UAT Plan

for

Alakazam’s Hand

Index

1. Scope 3

1.1. Objectives and business requirements 3

1.2. Scope 3

2. Testing team 4

3. Environmental requirements 5

3.1. Hardware requirements 5

3.2. Software requirements 5

4. Test Scripts 1

# **1.** **Scope**

## **1.1.** **Objectives and business requirements**

In this section, outline the business requirements. In other words:

● What are our goals? What are we hoping to accomplish with this project/feature?

● How will we measure success?

*The goal of this user acceptance test is to ensure all features of Alakazam’s Hand work as intended to help Anna play the video games she loves. We hope for Alakazam’s Hand to be able to accurately read arm acceleration in multiple directions and convert that data into character movement in the chosen game, Temple Run 2.*

## **1.2.** **Scope**

In this section, outline the scope. This means:

● What is the pain point we’re trying to fix?

● What are we testing exactly, and what are we *not* testing?

*For this UAT test, we are aiming to test whether:*

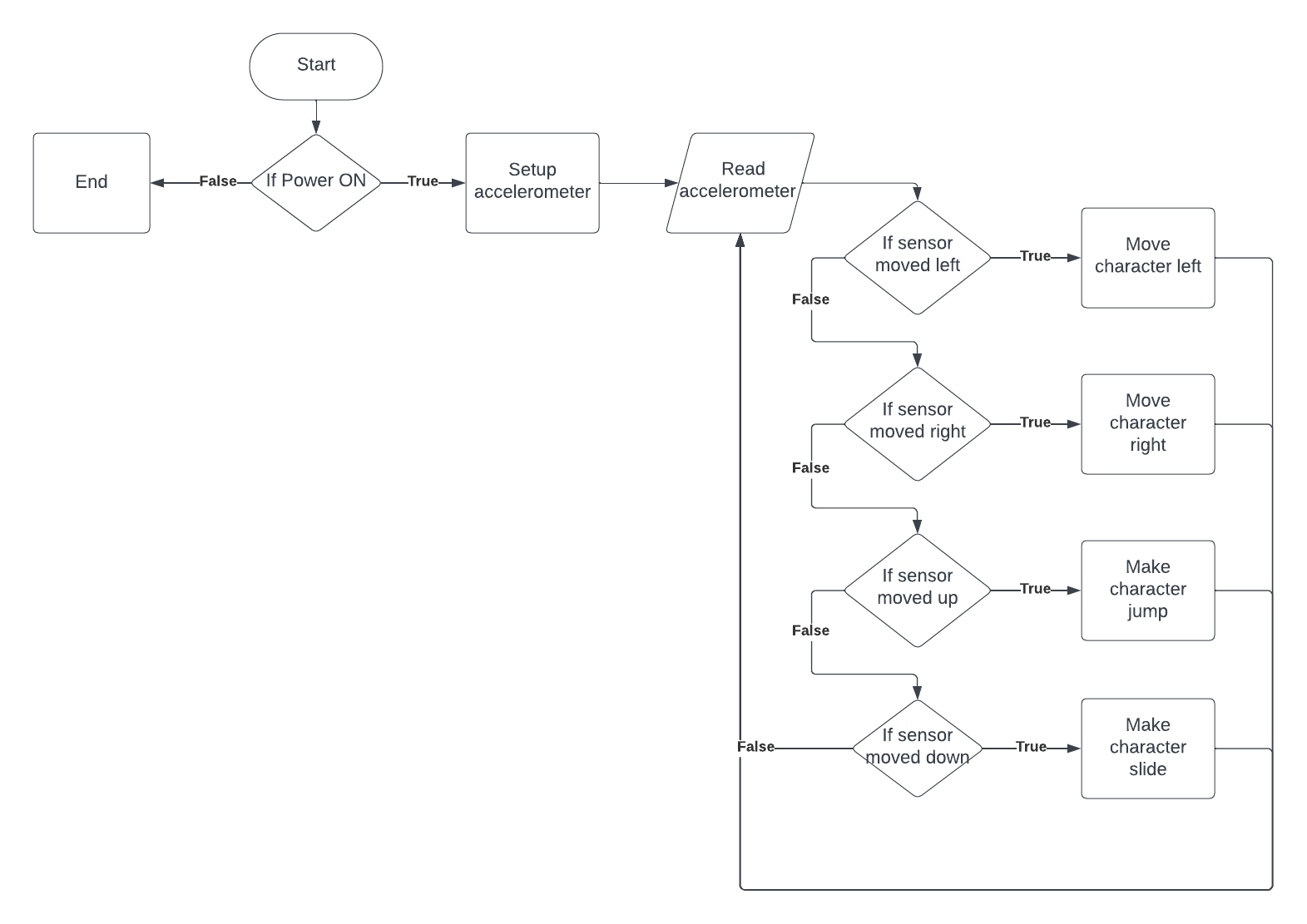
* *The gesture sensor detects movement in multiple directions*
* *The system can convert the gesture sensor’s data into inputs*
* *The gesture sensor turns off when supposed to*

