Project Plan

for

Melody Maker

**Distribution:**

<Organization., Belguim Campus>

**Appendices:**

<Appendix 1>

**Contents**

[OVERVIEW 3](#_Toc512782023)

[GOALS AND SCOPE 3](#_Toc512782024)

[Project Goals 3](#_Toc512782025)

[Project Scope 3](#_Toc512782026)

[Included 3](#_Toc512782027)

[Excluded 4](#_Toc512782028)

[ORGANIZATION 4](#_Toc512782029)

[Organizational Boundaries and Interfaces 4](#_Toc512782030)

[Project Organization 4](#_Toc512782031)

[Project Manager 4](#_Toc512782032)

[Project-internal Functions 5](#_Toc512782033)

[Project Team 6](#_Toc512782034)

[Steering Committee 6](#_Toc512782035)

[SCHEDULE AND BUDGET 6](#_Toc512782036)

[Schedule and Milestones 6](#_Toc512782037)

[Budget 7](#_Toc512782038)

[Risk Management 7](#_Toc512782039)

[Risk Identification Table 7](#_Toc512782040)

[SUB-CONTRACT MANAGEMENT 7](#_Toc512782041)

[COMMUNICATION AND REPORTING 8](#_Toc512782042)

[DELIVERY PLAN 8](#_Toc512782043)

[Deliverables and Receivers 8](#_Toc512782044)

[**QUALITY** ASSURANCE 9](#_Toc512782045)

[CONFIGURATION AND CHANGE MANAGEMENT 10](#_Toc512782046)

[Configuration Items: 10](#_Toc512782047)

[Changes 10](#_Toc512782048)

[Configuration management 11](#_Toc512782049)

[SECURITY ASPECTS 11](#_Toc512782050)

OVERVIEW

We will create MelodyMaker by using a neural network, we can train a neural network to recognize/analyze melodies in sheet music -. midi files and learn from them. We will be using Tensorflow on the Anaconda IDE to develop our system. After we have trained the neural net to create its own melodies we can then add other musical components such as a drum beat and a bass line. The neuralnet should then be able to generate its own unique music. Will it be capable of producing music indistinguishable from a human composer?

GOALS AND SCOPE

Project Goals

|  |  |  |
| --- | --- | --- |
| **Project Goal** | **Priority** | **Comment/Description/Reference** |
| **Functional Goals:** |  |  |
| <Functional goal #1> | 1 | Create a neural network that can learn from input - .midi files converted into .csv files and then generate its own music. |
| <Functional goal #2> | 2 | The neural net should be able to focus on melodies and then learn to create its own and build around that concept. |
| **Business Goals:** |  |  |
| <Efficiency > | 4 | Provide a way of generating high quality music for musicians without wasting any resources on composers. |
| **Technological Goals:** |  |  |
| <Technical goal > | 1 | Neural network should be able to that we can train to recognize/analyze sheet music |
| **Quality Goals:** |  |  |
| <Quality goal > | 3 | Generate high quality music focused on maintaining a melody. |
| **Constraints:** |  |  |
| <appl. specific standards> | 3 | The music that is generated from the neural network must be indistinguishable from a human composer. |

Project Scope

Included

* Project plan
* Test plan
* Source code – Actual neural net
* Requirements

**Tensorflow** – a deep learning library created by google. We will be creating our neural network on this platform.

**Anaconda IDE** – A python IDE developed to help connect deep learning libraries. We have installed our Tensorflow here.

**Udemy Courses** – Udemy has a wide variety of courses with info we need to complete this project. We have already purchased around 20 of these courses to help us accomplish our goals.

* Specification files
* Documentation
* System Design

Excluded

* Hardware will not be included
* End user training documents

ORGANIZATION

Organizational Boundaries and Interfaces

*MelodyMaker’s boundaries will be the input data received from users that used the method previously. The melodies from data that has already been stored will form the interface that will enable the program to create new songs and melodies.*

*The program will look at aspects such as repetition and structure and use this to refine the new melodies that are to be written. Using these repetition and structure repetitions the program will be able to ensure that the melodies ”sound” harmonious and that it is audible to the listener.*

Project Organization

*The staff consists of: Ernes Smit (Technical Project manager; Programmer) , Nicolas van der Walt (Project Manager) , Marius Liebenberg (Programmer) , Drikus Mostert (Administrative leader; Programmer), Jo’ash Munian(Administrative coordinator), Mkhize (Administrative assistant)*

