**Localization Tool**

**Planning Document**

Assignment 1

SP1 2017 Introduction to C# and Tools Development

Academy of Interactive Entertainment

|  |  |
| --- | --- |
| **Author** | Sarthak Saxena |
| **Contact** | Sarthaksaxena42@yahoo.com |

Contents

[Executive Summary 2](#_Toc17708116)

[Screen Mock-Ups 3](#_Toc17708117)

[Main Window 3](#_Toc17708118)

[Attributes Screen 4](#_Toc17708119)

[State Diagram 8](#_Toc17708120)

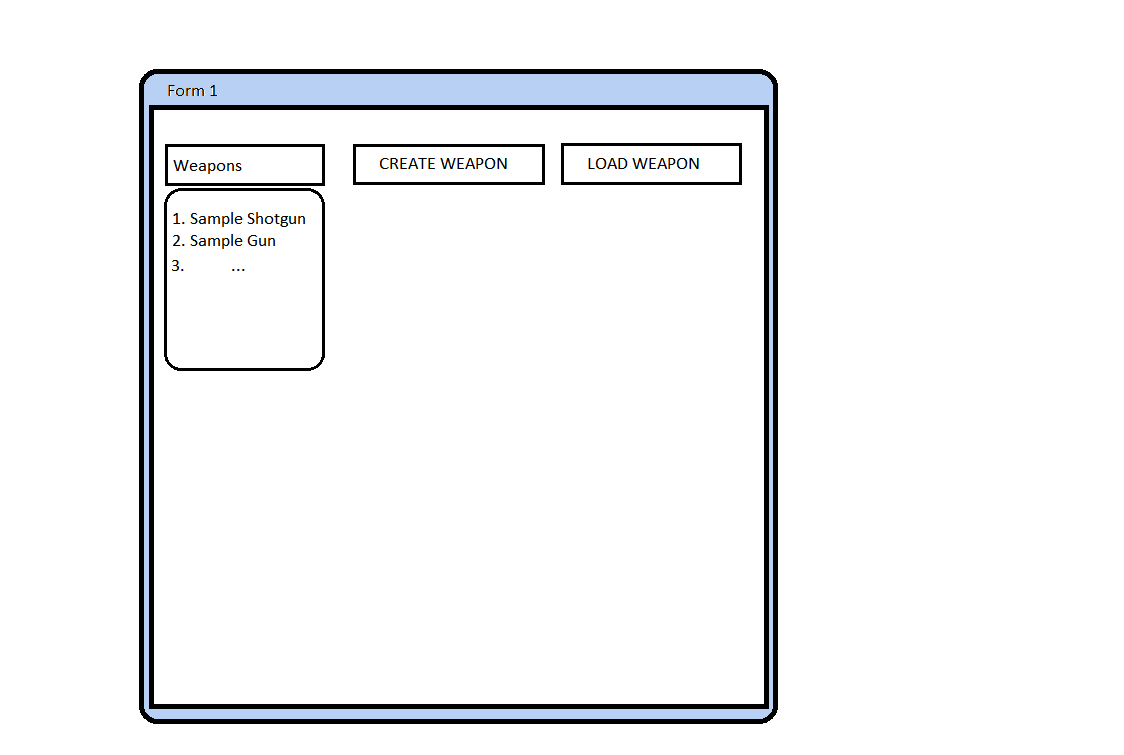
[Save Data Format 9](#_Toc17708121)

# Executive Summary

This document details the proposed architecture for a tool to be used to assist in game development.  
  
This tool will allow the user to create a weapon, with attributes of their choice to be used in a game. Users can create their own weapon, with their own attributes plus name. The tool allows the user to use create their own weapon without outsourcing any image file, or researching attribute types.  
  
The program will compile a local database of the data entered that can be imported into a game. Data created can also be saved in a xml file format, to be used for future purposes in other games.

