Overview of Selected Projects

Kelvin Beltre

This overview will take you through some selected projects, showcasing features and integrated systems. Each page will go through a project's focus, the design philosophy and applied methods, and some interesting observations from my time working with the project. All featured projects can be found at github.com/Bluechacho with repos linked below.

Project Focuses

Rapidly-exploring Random Trees

Board Game Collection 2

Travel Reservation Systems & Database

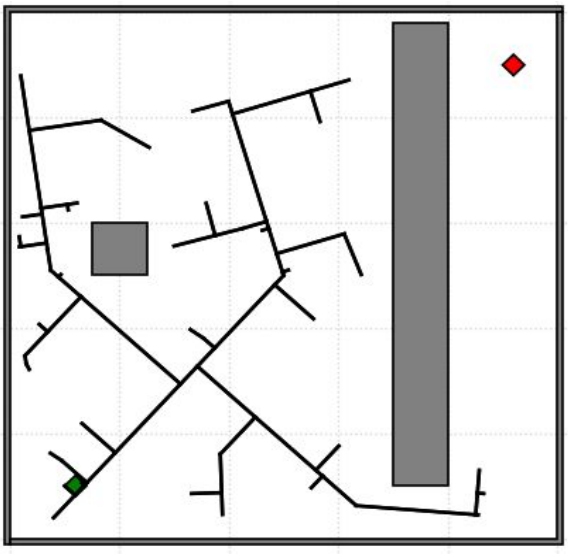
Rapidly-exploring Random Trees

Python Focus

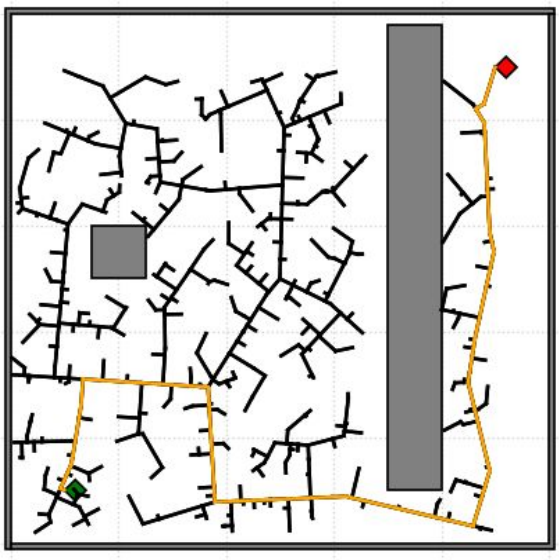
Philosophy: Creating a rapidly-exploring random tree to connect two points on a graph

Methods: Nearest neighbor connection, uniform cost search using Dijkstra's algorithm, collection checking along tree branches

Observations: Within a bounded graph, we can generate valid points with low dispersion for dense coverage, connect *k* nearest neighbors, and connect a start and goal point to the generated RRT. Generating more points leads to a more robust RRT, capable of "traversing" around even the widest obstacles to connect start and goal.



50 point RRT



500 point RRT

https://github.com/Bluechacho/rapidlyExploringRandomTrees

Board Game Collection 2

Angular Focus

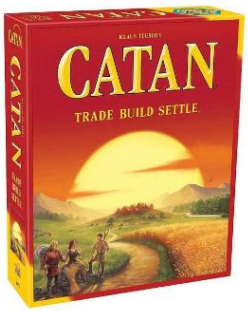
Philosophy: Create a single-page application demonstrating a modular display page using AngularJS

Methods: AngularJS application and controller implementation, CSS flexbox display, multi-stack structure

Observations: Angular works really well with indexed lists: the controller will read the stack at the given index, format the data as requested, and output all the information via the application. Then we can easily modify the data or the order by manipulating the stack - the controller will tweak the HTML and CSS to match.

```
$scope.currentIndex = 5
$scope.name = Catan
$scope.theme = Medieval
$scope.playerCount = 2 - 4
```

displayInfoCtrl (input)



displayInfoApp (output)

