**CSC 335 Analysis and Design Artifacts for Pokemon**

*Each team complete this form, put it in your project in a folder named* **doc** *and push to Github. This will be part of your Iteration 1 grade*

**1. Team Name:**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

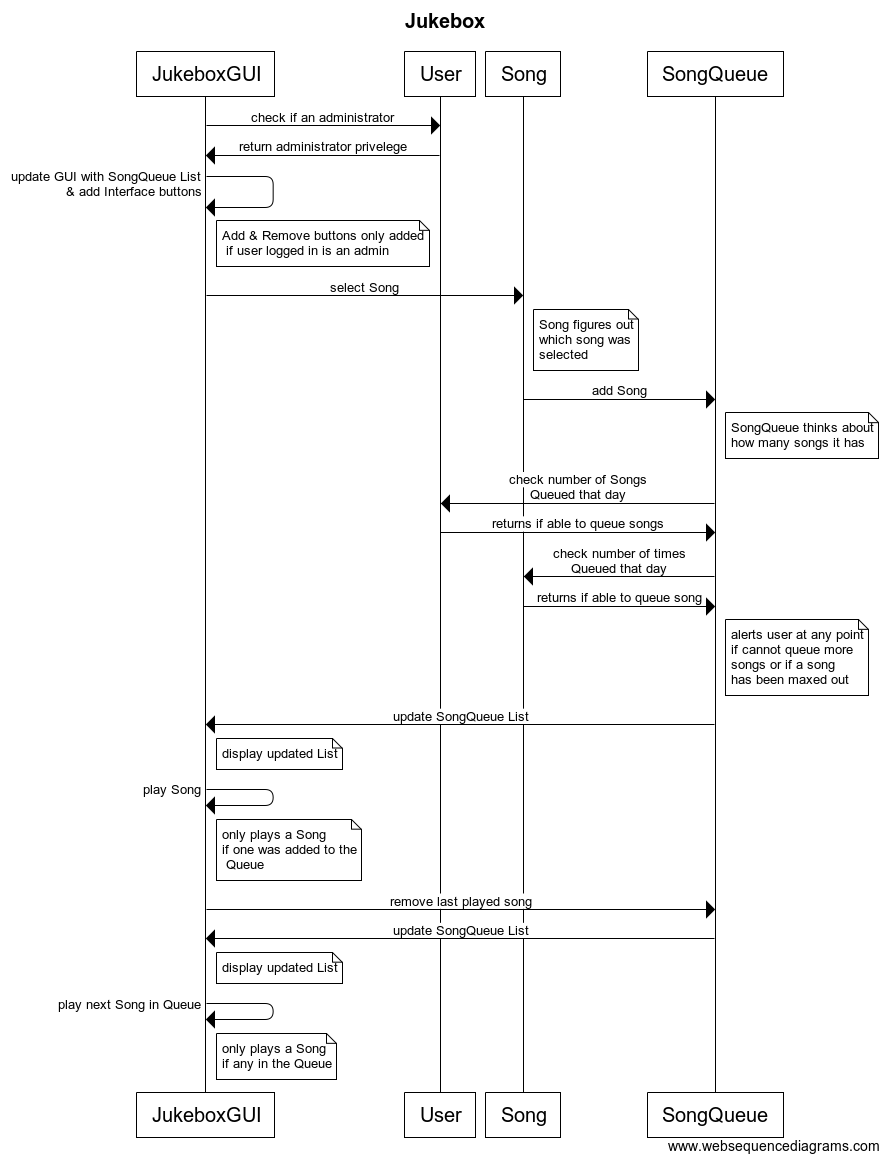
**2. Project (Risk, Pokemon, or Tower Defense): \_\_\_\_\_Pokemon\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**3. Team Members**: \_\_\_\_\_Gavin Daniel\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_Abdullah Asaad\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

List the most important objects, or an inheritance hierarchy name, and the responsibility of each.