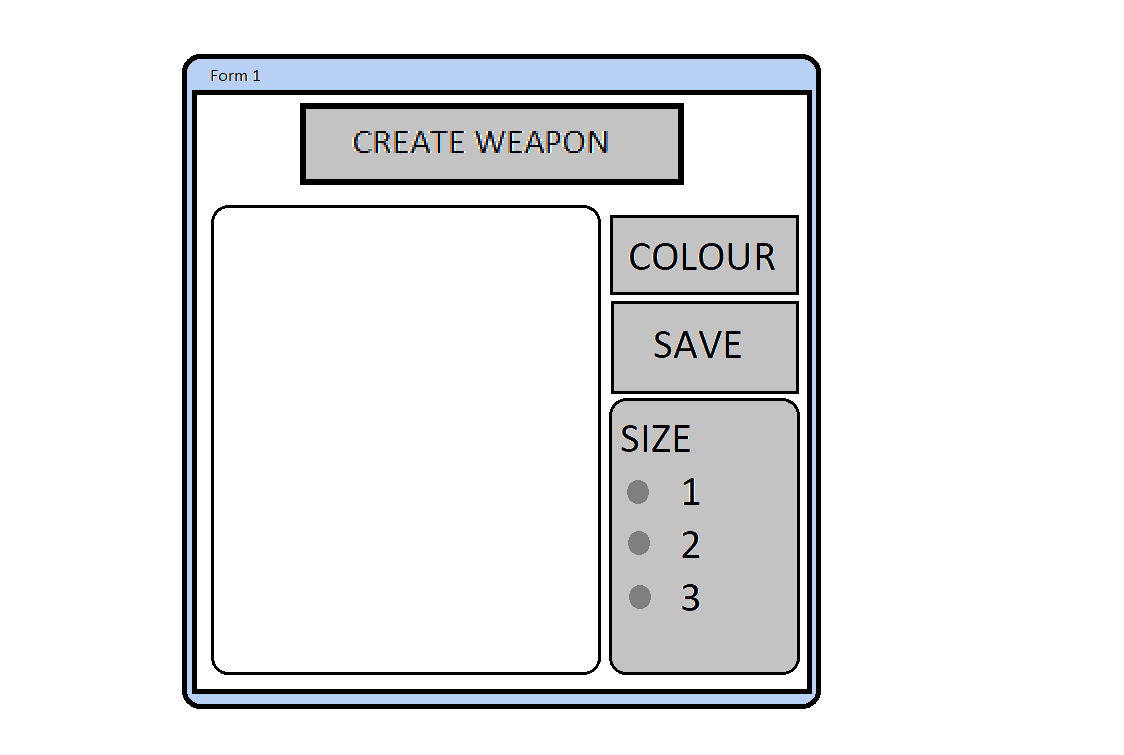
# Screen Mock-Ups

## Main Window

Figure 1. Shows the main interface for the tool.  


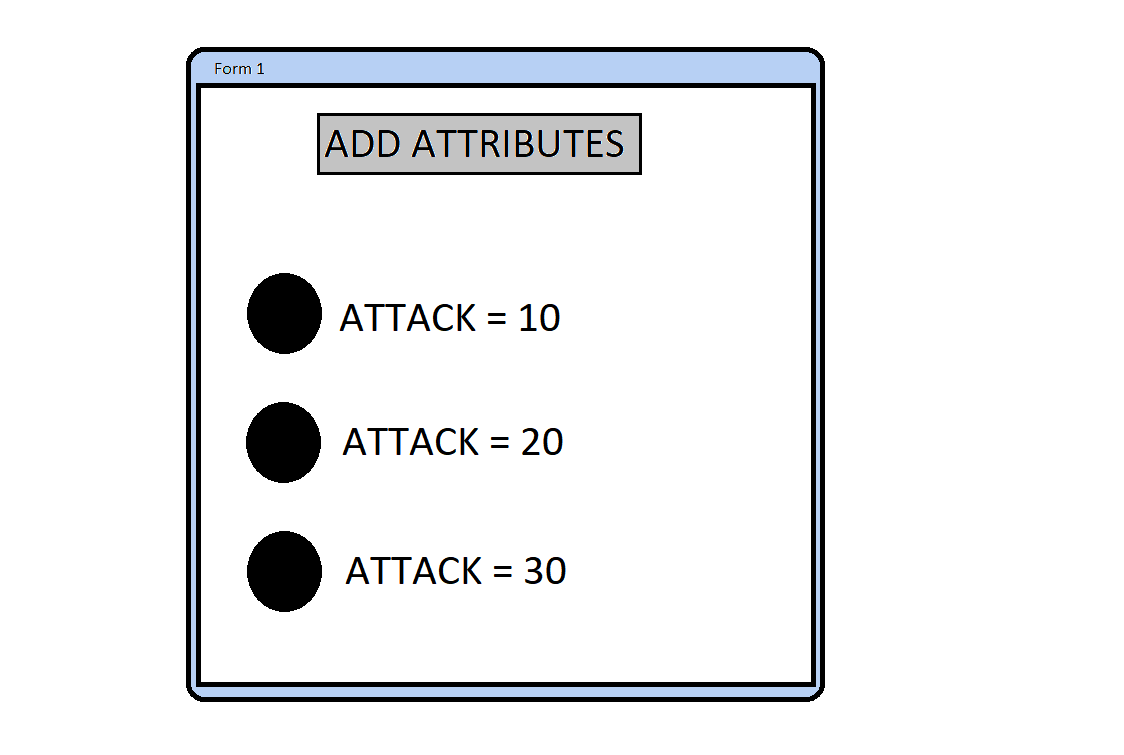
**Figure 1. Shows the main interface for the tool.**