*Communication within the hierarchy will follow the depicted line of communication (Upwards). The structure of the organogram will also make provision for horizontal lines of communication. These horizontal lines of communication will hold no authoritative bearing but will serve as instructional guidelines and in – service training from more experienced employees.*

*Relevant responsibilities and Titles have connoted at the top next to staff names.*

Project Manager

|  |  |
| --- | --- |
| **Role** | **Organization: Name** |
| Project Manager | Nicolas vd Walt |
| Technical Project Mgr. | Ernes Smit |

Roles:

Project manager: Overall overseeing of the functioning and implementation of all initiatives and ideas as well as the coordination of the project finalization. Continuous communication must be given to all other staff members regarding updates on the project completion and implementation. Furthermore, the project manager will oversee that all financial matters are regulated and conducted within the budget framework and that the necessary meetings are held if budget adjustments are to be made. Also, frequent communication with all external stakeholders and possible clients should be done in order to ensure that their inputs and opinions will be implemented into the project.

Project-internal Functions

|  |  |  |
| --- | --- | --- |
| **Function** | **Organization: Name** | **Comment** |
| Quality Assurance | Ernes Smit | Making sure the programme is implemented effectively and if errors arise they are eliminated and provision is made for them to not occur again |
| System Test Lead | Ernes Smit | The system will be tested on a frequent bases before it is released and after it has been released it will have to be tested on a frequent basis when new updates and improves have been made |
| Validation Lead | Ernes Smit | When the melodies have been recorded into the program and the program has manufactured new melodies of its own, the new melodies will have to be verified as authentic. |
| Configuration Mgmt | Drikus Mostert | As software and hardware necessities arise they will have to be installed and introduced to the already existing structure of software and hardware |
| Change Mgmt | Marius Liebenberg | When clients have requests on changes that they would like to have implemented, they together with their requests will have to be managed and implemented |
| Communication | Drikus Mostert | Internal as well as external communication should go through an individual authoritative person in order to ensure that no miscommunication takes place. |
| Administration | Mkhize | All communication and all documents will have to be done in a professional manner and kept in a professional and safe place. |
| Financial Management | *Jo’ash Munian* | All budgets and expenditure will have to be managed and assurance that expenditure does not exceed the specified budget will need to be a major priority |

Project Team

|  |  |  |
| --- | --- | --- |
| **Organization: Name** | **Availability** | **Comment** |
| Nicolas van der Walt | Available | Active in project |
| Ernes Smit | Available | Active in project |
| Drikus Mostert | Available | Active in project |
| Mkhize | Available | Active in project |
| Marius Liebenberg | Available | Active in project |
| *Jo’ash Munian* | Available | Active in project |

Steering Committee

|  |  |  |
| --- | --- | --- |
| **Organization** | **Name** | **Comment** |
| Belgium Campus | Nicolas vd Walt | Project Manager |
| Belgium Campus | Desire Sundire | Client of the project |
|  |  |  |
|  |  |  |

SCHEDULE AND BUDGET

Schedule and Milestones

|  |  |  |  |
| --- | --- | --- | --- |
| Deliverable | START DATE | DURATION | |
| Proposals due. | 2018-03-21 | - | |
| Release and distribution of RFP. | 2018-03-22 | - | |
| Initial meetings. | Once per week on Tuesday | 1 hour | |
| Sponsor selected. | 2018-03-22 | - | |
| Deadline for sponsor to submit written questions and/or non-mandatory notice of intent. | 2018-04-16 | 6 weeks | |
| Questions with written answers provided to sponsor. | 2018-04-30 | 2 weeks | |
| Analyze requirements | 2018-05-01 |  | |
| Development methodology chosen | 2018-05-01 |  | |
| Development milestones Design architecture, frontend, backend | 2018-05-01 |  | |
| Build app | 2018-07-01 |  | |
| Testing. | 2018-07-02 | 3 weeks | |
| Final testing and debugging | 2018-10-01 |  | |
| Finished due date | 2018-11-01 |  | |
|  | | | |
| Budget  Budget summary: Seeing as we can fully create this project on a deep learning library – Open source and free. Our only costs will be acquiring the needed skills to finish this project.  Udemy has many courses in Python – The main language used by deep learning libraries.  I and my team have purchased a few to help us in development.  Receipts available on request. | | | |
|  | | | |
|  | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Cost Item** | **QTY** | **Unit Cost** | **Total Cost** | **Maintenance first Years** | **Maintenance 5 years** | **Maintenance in %** |
| Udemy course | 20 | 120 | 2400.00 |  |  |  |
|  |  |  |  |  |  |  |
| **TOTAL COST** |  |  | 2400.00 |  |  |  |

Risk Management

The purpose of this step is to identify what could go wrong and what are the consequences of it occurring.

