Robert W. Herber

Las Vegas, NV | (678) 850-0756 | rwherber@gmail.com
rwherber.com | github.com/rob893

Personal Projects

WoW Auction House Watcher | Web Application | https://github.com/rob893/wow-market-watcher

- Demo URL: https://rob893.github.io/wow-market-watcher-ui
- Developed a service and accompanying progressive web application (PWA) that tracks auction prices for World of Warcraft that allows users to create watch lists and alerts based on price movements
- API built to RESTful standards (including the use of JSON patch documents for patch requests)
- MySQL and Entity Framework used for a code-first approach to database creation/updating
- .NET Core Identity and JSON Web Tokens used for user identity, authentication, and claims
- Hangfire used for background job processing to collect auction data and clean old data
- Azure Event Grid used for event-based alerting
- Twilio SendGrid used for email alerts
- Hosted using Docker and Azure

Fireheart - Action Role Playing Game | 3D Unity Game | https://github.com/rob893/RPG-Game-Scripts

- Website: https://rwherber.com/Fireheart/index.html
- Developed (using C# and the Unity game engine) a 3D action role playing game where the player can interact with game objects, fight enemies using a complex ability system, complete quests, acquire new equipment, and explore a hand-crafted 3D open world
- Utilized the state design pattern to implement non-player character behaviors. NPCs can search for specific targets, patrol areas, chase targets, attack targets using various abilities, follow and protect the player, and reset their states when needed
- Implemented a questing system that allows for quest progression, quest chains, and rewards
- Developed an ability system using the strategy pattern where the player can utilize many different abilities each with their own unique behaviors and effects

Entropy Game Engine | Game Engine | https://github.com/rob893/Entropy-Game-Engine

- Demo URL: https://rwherber.com/entropy/sample-game-1/
- Developed a fully functional 2D game engine that can be used to build many different types of 2D games
- Modeled after the Unity game engine's architecture (component-based game objects, scenes, etc.)
- Implemented computer path finding using the A* algorithm backed by weighted graphs, priority queues, and binary heaps
- Implemented collision detection using spatial hashing
- Built a system where the developer can programmatically describe the shape of a room/map and point to specific sprites on a sprite sheet and the engine will take that data, slice up the sprite sheet, build the described room with the smaller images, and then reassemble them into a single HTML image element for ready rendering
- Implemented positional logic using 2D vector math
- Built a custom system for input and event management
- Built two sample games using the engine. One to demonstrate pathfinding and general engine features and another to demonstrate the physics engine
- (Poorly) Tested using the Jest JavaScript/TypeScript testing framework

TypeScript Extended Linq | NPM Library | https://github.com/rob893/typescript-extended-linq

- Developed a library for JavaScript/TypeScript modeled after .NET's System.Linq with additional features from the popular MoreLINQ library
- Published the library to NPM for anyone to use

Workout Scheduler | Web Application | https://github.com/rob893/Workout-App

- Currently developing a web application that can be used to schedule workout sessions
- Built a GraphQL layer using Apollo Server (NodeJS) and TypeScript hydrated by the REST API
- API built to RESTful standards (including the use of JSON patch documents for patch requests)
- MySQL and Entity Framework used for a code-first approach to database creation/updating using migrations
- .NET Core Identity and JSON Web Tokens used for user identity, authentication, and claims

First Person Shooting Game | 3D Unity Game | https://github.com/rob893/DesignPatternsGameScripts

- Developed (using C# and the Unity game engine) a 3D first person shooter game where the player must shoot monsters and find supplies to survive an endless onslaught of enemies

It's Another Clue | Web-Based Riddle Game | https://github.com/rob893/AnotherClue

- URL: https://rwherber.com/another-clue
- Developed a web-based game where the player must answer five riddles in a row to win. The possible answers for each riddle are displayed as a list of 12 images: all with Nicolas Cage's face!
- The system randomly pulls from a pool of 16 riddles. The system keeps track of which riddles the player has seen in the current playthrough so that the player will not see the same riddle twice in each playthrough
- The system will randomly pull from a pool of 30+ images to create the list of 12 possible answers for each riddle (11 images chosen at random and the answer image). The system ensures that no two images are the same for a given riddle answer pool

WoW Guild Sales App | Web App | https://github.com/rob893/OrderManagementSystem

- Developed (using PHP, Bootstrap, SQL, JavaScript, and jQuery) a web application that allows the user to add sales events, customers, services to be sold and their associated costs, create groups of players for each sales event, track amount of in-game currency owed to the group from the buyers, how much the buyers have already paid, and how much is owed to each player. Data is driven by a local MySQL database, Blizzard's Battle.net REST API, and Warcraft Logs' REST API
- Integrated Blizzard's Battle.net REST API to automate populating and updating the player database
- Integrated Warcraft Logs' REST API to automate calculation of total in-game currency owned to each player based on who participated in each sold event. The system also filters out players who purchased the service from the calculations based on data from Warcraft Logs and the local player database
- The system ties together all a player's characters in the database so that reports will show currency owned to that player instead of being split between each character that player played for the sales events

Restaurant Suggester App | Web App | https://github.com/rob893/LaravelWDYWTE

- Developed (using PHP, Bootstrap, the Google Maps REST API, and the Laravel framework) a web application that will suggest a random nearby restaurant for indecisive individuals
- The system allows the user to refine his or her search by searching by keyword and/or by distance
- The system caches the results from the API request which allows the user to cycle through the results with minimal API calls (a single Google Maps query can return up to 60 results split into 3 pages. If more than 20 results are returned, the system uses the next page token to retrieve results 21 - 40 and 41 - 60 for a total of 3 API calls per 60 results)

Side Scroller 2D Android Game | Mobile App | https://github.com/rob893/PoliticalPunchout

- Developed (using Java and Android Studio, no game engine) a 2d side scroller game where the player dodges falling objects and earns points by shooting down incoming enemies
- The game has infinite levels where each level is increasingly more difficult than the last