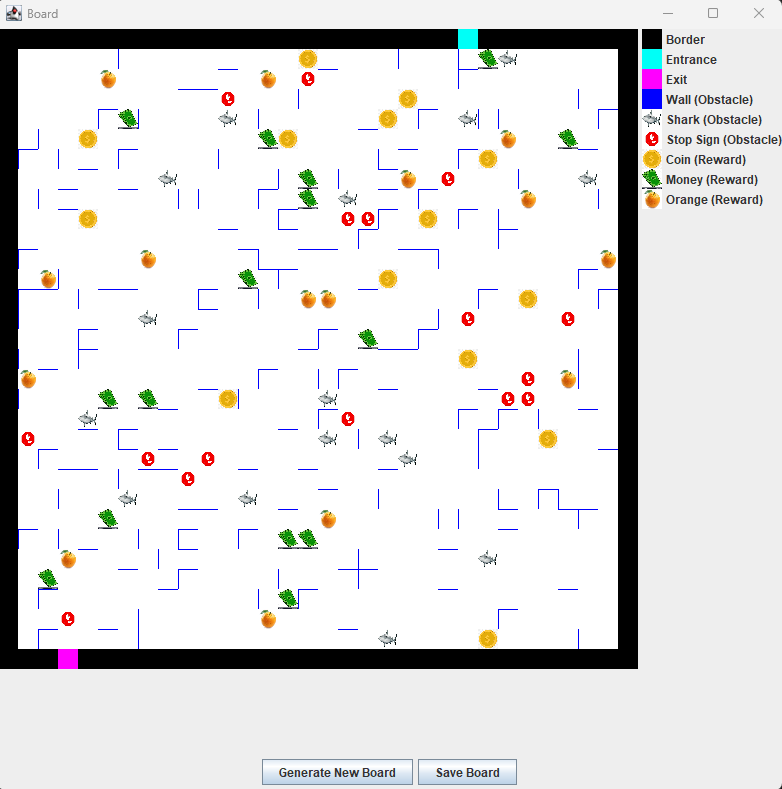
Miles Davis

Ethan Dolbear  
  
The Object-Oriented (OO) design implemented in our application is structured around three main classes: BoardApp, BoardObject, and BoardPanel. The BoardApp class serves as the entry point and orchestrates the graphical user interface using Java Swing components. It creates a JFrame, sets up a JMenuBar with "File" menu options, and responds to user actions by generating a new board or initiating a board loading process. The BoardObject class is an abstract class with an enum defining different types of board objects, providing a foundation for diverse objects with the commonality of being drawable. The BoardPanel class, extending JPanel and implementing Serializable, encapsulates the core logic of the board. It contains a 2D array of BoardObject cells, utilizes image resources to represent objects, and handles the initialization, generation, and rendering of the board.

Throughout the design process, considerations were made for extensibility, flexibility, and encapsulation. The abstract BoardObject class allows for the easy addition of new board object types, promoting a modular and scalable design. The use of interfaces, such as Drawable, facilitates a standardized approach to rendering objects. The BoardPanel class's methods are structured to promote reusability and readability, with a focus on generating a random but balanced board configuration. The design also leverages Java Swing components for a responsive and user-friendly GUI.

**PICTURE BELOW  
|  
|  
|  
|  
|  
|  
\/**



The group: Ethan Dolbear, Miles Davis

We faced a few different issues during the process. Everything from board generation to proper serialization. Our trial-and-error approach to the ordeal ended up working for the most part. This culminated in a working prototype of our board generation.