Risk Identification Table

|  |  |  |
| --- | --- | --- |
| **Risk Identification** | **Risk Severity** | **Risk Response** |
| Hardware Failure | Low – Computer hardware has a chance of failing and crashing. | Risk Acceptance – this is an inherent risk in creating a system. |
| Security Breach | High – Secret information being stolen or shared. | Transferring the Risk – Make all members involved sign a non-disclosure agreement. |
| Lack of Skills | Low – Project hitting a wall and not being able to complete. | Risk Acceptance – This is an inherent risk all be it low. |
| Natural Disaster | Low | Reducing the risk – Creating backups of the project on a scheduled interval. |

SUB-CONTRACT MANAGEMENT

|  |  |  |  |
| --- | --- | --- | --- |
| **Sub-contractor** | | **Sub-contracted Work** | **Ref. to sub-contract** |
| **Company** | **Contact** |
| None |  |  |  |
|  |  |  |  |
|  |  |  |  |

COMMUNICATION AND REPORTING

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Type of Communication** | **Method / Tool** | **Frequency/Schedule** | **Information** | **Participants / Responsibles** |
| **Internal Communication:** | | | | |
| Project Meetings | Teamspeak and in person | Two Weekly and on event | Project status, problems, risks, changed requirements | Project Manager Project Team |
| Sharing of project data | Shared Project Server and in person | When available | All project documentation and reports | Project Manager  Project Team |
|  |  |  |  |  |
| Milestone Meetings | Teamspeak or in person | Before milestones | Project status (progress) | Project Manager Sub-project Team |
| Final Project Meeting | In person | M6 | Wrap-up  Experiences | Project Manager Project Team |
| **External Communication and Reporting:** | | | | |
| Project Report | Excel sheet | Monthly | Project status - progress - forecast - risks | Project Manager Sub-Project Managers |
| Belgium Campus Meetings | In person | Monthly |  | Project Manager, Belgium campus |
|  |  |  |  |  |

DELIVERY PLAN

Deliverables and Receivers

|  |  |  |  |
| --- | --- | --- | --- |
| **Identity** | **Deliverable** | **Planned Date** | **Receiver** |
| D1 | Individual project proposal document submission | 2018-02-22 | A.Joy |
| D2 | Milestone 1  Final project proposal document submission | 2018-03-21 | A.Joy |
| D3 | Milestone 2  Planning document submission | 2018-04-13 | A.Joy |
| D4 | Milestone 3  System Analysis and Design document submission | 2018-05-04 | A.Joy |
| D5 | Milestone 4  Technical 1: Database Design | 2018-05-25 | A.Joy |
| D6 | Milestone 5  Technical 2 : Class Design | 2018-06-14 | A.Joy |
| D7 | Milestone 6  Technical 3 : Object Behaviour Model | 2018-07-04 | A.Joy |
| D8 | Milestone 7  Technical 4 : Coding | 2018-08-04 | A.Joy |
| D9 | Test plan document submission | 2018-09-14 | A.Joy |
| D10 | Milestone 8  Testing | 2018-10-22 | A.Joy |
| D11 | Milestone 9  Implementation | 2018-10-29 | A.Joy |
| D12 | Project Submission | 2018-11-01 | A.Joy |

**QUALITY** ASSURANCE

This refers to the maintenance of a desired level of quality in a service or product, especially by means of attention to every stage of the process of delivery or production.

One of the key factors that product quality is achieved is by the use of the TensorFlow platform. When it comes to neural machine translation, TensorFlow reduces errors by 55%-85%. In Neural architecture search, one can figure out what is the right neural network to use for a problem. TensorFlow allows coders to iterate quickly, train models faster and run more experiments. With regards to quality assurance in production as well as support— teams can run TensorFlow on large-scale server farms embedded on devices, CPUs, GPUs, TPUs.

Furthermore, the use of GitHub widely extends development quality. It makes the acquisition of documentation quite easy. This also includes help guides. GitHub can integrate with common platforms such as Amazon and Google Cloud, services such as Code Climate to track your feedback, and can highlight syntax in over 200 different programming languages, thus, making quality of integration a key advantage. GitHub is a repository so all work can be publicized.

