Group Member:  
  
Andry Canel

Creating the Minecraft-inspired Gameboard involved using a smart, organized method based on objects. I grouped specific tasks into abstract classes and interfaces, making the design strong and well-structured. The Drawable hierarchy took care of dealing with images, while the GameObject hierarchy expanded these capabilities for obstacles and rewards. The Cell class played a crucial role in placing objects and rewards on the game board, keeping everything organized and easy to modify. I tackled issues with saving and loading by turning images into byte arrays. The Gameboard class brought it all together, making it easy to design the board and manage saving and loading. The whole process was guided by a solid understanding of object-oriented principles, ensuring the design is flexible and can grow as needed.

The journey began by identifying the game's key elements: objects you can draw, obstacles, rewards, and the cells on the game board. I used smart ideas like abstraction and encapsulation to create abstract classes and interfaces, organizing shared tasks and keeping everything well-structured. The Cell class became crucial, managing where things go on the board and handling rewards. Serialization, a tricky part, was tackled by turning images into byte arrays. I used Swing components for a user-friendly interface. The process involved refining game rules, dealing with serialization challenges, and sticking to object-oriented principles. The result? A Minecraft-inspired Gameboard that blends design principles with practical game development.

Gameboard

|

|---Cell

| |

| |---GameObject

| | |

| | |---Obstacle

| | |---Reward

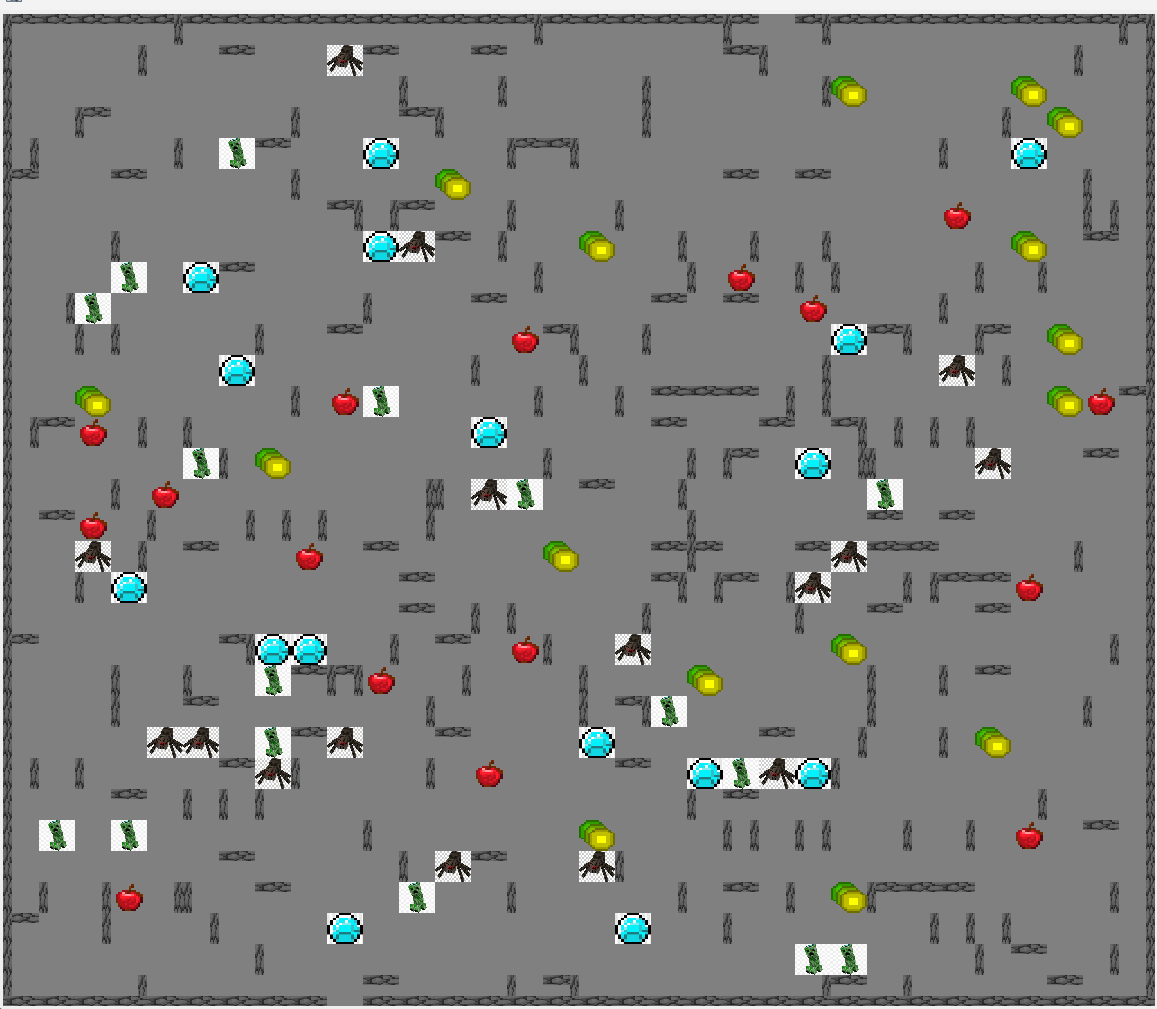
| | |---Wall

| |

| |---Drawable

| |

| |---DrawableInterface



Before I added better images