Buttons such as the “Create Weapon” allows the creation of weapons. “Load Weapon” loads in saved weapons from list of “Weapons” list. By pressing on the one of the items in list, then pressing “Load Weapon” it will load weapons image, and attributes assigned to it from a local saved folder.  
  
Create WindowAllows the user to create a weapon by hand and save it referring to figure 3.   
Buttons such as “Colour” will allow the user to pick from a range of colours to draw weapon. As well as the button “Save” opens up a save window referred in figure 3. from which you may save your creation. The “Size” button is to add depth to your weapon, so the user feels more enticed into creating something with a different size so each weapon looks creative and unique. Once the user is happy with their image they can “Save” or press “Create Weapon” to open up attributes screen referred on Figure 3.



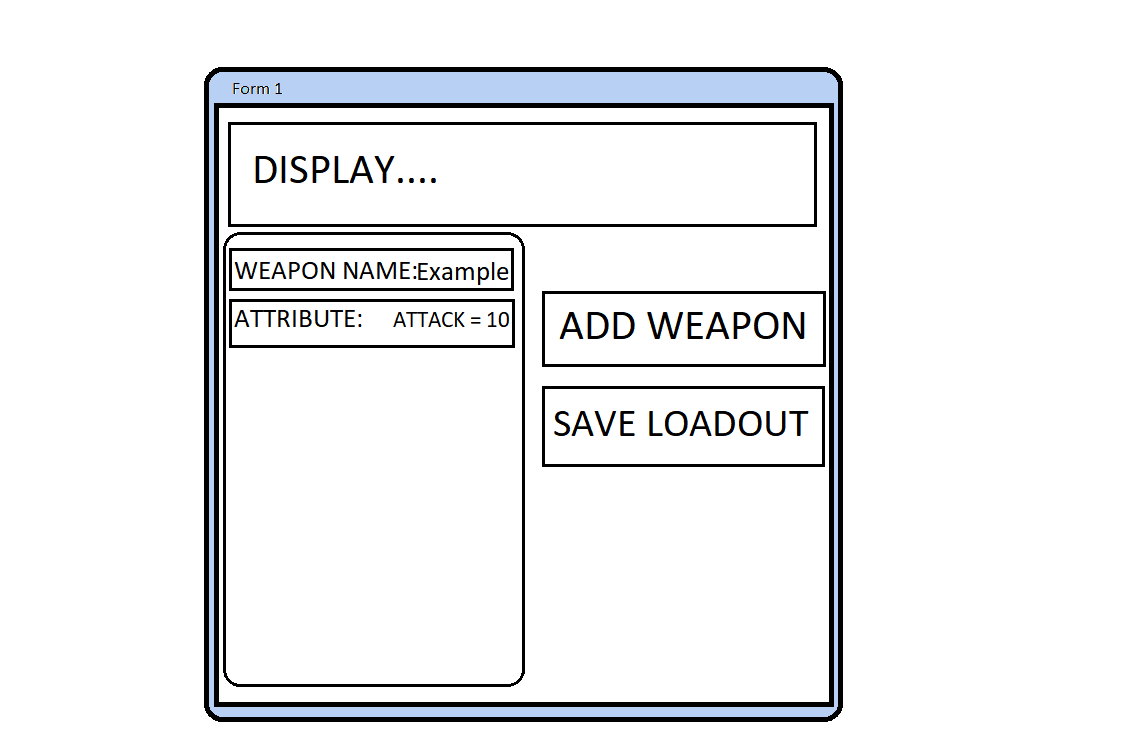
**Figure 2. Open up window to draw Weapon**