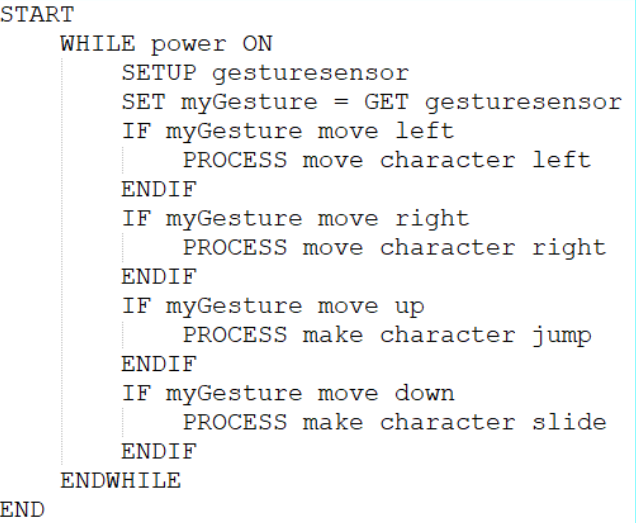
*During the UAT test, we are not testing:*

* *Whether the charging system works*
* *How long the battery lasts for*
* *How flexible our material is*
* *The delay between inputs and outputs*

## **1.3.** **System Diagrams**

In this section, paste any drawings or diagrams that help the UAT team understand the program being tested. With each drawing include a brief explanation of how the drawing represents the application or system being tested.

****

****

*The flowchart and pseudocode show how the gesture sensor should be collecting data and transmitting it to the computer. Depending on which direction the gesture sensor moves, will determine how it controls the character inside her game.*

# **2.** **Testing team**

In this section, list out members of your QA team and what their roles will be during UAT.

| **Name** | **Responsibilities** |
| --- | --- |
| Sunny Li | Communicate results between UAT testers and UAT reporter |
| William Phan | Tests whether the gesture sensor detects movement in each direction |
| Roman Lacbungan | Tests whether the system is able to read the gesture sensor’s data and convert that into computer inputs |
| Deklan Low | Tests whether gesture sensor gathers data when supposed to and whether system reads data when supposed to |
| Ethan Czapla | Tests whether data is converted into inputs at the correct time and how accurate it is |
| Cameron Ly | Writes final UAT report with tester results |

# **3.** **Environmental requirements**

## **3.1.** **Hardware requirements**

What hardware has the solution been designed for and should be tested on.

If that is the case, outline the minimal and recommended requirements so the QA team can verify that the software runs on the testers’ machines.

* *Computer running Temple Run 2*
* *Alakazam’s Hand (controller glove)*

## **3.2.** **Software requirements**

If any extra software or dependencies must be downloaded and installed, list them here.

* *Arduino IDE v2.1.1*
* *Google Drive*
* *Google Chrome / Mozilla Firefox / Microsoft Edge*
* *GitHub*
* [*LIS3DHTR.cpp*](https://github.com/TempeHS/TempeHS_Ardunio_Boilerplate/blob/main/TempeHS_Sensor_Catalogue/Sensor%20Kit/3_Axis_Accel_Sensor/LIS3DHTR.cpp) *(download and add to Arduino project file)*
* [*LIS3DHTR.h*](https://github.com/TempeHS/TempeHS_Ardunio_Boilerplate/blob/main/TempeHS_Sensor_Catalogue/Sensor%20Kit/3_Axis_Accel_Sensor/LIS3DHTR.h) *(download and add to Arduino project file)*

## **3.3.** **Network requirements**

Some software (design, video editing…) can be demanding on hardware specifications.

If that is the case, outline the minimal and recommended requirements so the QA team can verify that the software runs on the testers’ machines.

* *Internet connection*

# **4.** **Test Scripts**

This section is more important than it seems—it is crucial that both the QA team and the testers know what features must be tested, especially if you’re testing a lot at once.

| **Test** | **Describe the feature being tested** | **Describe the user input or test data** | **Describe the pass criteria** |  |
| --- | --- | --- | --- | --- |
| 1.1 | Gesture sensor movements are converted into computer inputs for game | 1. User turns on gesture sensor  2. User opens up game  3. User starts game  4. User moves glove left, right, up, and down | 1. User sees gesture sensor is working  2. When user moves glove, game character moves in corresponding direction | Tester name:   |  | PASS | | --- | --- | |  | FAIL |   Observations: |
| 1.2 | Gesture sensor gathers and reads data when supposed to | 1. User opens arduino  2. User turns on gesture sensor  3. User moves glove around  4. User checks the serial monitor for prints  5. User checks the serial  6. User checks the serial monitor for prints | 1. User moves glove around  2. User sees gesture sensor is creating correct outputs when moving glove  3. User sees gesture sensor is no longer creating outputs when glove is not moving | Tester name:   |  | PASS | | --- | --- | |  | FAIL |   Observations: |
| 1.3 | Glove movement and game movement are in sync | 1. User turns on gesture sensor  2. User opens game  3. User controls game character | 1. User moves glove around to control game  2. User sees there is no delay between movement actions | Tester name:   |  | PASS | | --- | --- | |  | FAIL |   Observations: |

Write step-by-step, detailed but concise instructions on how to test the feature.