This is **Catan**. It is a **Medieval** themed game, good for **2 - 4** players. Here is a short description:

https://github.com/Bluechacho/boardGameCollection2

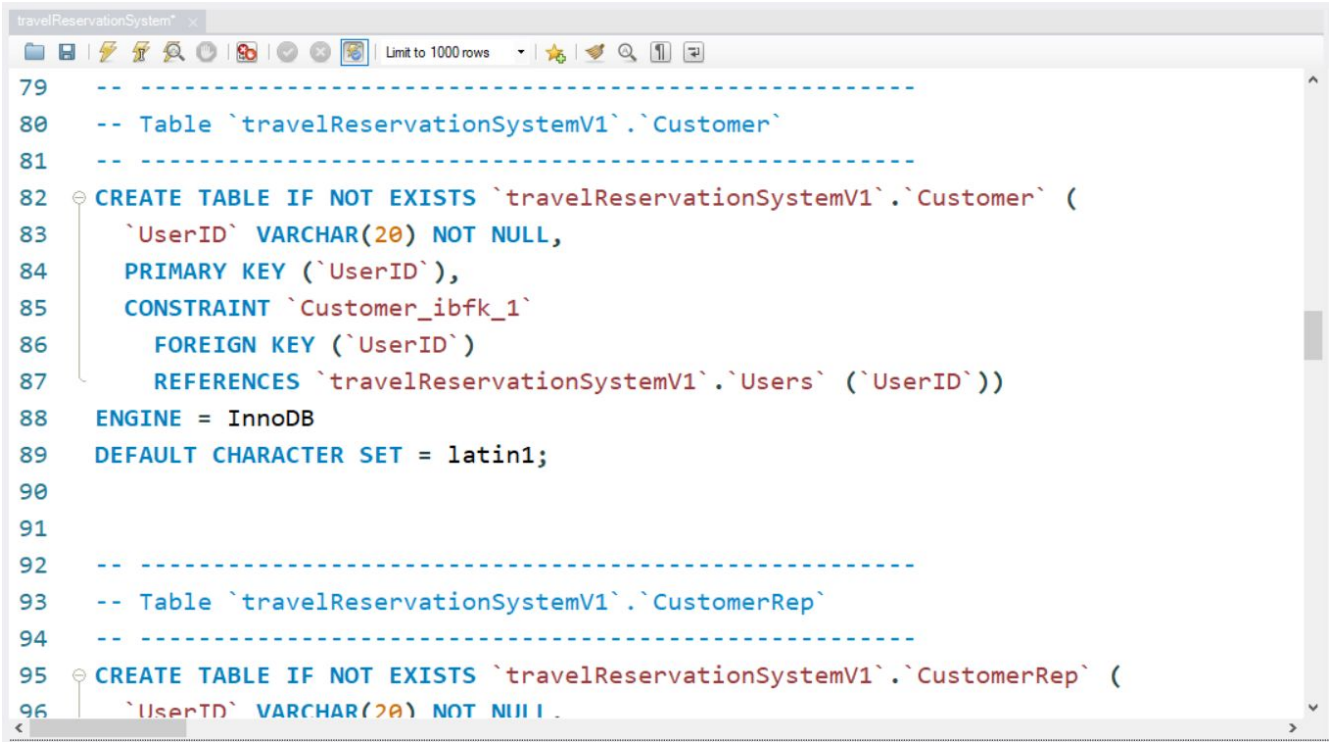
Travel Reservation Systems & DB

SQL Focus

Philosophy: Implement a travel reservation database to create, read, upload, and delete (CRUD) accounts, flights

Methods: Apache Tomcat JSP deployment, mySQL Workshop access, Amazon Web Services utilization

Observations: Amazon Web Services is sufficient to host and access a travel reservation database – the data is manipulable on many different access levels, depending on the user's level of clearance. Certain accounts are permitted from writing or reading information based on if the user is a customer, representative, or server admin.



https://github.com/Bluechacho/travelReservationSystem