**Software Implementation and Testing Document**

**For**

**Group “Riskier”**

Version 3.0

**Authors**:

Bradley Hodson

Michael Styron

Antonio Vidal

Grayson Wagstaff

Wesley Watkins

# Programming Languages (5 points)

1. Python
   1. Python is the language used to create Riskier. It is used for both the structure of the game, as well as to create and train the AI within the game.
   2. This language was chosen for two main reasons. First, it happened to be a language that a majority of our group had some experience working with. Second, it was a viable alternative after issues arose with the Unity engine and C# scripting.

# Platforms, APIs, Databases, and other technologies used (5 points)

1. Pygame
   1. Pygame is a python library designed for creating games using the python language. Its modules are used to create the game itself, as well as the AI in the game.

# Execution-based Functional Testing (10 points)

In the project, before a branch is merged into the master, two reviewers are required to test and document each set requirement for that branch. This will check to see if the implementer has met the proper requirements, or if there are additional changes that need to be made. Although this would be at the end of a sprint, implementers will attempt to submit early enough to still make revisions before the final day of the sprint.

# Execution-based Non-Functional Testing (10 points)

In the project we plan to use automated pipelines within GitHub to pass build checks. By using these automated pipelines, we will test if the program will compile. Alerts will be sent to the programmer if certain requirements are not met (such as syntax and ability to execute).

# Non-Execution-based Testing (10 points)

This will mostly come down to code reviews between members of the group. These code reviews occur mostly during team meetings. While code reviews are not specified as part of the review process, by reviewing the code as a group we will be able to give constructive criticism to one another, while also being able to learn portions of the code together.