**Computer Science and Engineering, University of Nevada, Reno**

**MelodyBot**

**Team #19**

**Nicholas Harris**

**David Neilsen**

**Joseph San Nicolas**

**Stefan Stukelja**

**Instructors: Devrin Lee & Sergiu Dascalu**

**External advisor(s): Dave Feil-Seifer**

**Benjamin Brown**

**February 9th, 2018**

# **Table of Contents**

Abstract …………………………………………………….........2

Project Description……………………………………………….2

Significance……………………………………………………….3

Legal and Ethical Aspects……………………………………….3

Changes and Progress since the Initial Project Concept…….4

Project Responsibilities…………………………………………..4

Project Monitoring and Risks…………………………………….5

Team Overview……………………………………………………5

Contribution of Team Members ………………………………....6

References………………………………………………………...7

# 

# **Abstract**

MelodyBot is is an artificial intelligence whose purpose is to create unique music samples based on the genre and other parameters that are specified by the user. MelodyBot is able to create the desired music sample thanks to its use of a Recurrent Neural Network (RNN). Currently, the neural network has been able to generate short music samples ranging from 6-30 seconds that are of a simple nature. The developers of MelodyBot also have created functioning user interface prototypes in Python, Visual Basic, and with Qt. The current UI prototypes are able to load sound files in a viewable list, play and stop/pause said sound files, and remove the sound files from the list.

# **Project Description**

The goal of the project is to create a functional artificial intelligence which can generate music of a selected genre, taking into account other custom choices imposed by the user. This product will utilize a trained neural network which will work to assist and entertain its users. This artificial intelligence would be significant on the market because the music-generation market is small, and the opportunity to gain a competitive advantage is present.

The main functionality of the product comes through the artificial intelligence which will generate music depending on user input. It will achieve this through its implementation of a neural network which will use information from sampled music to assist in the generation of music. There will also be an interface with which the user can interact with to provide a wide range of input. The interface will include choices corresponding to different genres of music for the user to choose from. This decision will determine what music gets sampled. Currently the interface is minimal however additional features will be considered during planning. The initial implementation will be in Python to lay down the groundwork for the product. The product will be in the form of a desktop app. The intended audience for this product could include anyone who enjoys listening to music, but in particular those who are interested in generating new music. As a result, this audience is potentially global and provides the product a very large demographic to allow it to grow. Users will benefit from this product by having the option to use and listen to new, unique music.

The potential for future enhancement of the project is almost infinite. This is because the quality of music the neural net produces can continuously be improved through the use of new algorithms or more powerful training. In addition, there will almost always be room for improvement and expansion of the user interface, whether it be on a website, desktop application, or in a mobile application. There will also be the potential for additional products after the one produced for the course. These could include applications for additional platforms such as iOS.

The primary challenge of the development of the project will likely be the difficulty of working with neural nets to produce quality music. Initial work has already been done on this front as a proof-of-concept. Working as a team around this relatively difficult technology will also pose challenges. Finally, releasing a final product into the market has no guarantee of success, and attention must be paid to the quality of the product to improve market acceptance.

This project will make use of the extensive Python neural network libraries to produce a program capable of learning from various genres of music and producing its approximation of what it has learned. Due to the sequential nature of music, a recurrent neural network (RNN), is a promising tool for achieving this project’s goal. For a finished product, Android Studio or another development environment will allow the project to reach a market platform.

# **Significance**

MelodyBot is an interesting project due to its inherently creative nature. The goal is to make use of the artificial intelligence to create novel and enjoyable music and to allow the user to tailor the settings of the AI to allow experimentation and fine-tuning by the user to create even more novel musical samples. MelodyBot should be able to serve as a creative outlet for those who are interested in producing music through artificial intelligence. Because of the potential for others to find interest and enjoyment in the application, and because of the challenge and inspiration it provides for its developers, MelodyBot is a project worthy of development and continuous improvement.

Developing MelodyBot is a challenge for many reasons, and it will no doubt expand every team member’s professional skills to complete it. Firstly, it makes use of a neural network, a powerful and advanced tool that is becoming increasingly popular in modern applications and in many industries. It is also not trivial to incorporate the neural network into a polished and robust user-friendly application, and the process of learning how to do so and improve the implementation will be valuable experience for the developers. The MelodyBot team fully intends to make a professional application, and to learn at every step of the process.

There are multiple examples of AI-assisted music generation applications on the market today. Some of these are: Computoser, an in-browser application that generates short songs; Jukedeck, an application meant to make royalty-free music suitable for videos; and Flow Machines, a tool mean to assist musicians in composing music. The team hopes to set MelodyBot apart from these other tools by giving the user a much larger set of options for customization to fine-tune the produced musical samples for their needs. In addition, a professional and intuitive user interface will make MelodyBot a quality product worthy of attention.