Attributes ScreenThe user will be guided to a list of attributes to place on their weapon image, to finalize. A list of attributes will be displayed, and the player has to pick between which option to keep on the weapon. This is displayed through Figure 3.

****

**Figure 3. Shows adding attributes to Weapon**

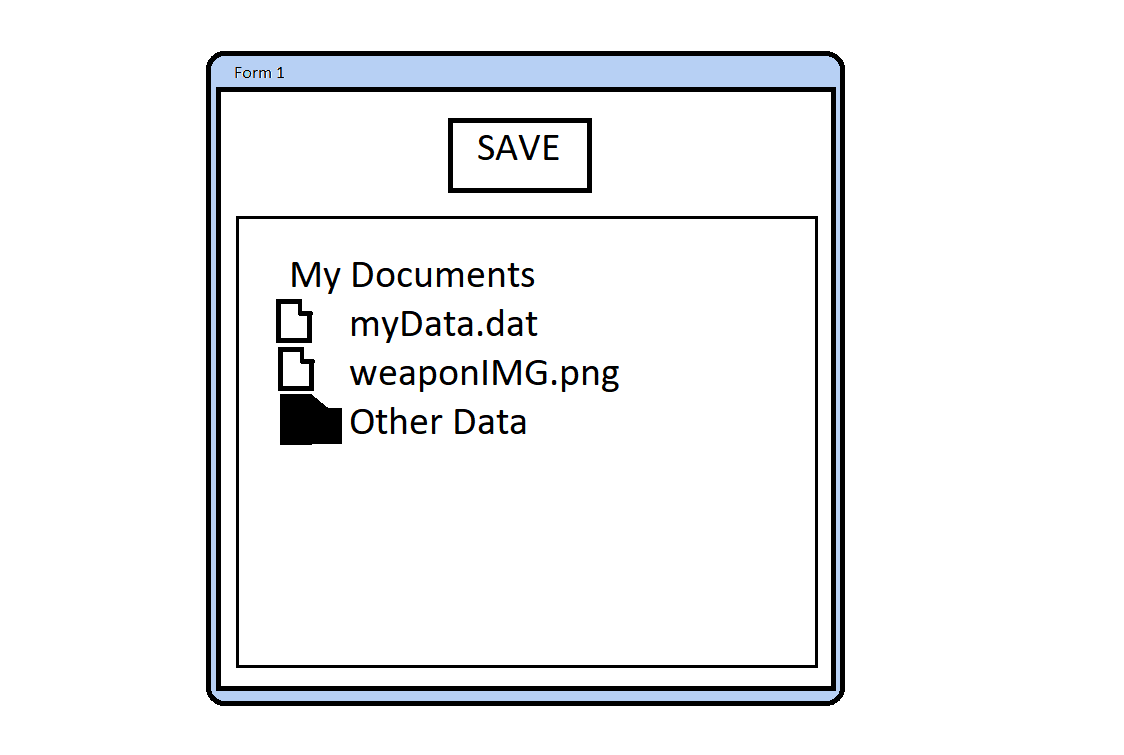
Buttons displayed are the attributes that are accessible to user. The Button “Add Attributes” allows the user to finalize their choice after picking one of the three mock-up attributes. This will guide the user to their next screen in Figure 4.Finalize Weapon ScreenThe weapon screen displayed in Figure 4. allows the user to finalize the weapon created solely based on the user’s desire.

