Sam Leitch – Create Your Own Documented Framework TDD

07.04.2021

—

Sam Leitch

Changelog

Version	Date	Changes
1.0.0	15/03/2021	Initial Setup
1.0.0	18/03/2021	Basic structure in place
1.0.0	25/03/2021	Begin testing allozymic side of things
1.0.0	04/04/2021	Editor manipulation

Contents

Introduction

Our Framework will be known widely as the **StatsFramework.**

Rationale

We are looking to create a ready to use drop in stats system for use in any game requiring basic info to be attached to a player or NPC.

Background

Stats and the statistic of characters harkens back to board games and tactical RPG like dungeons and dragons. They are set in place to show secularities and abilities such as magic of defense. They allow a player to hone in on there favourite aspects of a game if there is an adjustable stats system.

Terminology

An RPG is a role-playing game. Harkens to listen, recall or remember.

Non-Goals

An experience system would closely relate to the stats system. Not all games require an experience system for leveling up but it would be great as a stretch.

Proposed Design

A script that can be attached to a character and individual strings with names and descriptions can be added as well as a Int value to represent that stat itself.

Software and Hardware Requirements

Any computer capable of running unity and visual studio.

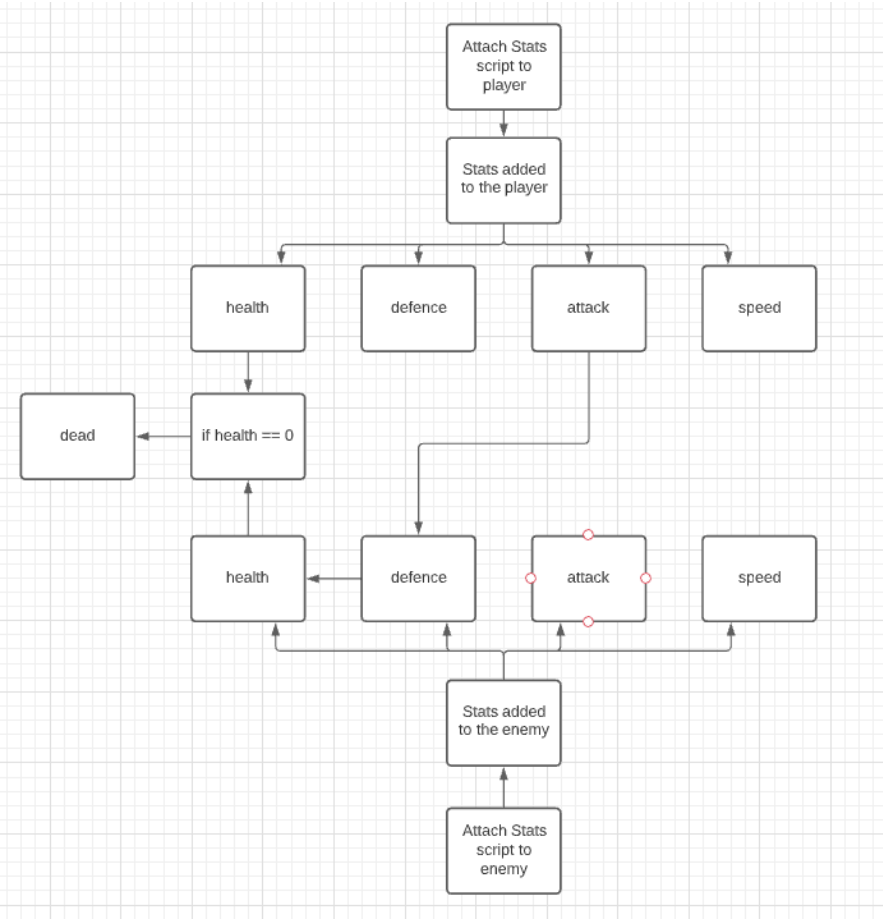
Gameplay

Gameplay Mechanics

Mechanic #1

A stats system can be attacked to be player and referenced when looking for damage calculation or skill checks.

System Architecture



In the example given both characters have a stats system attached, and the player is attacking the enemy. Using the players attack – the enemy's defense is the amount of HP that will be removed from the health system.

Data Model

The data is stored as an Int in the character itself.

Risks

If incorrectly set up all characters that share the same scrip will take damage.

Research and Describe

Sorting Algorithms

Selection Sort – the selection sort is one of the simplest sorts available to programmers. This sort moves through the array and finds the lowest number and swaps it into the appropriate spot at the start of the list. This works with the smallest number being placed as index 0, the second smallest being placed index 1. Big O complexity of 0(n2)

Bubble Sort – Bubbles raise to the top of the carbonated drink; the larger numbers raise to the top of the bubble sort. Bubbles are one of the least efficient sorting algorithms and a recommended to be ignored for any larger arrays. It moves through the list and compared adjacent values and swaps them if they are not in the correct order. Big O complexity of 0(n^2).

Insertion Sort – an insertion sort is a simple algorithm commonly used for a small list or elements. It compares the current key element with the previous element, if the element previous is greater you would move the previous element to the next current position and continue through the list. Big O complexity of 0(n^2).

I will use the Insertion Sort in my lists as I only have a small number of items.

Search Algorithms

Linier Search – In a linier search the algorithm starts on the left on the array and one by one checks to see if it has found a match. If it finds the required element it can return its location or it can return “element not present” if not found. Big O complexity of 0(n).

Jump Search – is a search algorithm for sorted arrays, the idea is to not have to check as many elements, to speed up the process. This search jumps forward by fix steps until it finds or passes its desired target, it them backtracks if it has passed it. Big O complexity of 0(√n).

Binary Search – this search also requires a sorted array, it starts to repeatedly halving the array, it takes I the whole array and finds the middle. If the element is higher or lower than the middle it knows which half of the array to remove. The Binary search continues this until it has the appropriate element. Big O complexity of 0(log n).

I am a large proponent of the binary search as this concept aligns with my philosophy on life. Find the middle and ditch the half that doesn’t match up!

Tools for API Documentation

Using tools for automatically prepare API documentation can have a resounding effect on the onboarding process and aiding staff and clients that do not need to know the nitty gritty of the complete programming process and procedure. It also is a great resource for current programmers to quick reference their own code. Below we look at a number of the best Auto API documenting software.

Swagger UI – is fully customisable to the users and access to the full source code is provided so clients are able to tweak Swagger to their own specifications. It works with OpenAPI specification version up to level 3.0 and because it is so popular there is a fantastic customer support and user base to bounce questions off.

ReDoc – is free and like Swagger is completely open sourced so it can be tweaked, it also supports OAS 2 and 3. ReDoc can be run from in browser or if preferred it can be run from Docker image, a React component or as a command line tool.

OpenAPI Generator – is marketed as an easy-to-use auto generating documentation tool for OAS 2 and 3, as well as server stubs and libraries. Known for its simplicity and its ability to be highly extendable. It also has a vast and wide userbase who are very helpful to all levels of knowledge and intellect.

Debugging

Print debugging - is the processing of watching the code in real time or possibly through recording, it allows using trace statements to look for errors and trace the codes flow.

Remote debugging - is the process of watching and debugging the code from a device other than the one the program is running on.

Post mortem debugging - is the process of debugging the code after is has already crashed.

I was unable to get the test scene working for the framework as I could not get the enemy's health to sync with the health bar I had created. I will rectify this in the future.

There are two other scenes in the project, one has the Insertion sort program and the other has the search.

Resources

https://www.freecodecamp.org/

https://www.geeksforgeeks.org/

https://www.elprocus.com/