There is potential for MelodyBot to become a popular application. Further, there is almost endless room for improvement after an initial release, as there are many opportunities to expand the functionality of the application and to improve the quality of the output by experimenting with different neural network architectures.

# **Legal and Ethical Aspects**

MelodyBot does not rely on proprietary software and so should have no major legal or ethical concerns. There may be some issue in acquiring a large dataset of songs of a modern genre, due to copyright and potential expense. However, there are huge archives of songs in the public domain for a wide swath of genres, so there should be no overarching issue in acquiring a dataset large enough for the AI to produce quality music in general.

The integrity of the application is important to the developers, and each is committed to producing a professional-grade application. To do this, the team will collaborate to identify and tackle the problems that need to be solved in order to progress. This means being realistic about how to approach the features the team wants to implement, being expedient and honest when it comes to communication, and thoroughly test all aspects of the application before it sees release.

# **Changes and Progress since the Initial Project Concept**

MelodyBot has an initial implementation of both the artificial intelligence and the user interface. Over the course of the development of the user interface Team 19 has created prototypes using Python, Visual Basic, and Qt. Ultimately, Visual Basic has been chosen for the final implementation as it fulfills all of the requirements necessary to create MelodyBot. Qt was recommended due to its visual appeal, however after extensive testing it was concluded that MIDI files were incompatible with the frameworks Qt is built on and would not be able to fulfill the requirements. GUI development with Python libraries was prototyped but found to be lacking in visual appeal when compared to Visual Basic.

On the artificial intelligence side of things, the decision was made to utilize a Recurrent Neural Network (RNN) for the initial implementation of MelodyBot. This decision was made after experimentation with both RNNs and Generative Adversarial Networks, the original candidate for use in the implementation, resulted in a preference of the developers for RNNs.

# **Project Responsibilities**

**Neural Network:** Creating and improving the machine learning of the artificial intelligence, which directly corresponds to the quality of the generated music samples - Nicholas

**User Interface Development:** All the functionality of the main user interface - Stefan, Joseph

**Website:** Frequently updates the project website with new information - Joseph, David

**Project Management:** Facilitating communication and collaboration between group members to focus work on new stages of development - Stefan, David

# **Project Monitoring and Risks**

In order to monitor the quality of the neural network’s music generation, the team allows the neural network to train under certain parameters for a set amount of time, let the network generate multiple music samples, and have the network train again with the same parameter and time constraints. The team then compare the samples it generates with the previously generated samples. To ensure that the user interface will be complete, each team member that is working on the UI will focus on a different aspect of the UI at a time. The team members will then go over with each other what they were able to accomplish in that certain time constraint. The UI component that still requires work takes priority. The team updates the website whenever any major information regarding the project’s development occurs.



# **Team Overview**

Each member of this team has skills conducive to this project’s success. The following are each team member’s skills and expected contribution to this project:

* Nicholas Harris: Experience with web development and Python implementations of neural networks. Will work with team to produce project core and its final implementation.
* David Neilsen: Experience with Android Studio, Java, and other languages useful for a final implementation. Will assist all aspects of project development.
* Joseph San Nicolas: Excellent team coordination skills and proficient in software development. Excited to work on this project’s final implementation.
* Stefan Stukelja: Skilled problem solver and coder. Will be useful for all software aspects of the project, from its core to the development of it for a specific platform.

Benjamin Brown and David Feil-Seifer of the University of Nevada Reno are external advisors for this project.

Artificial intelligence is a growing industry, and having demonstrable experience with it will be excellent for the professional development of each team member. The development of the project for a particular platform, be it web-based or as a mobile application, will also aid in the professional growth of the team. Taking an idea from start to finish and ending with a working product will be great experience for software development in professional industry.

# **Contribution of Team Members**

**Time Log**

This is a record of each team member’s time spent on each section of this report:

* **Nicholas Harris**
  + **Significance Section, researching related projects: 2 hours**
  + **Legal and Ethical Aspects Section: 1 hour**
* **David Neilsen**
  + **Assistance with Significance, Legal and Ethical Aspects: 1 hour**
  + **Revising paper: 1 hour**
* **Joseph San Nicolas**
  + **Updating project website: 1 hours**
  + **Abstract, Project Responsibilities, and Project Monitoring: 2 hour**
  + **Revision and edits of paper: 1 hour**
* **Stefan Stukelja**
  + **User Interface Prototyping/Research: 3 hours**
  + **Changes and Progress/Risk Management/ Edits & Revisions: 1 hour**

# 

# **References**

Related Projects:

Flow Machines: AI Music Making

<http://www.flow-machines.com/>

Jukedeck: Royalty-free music generation

<https://www.jukedeck.com/>

Computoser: A browser-based music creation tool

<http://computoser.com/>