMs have a sense of time unlike feedforward networks, and can uncover global musical structures (longer than one time step) unlike regular recurrent neural networks. In the blog post from [1], Johnson explains using a stack of identical LSTMs for each note, which in a sense is multiple agents since it's multiple networks working together. But I want to go further with the multi-agent setup. I will use a set of MIDI files which have different voices like guitar and bass, and use a separate LSTM stack to output the separate voices. I would like to see the differences in the results with communication (using the other voices as inputs) and without (not using the other voices as inputs). One could use a central stack for the entire composition, but I think separating them will help them specialize their musical roles better.

The LSTM in [1] works by having a network for each note. The inputs are the MIDI note value, the pitch class, whether the surrounding notes were played previously, the previously played pitchclasses, and the beat. The first stack of hidden layers are recurrent in time. The second stack of hidden layers connects nearby notes' networks. Then the play probability and articulation probability are outputted. I think including inputs from either notes or chords from other voices would allow cooperation among them.

Another approach to using multiple agents I found was coevolving networks together based on randomizing network nodes and creating chromosomes for each agent, then measuring the fitness, like in [2]. The agents' emergent behavior showed role-based actions, which would be nice for me since I am trying to create role-based musical agents, but the evolutionary algorithm as applied in my case would involve learning music from scratch, which I am trying to avoid, and a fitness function, which would be hard to define.

There are a few more papers involving multiple agents with LSTM networks, some of which involve reinforcement learning, that I would like to read before using my strategy above, but in the meantime I also wanted to get a machine learning environment set up.

3. Since Last Submission:

I saw a lot of references to [1] on the internet, and in my opinion it took some work and improved on it in a fine way and had the best results I could find without overfitting. Therefore, I decided to download the code from github and get it up and running so that I can tinker with it. I have also exported some of my own music to MIDI so that I can use it with the LSTM. I have also read numerous papers while I've looked for the best approach to my problem. I have been trying to come up with ways to merge different concepts.

4. I'm pretty much on track from where I thought I would be at this point. I still need to solidify some ideas, but I am in the right direction.

5. One thing I will change from the original plan is that I will not try to generate music in real-time. Generation comes from inputting seed values into the nets and then letting them run indefinitely, but this is too time-consuming to be able to listen to at the same time.

6. I had not originally planned for FIPA-ACL.

a. The communication information would be sent as inputs to the networks.

b. It would provide more information about who is sending the information and make it easier to place the information in the proper inputs and allow the agents to act accordingly. It may provide for a hierarchical structure, like following the bass line to determine chords. It would also allow the networks to be more distributed because a proper protocol would be needed to run on different machines.

c. Training networks with communication protocols would complicate the training process. There might also be too much information in the protocol, and too much information hurts generalization.

d. I would have to change the structure of the code to allow message passing from network to network.

References:  
 [1] – Johnson, Daniel. Composing Music With Recurrent Neural Networks (8/2/2015), <http://www.hexahedria.com/2015/08/03/composing-music-with-recurrent-neural-networks/> (DOI)

[2] – Yong, Chern Han and Miikkulainen, Risto. Cooperative Coevolution of Multi-Agent Systems. Department of Computer Sciences, University of Texas at Austin. <http://nn.cs.utexas.edu/downloads/papers/yong.tr287.pdf>