Software quality is regarded as the highly important factors for assembling the global competitive position of any software product. For quality prediction a neural networks are used for better prediction accuracy. The application software is first subjected to the test case generation and once they are generated they are applied to advance neural network for prediction quality.

The neural network is improved and then optimizes the weight factor for improving the prediction. The quality metrics like maintainability and reliability are estimated for predicting the software quality and the results are compared with other existing techniques to verify the effectiveness of the proposed method.

CONFIGURATION AND CHANGE MANAGEMENT

Configuration management systems and change management systems are used to manage any sudden changes in a project or system. Some changes may affect the project baselines, such as scope, the time and the cost. Some changes that might be required may be that of the product itself, like the specifications.

Configuration Items:

* Final project proposal document - this includes the final proposal that will be assessed before commencement of the project. It is the first item of the development process that will be used to guide all other items on a basic level.
* Planning document – Much like the proposal, the planning document will be used to guide the development of all other items, however, it will be used on a more in-depth scale. It provides guides and steps in a predefined manner that allows the configuration of all items to be executed as harmoniously as possible
* System Analysis and Design document – Here we identify, break down and assess all major and minor aspects of the development of the system, the environment, effects and the functionality of the system itself.
* Database Design – This is technical aspect of the system. The database is one of the most important physical components of the system. Once this has been correctly developed, it an also be used for Class creation which is the next item.
* Class Design – This item can be based on the elements found in the database design. It is a technical item that defines the basis of code development of the system.
* Object Behavior – This technical item works with thee classes and database as a baseline. However, it is also the baseline for coding the system. It will be assessed and defined before coding commences for the purpose of simplifying the intense complicated nature of coding.
* Coding - This is the item that configures all previous items. It is the physical compilation of all planning up until this point. The system is physically created here.
* Owner and Creator – In this situation, the system will be owned by the creators. It is not custom built for any organization or party, however, future versions of the system may allow customization either done by the creators or trained personnel.
* Documentation – All documentation will be handled and recorded by the creators of the system and will be assessed by the receiver of all deliverables which are discussed in **Number 7.** Documentation will also be handled with the use of the GitHub platform.

Changes

* Client needs

This is one of the factors that influence us to make changes in the system because if the client is not very happy with the system then some changes will have to be made. That is why it is important for us to consult the client during each and every phase so that they check if they still satisfied or not.

* Project Schedule

This is a very important factor since the project will have to be completed in time, regardless of whether the time has been cut short or extended.

Configuration management

* Adding a feature to the system

This is with regards to any functionality of the system that might need to be changed or add a new one. This will require new configuration, for example if our client needs us to add new feature or change a feature in our system then we are going to change everything.

* Product Specifications

If there are any other specifications that can be added to the system to make it perform even better, we will have to look at the environment first then check the market for our competitors.

Fortunately, the life cycle for the development of the system allows for easy methods of identification, assessment and implementation of changes within the development process. The configuration items are all key points of each milestone in the development process. This makes it easy to pin point easier or more effective ways to commence each step. Thus, smaller and larger changes can be made in a timely manner to ensure that all members are notified and included in these changes. The timeliness factor is also helpful because many changes do not take place simultaneously which means the process will not be complicated.

Regardless of the simplicity of making changes, we still need to follow steps that will make sure these changes are efficiently superior as opposed to keeping it the same. These are the steps:

|  |  |
| --- | --- |
| **Step** | **Description** |
| Generate CR | A submitter completes a CR Form and sends the completed form to the Change Manager |
| Log CR Status | The Change Manager enters the CR into the CR Log. The CR’s status is updated throughout the CR process as needed. |
| Evaluate CR | Project personnel review the CR and provide an estimated level of effort to process, and develop a proposed solution for the suggested change |
| Authorize | Approval to move forward with incorporating the suggested change into the project/product |
| Implement | If approved, make the necessary adjustments to carry out the requested change and communicate CR status to the submitter and other members |

SECURITY ASPECTS

Neural network based applications have been used successfully in the area of networks security as an intrusion detection system, misuse detection and firewalls. Also, in the field of application security, neural network has been proposed to be a virus detection system. It would be noticed however, that these neural networks can only provide a form security after software deployment. The project manager is in charge of deciding who the information of the system be given to, the client is the one who has the most rights to have any information related to the proposed system.

The availability of the information of the system will be given according to the roles the people play in the actual development of the system.

The artificial neural network voltage security monitoring and control is used. The neural network uses its association mechanism, the inherent parallel information processing nature of the neural network, which provides the capability of fast computation, enables the neural network approach to meet the demands of real time monitoring control