Michael Ceryak [63909655]
Jacob Early [41786758]

Space Explorer Project Report

I. Structure

A. Design

1. We implemented a Crew class which initializes the Crew object. This object has a name, HashMap of CrewMember objects, HashSet of integers representing the abilities of the crew members in the crew, as well as a SpaceBus object and Inventory object. CrewMember, SpaceBus, and Inventory objects all contain their own gui object along with other attributes. A CrewMember object has name, type, health, max health, hunger, tiredness, actions, and plague attributes. Within the Inventory object are HashMaps for food items and medical items, as well as the CartmanCoin object and InventoryGui object. The SpaceBus object has a name, shield health, missing pieces, and Action object.
2. The ActionSet class utilizes the Singleton OOP concept. It has a CrewMember, a HashSet of Action objects, a boolean telling if a panel is open, and a single instance. An Action object has a JLabel, a CrewMember, and a performAction method. When a CrewMember object is selected, the label of an Action object becomes available and this CrewMember object becomes part of the Action object.
3. We also implemented a CreateRandomEvent class which calls one of three RandomEvent objects. Each random event object implements a method which takes a Crew object and modifies it in some way
4. The GUI is controlled by one main class called GUIManager. It creates the Crew object used throughout the game, as well as the single instance of ActionSet. This class opens and closes windows in order as they should in the game. These windows control that each event that happens is rendered to the screen.

B. Inheritance

1. 6 types of crew members extend abstract CrewMember class. 3 types of medical items extend abstract MedicalItem class. 6 types of food items extend abstract FoodItem class. Abstract classes MedicalItem and FoodItem implement Item interface. Each action is an instance extending abstract Action class. Each random event class implements RandomEvent interface.

C. Collections

1. HashMap of CrewMember objects in Crew class
    a) Map: CrewMember cm -> Boolean isAlive
2. HashSet of Ability Identifiers in Crew class
    a) Contains integers representing certain abilities of crew members in crew
3. HashMaps of Food/Medical Items
    a) Map: Food/Medical Item item -> int quantity
4. HashSet in ActionSet class
    a) Contains each Action object in project

II. Thoughts and Feedback

A. Went Well

1. The project is well organized and coded efficiently by utilizing many different OOP concepts. We were able to create a very pleasing GUI for the game in the given time constraint. We had great communication and no issues with dividing up the work. Github made it really easy to work together.

B. Went Poorly

1. We had trouble totally separating game logic from GUI logic. Since, the observable class has been deprecated in Java, we had to come up with other ways of performing events when game logic requires.
2. It was also a challenge to set up our code so that it runs correctly on the lab computers. We had to modify sizes of Swing components that caused the app to look strange on our computers so that it would fit and look fine on the lab computers. We also had to downgrade the version of Java used to build the program since the lab computers run an older version of Java.

C. Improvements for next project

1. It may be helpful to write some of the JavaDoc before coding, so that we know for sure how things are going to work and connect before we start coding.
2. A better timeline of what we were going to do and when would have helped with organising ourselves early on.

III. Effort

A. Hours spent

1. Michael Ceryak: 35
2. Jacob Early: 20

B. % Contribution to project

1. Michael Ceryak: 60
2. Jacob Early: 40