|  |  |
| --- | --- |
| **Candidate Object** | **Responsibility in 1 or 2 sentences** |
| PokemonGame | Keeps track if the running game, number of players in the game, and the current GUI View. |
| MapView | Shows the viewable are of the map within the map region that the Trainer is in. Displays all objects and their images and gets this information from Map. |
| CaptureView | Shows the view of the Pokemon vs. Trainer, actions in the battle, the pokemon and trainer involved in the battle, and animates all actions of the trainer or pokemon. |
| BattleView | Shows the view of the Pokemon vs. Pokemon, actions in the battle, the pokemon involved in the battle, and animates all actions of the pokemon. |
| Animation | Holds all relevant information and functions related to any animations involved with the Game. Depending on the Object that needs to be animated this class will determine the animation necessary and execute that animation. |
| Map | Overseeing map that keeps track of every Tile in the game and its location along with the player location and pokemon locations. |
| Tile | Holds all information related to each individual Tile on the Map in the Game. This information includes things like background image, whether a pokemon is hiding there, and if the player is currently on that Tile. |
| Trainer | Holds all information related to the Trainer such as Gender, their List of Pokemon, List of Items (Pokeballs, Revives, etc.) , number of Steps taken, direction facing and direction they are moving in. |
| Pokemon | Holds all information related to the Pokemon such as Type, HP, Moves, Rareness, Chance of being Caught. |
| Move | Holds all information related to Pokemon Moves such as Type, Damage Amount, PP (number of uses left, number of uses) |
| Item | Holds all information related to items such as Type (Pokeball, Rock, Revive, Bait, etc) |
|  |  |
|  |  |
|  |  |
|  |  |

**3) Sequence Diagram:** Write a UML Sequence Diagram should show the most important scenario you can think of. Your sequence diagram should show most of your candidate objects you listed above and how they communicate with each other.

**4) Class Diagram:** Write a UML Class Diagram that shows all of your candidate objects from above. Show any relationships between them the classes such as inheritance or interface implementation. Draw general associations such as dependency or aggregation. Label some to help explain things. Add any multiplicity adornments that seem appropriate. Use notes to explain things if you feel it will help. Each UML class must show the class name. For full credit, each class must have an average of at least one attribute per class. There must be an average of about 1.5 methods per class.

**Optional:**  *Instead of using pencil, paper and scanning to insert an image, use a new Eclipse Project to add classes, methods, and instance variables. No implementation needed. Then use Object Aid UML tool to reverse engineer your code to get the UML class diagram.*

*Installation of Eclipse Plugin in case you don't have it:*

<http://www.objectaid.com/installation>

*Export and Image and copy and paste it here*

**5) Estimate and Assign Tasks** For each Iteration 1 task, estimate its difficulty using the numbers 1, 2, 3, 5, or 8. These are points that represent the relative complexity of the task. Mark 8 for the most difficult and/or time consuming and 1 for what appears to be the easiest. Indicate which person (s) will complete the task before iteration 1 due date.

|  |  |  |
| --- | --- | --- |
| **Points** | **Who will**  **complete this?** | **Task** |
|  | Everyone | Your GitHub repo has a completed copy of this analysis and design document in a folder named **doc** that represents a serious effort to analyze and design Jukebox |
|  | Gavin & Giulio & Abdullah | Main Menu GUI where Users can create a Player or choose from Saved Player Files. |
|  | Gavin & Giulio | A Map View of the viewable area where a Trainer is located is displayed. User can move around on the map using the arrow keys and interact with objects on the map based on their location. |
|  | Gavin & Giulio | GUI switches Map View when a User walks to the entrance to the next Section of the Map. |
|  | Gavin & Abdullah | Create Library of Pokemon (at least 15) |
|  | Ian & Abdullah | User can run into Pokemon and initiate a Battle Scene. In Capture Scene Trainer can try to capture, bait or run away from the Pokemon. Pokemon will react and try to escape, get caught, breakout of pokeball or be baited. |
|  | Abdullah & Ian | Trainers can run into each other if they are in the same MapView and battle each other if they both have (at least 1) Pokemon. Battle Scene is initiated and both Trainer’s first Pokemons will be displayed and their options to fight, use item, switch Pokemon, or run is displayed and actions are called based on which is selected. |
|  | Ian | Menu View displays Items, Pokemon for a given Trainer during Map View. |
|  | Giulio | Serialization of Data for Persistence. |
|  | Everyone | Determine when Game ends if User has used up all available steps in the Safari Zone or runs out of Pokeballs. |