Software Implementation and Testing Document

For

Group <X>

Version 1.0

Authors:
Cade G-M

1. Programming Languages (5 points)

List the programming languages use in your project, where you use them (what components of your project) and your reason for choosing them (whatever that may be).

Unity

-used to design the video game aspect of our project

-chosen because it is one of the more popular languages for video game design and seemed very apt for our needs

Arduino

-used to program our hardware

-chosen because it is great for beginners starting with embedded systems

KiCad

-used to design our printed circuit board

-chosen because it has more resources than similar PCB design languages/ software, and the interface is very user friendly

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

List all the platforms, APIs, Databases, and any other technologies you use in your project and where you use them (in what components of your project).

Github

-used to share files for our project, fix bugs, and update code

No APIs yet

No Databases for this increment but probably Oracle or Microsoft SQL.

3. Execution-based Functional Testing (10 points)

*Describe how/if you performed functional testing for your project (i.e., tested for the **functional requirements** listed in your RD).*

For the hardware, we built a basic breadboard with lights and an arduino unit. Successfully tested code where the breadboard sequentially turned the lights on and off without any connection to a computer. For the software, we designed the main dinosaur character, background, and obstacles.

4. Execution-based Non-Functional Testing (10 points)

*Describe how/if you performed non-functional testing for your project (i.e., tested for the **non-functional requirements** listed in your RD).*

We have not tested any non-functional requirements yet. Once we have our first fully functional prototype (software and hardware) we will begin testing.

5. Non-Execution-based Testing (10 points)

Describe how/if you performed non-execution-based testing (such as code reviews/ inspections/walkthroughs).

We had a team meeting to see our current progress and to work on each of our future assignments. Everyone reviewed and contributed to the code for the entire project on github.