# COMP3000 Computing Project

## 2023/2024

### Project Title

Sheet Music Automation Tool

### Links

Source code:<https://github.com/tcollyer1/sheet-music-automation>

### Project Vision

For those who frequently compose or simply enjoy working out music by ear at their instrument, this sheet music automation tool is an embedded system that will enable musicians to automatically have their melodic ideas transcribed for them – saving them from the inconvenience of having to notate their music manually, and leaving more time for the creative process.

*Outline your initial project vision here. Setting out your project vision at the start does not set it in stone, it can change as you develop your ideas, but you must provide a starting point.*

*Follow the format suggested here* [*https://martinfowler.com/articles/lean-inception/write-product-vision.html*](https://martinfowler.com/articles/lean-inception/write-product-vision.html)

*For [who?]*

*Whose [problem]*

*The [name of your product]*

*Is a [type of product]*

*That [what are the key reasons]*

*No need for the different from or our product.*

### Risk Plan

*Identify here potential risks to your project and outline how you will address those risks. For further information view the risk management resources provided in support materials section.*

|  |  |  |
| --- | --- | --- |
| **Risk** | **Mitigation** | **Risk Level** |
| Not having the correct hardware fit for capturing and processing sound | As well as conducting research online, speak with supervisor before commencing development to establish what hardware (i.e. type of Raspberry Pi, any external modules) is likely required. Establish memory/CPU/audio recording requirements for sampling sound in real time | High |
| Not using the correct language/software fit for interfacing with and processing incoming sound data | Use a fitting programming language for high-speed, low level systems (C/C++) | High |
| Alterations to original project vision/requirements | Make use of Agile practices – ensure to plan ahead of each sprint and review previous sprints. Make use of source control to make changes incrementally | Moderate |
| Being unable to process data in real time well enough on a Raspberry Pi | Careful planning using sprints to come up with alternative solutions – such as recording an audio stream, storing it and processing it rather than real time interpretation | Moderate |
| Implementation of certain facets of the system take longer than expected | Review each sprint and plan ahead on a sprint-by-sprint basis to keep on top of incomplete work | Low |
| System is unable to perceive notes from a real instrument accurately | First conduct tests on sound with no underlying harmonics, i.e. sine waves, before testing on an instrument | High |
| System is unable to perceive multiple notes simultaneously (chords) | Focus on processing single, monophonic notes before attempting to increase complexity | Moderate |
| Hard drive failure, losing access to project code | Regularly commit and push to GitHub repo to ensure work is saved and retrievable at any given time | High |

### Proposed Gantt chart

*Identify here your proposed work packages and how they will fit together over the time you have available. This is a high level estimate.*

### Keywords

*Provide keywords for your project to enable searching*

Music, sheet music, automation, transcription, automatic, notation, notes, instrument