Weapon Screen   


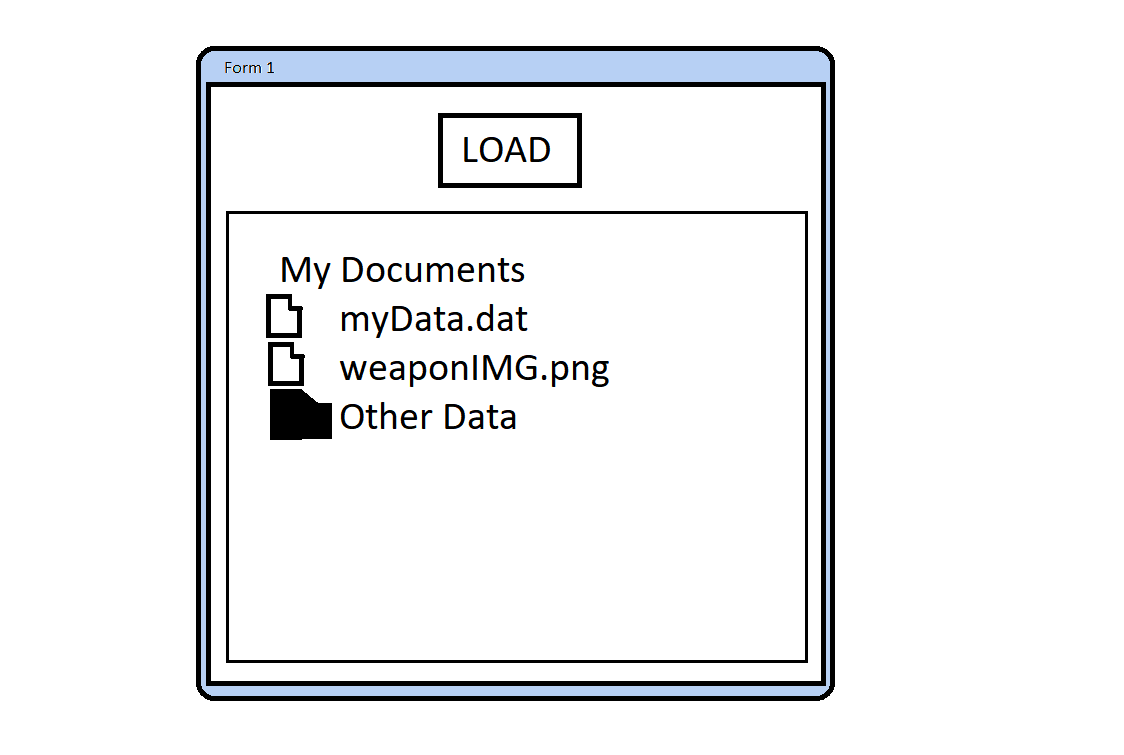
**Figure 4. Displays Final Created Weapon**

Buttons displayed as “Example” will be editable so that the user can manually write their desired name for the weapon. The button “Add Weapon” allows the user to finalize their weapon and add it to the weapons list in the Main Menu referred in Figure 1. The user may choose to save their entire load out which will lead them to the Save Screen referred in Figure 5.Dialogues

Saving and Loading dialog boxes are displayed when the save or load buttons (respectively) are pressed. They will be pressed at a further time in the tool, as shown in Figure 3 Save dialog box, and Figure 4 with the Load dialog box.

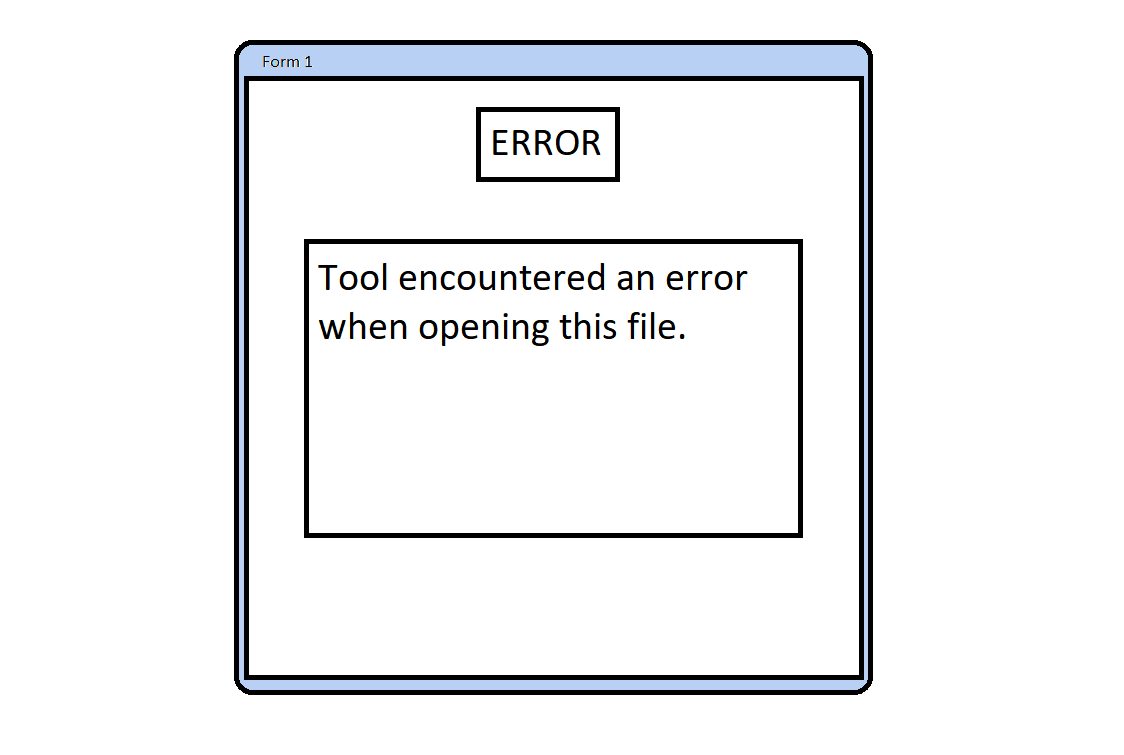


**Figure 5. Save Screen When Saving A Weapon’s Image and Data**



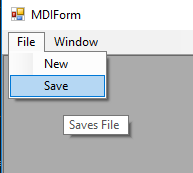
**Figure 6. Shows Loading in Weapon Attributes, and Weapon IMG**

If the user tries to load a file that is with incorrect format or corrupted the Error dialogue box shown in Figure 4 will be displayed.



**Figure 7. Error displayed when loading incorrectly**

Help File Implementation  
The help file is in each interactable object displayed to user, when hovered over will display helpful text.



**Figure 8. Help File Text Displayed**

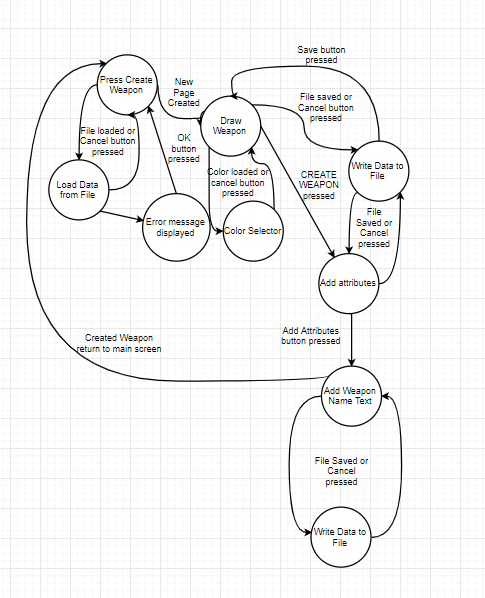
Drag and Drop Implementation  
The Drag-and-drop function can be accessed in the create weapon stage reference in figure 2. The user can drag a PNG image into the allocated draw area and use it as a backdrop, or as reference.

Array Use Implementation  
The arrays will be working with weapons stored and created. The weapons will be sorted each time a weapon is created, by their name in alphabetical order. The user can click on “Weapons” referenced in Figure 1. to search for a specific weapon by name.

2D Graphics Rendering Implementation  
The background of main menu will start light blue. In creating the weapon phase the save screen will start red, until pressed making it turn green. This will indicate that the user can move on to the next phase of the tool.

Data Structure Implementation  
The json saved file, will start with the variables of name and attributes, then the serialized format of an image. The “Create a Weapon” Phase will feature a save button, saving the drawn image as PNG in another folder. After creating attributes and saving the name the entire data including the image will be serialized into a json format file allocated as referred above.

State DiagramThe State Diagram shown in Figure 8. shows the various program states and the transition between them.  
  
The user will spend their time creating, and adding attributes, while transition between programming states when saving. The transition between the main screen to the create, add attributes, display and then back to the main menu when added to list. The User may then load the weapon created with its set attributes of their choice.

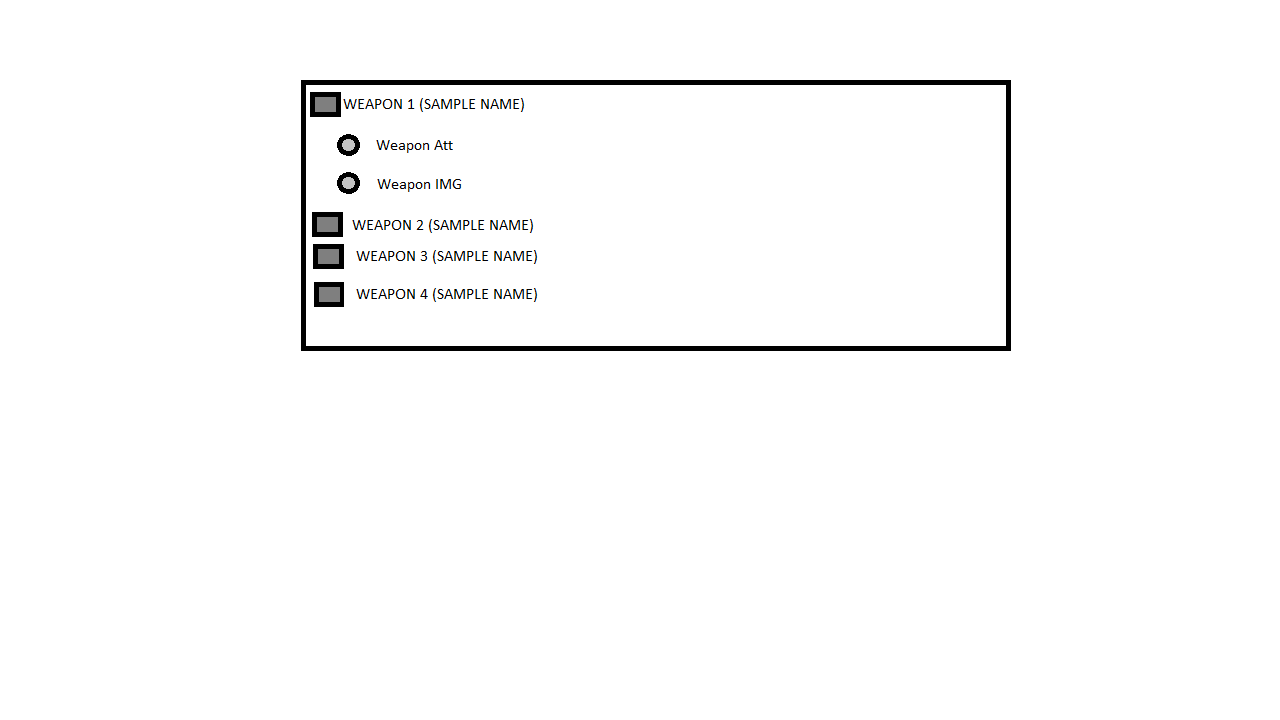


**Figure 9. The State Diagram Displayed**

# Save Data Format

Data will be saved in a json type format

The first entry will include the IMG file data, then weapon attributes data, followed by text saved in format of .csv.

An example of a data file saved by this program is shown in Figure 9.  
  


**Figure 10. Saved data format.**