CS 2340 Spring 2020 – Milestone 4: Project Iteration 2

Universe Creation, User Stories, and Sequence Diagram

BACKGROUND: For this iteration of the project, your team will need to begin to design and implement the world that your game will take place in. User stories are simple statements which describe typical use cases of your game. Sequence diagrams take user stories and provide implementation-specific details on the process.

PURPOSE: Planning is key when designing large systems. This milestone expands upon the planning started by M3 through the creation of more diagrams. These diagrams are intended to help students when they are deciding how to lay out their code. In industry, software engineering teams are expected to design diagrams regularly for planning and demonstration to clients.

TASK: This milestone has two team design deliverables, a required feature set for implementation, and a demo which will occur the week after milestones are due.

User Stories (Team Deliverable)

1. As a team, create 5 unique user stories for your project based on the implementation requirements.
    - You may follow the following format:
        - As a *user role*, I want to *goal* so that *benefit*.
    - Example:
        - As an administrator, I want to be able to add developers to projects so that each developer is able to self-assign themselves to issues.

Sequence Diagram (Team Deliverable)

1. Each team member should take one of the above created user stories and turn it into a UML sequence diagram.
2. For your sequence diagram, ensure that it illustrates the dynamic interactions between the objects in your project's design.
    - You should use some of the objects you brainstormed for M3 and used for the context diagram.
        - Example: Player, Region
    - For the GUI and other user input, you may represent it with an appropriate "class" name.
        - Example: CharacterInterface, InventoryScreen
3. You are required to use at least one ALT or LOOP structure within your sequence diagram, so choose your user story carefully!
4. Combine all sequence diagrams into one pdf

Implementation Requirements
You are to create a view of the universe. Your universe is composed of unique regions which your player can travel to. When the game begins, at least 10 regions should be initialized in the universe with random coordinates. Each region should have a unique name, a technology level, and a brief description, all of which are not visible to the player until they visit that region for the first time. The technology level will be used in a later milestone for marketplaces. After character creation, the player then randomly spawns into one of these regions.

The user must have some way to view the list of the existing Regions in the UI of the application. This can be as simple as a panel that you can cycle through the Regions with by clicking "Next Region" or as complex as a map of the current universe (for EC).

While a player is in a region, a display of the regions characteristics and marketplaces should be shown to the player. For this milestone, however, the marketplace display can simply be a blank canvas.

While a player is in a region, the player should have the option to travel to other nearby or previously visited regions. When a player enters the new region, the new region's display as well as its information should be shown to the user. Before a region is visited for the first time, its information should be "unknown" to the player. After a region is visited to for the first time, its information should be added to the universe view.

Reminder: All implementation details not specified by the rubric are up to you!

Extra Credit
If your game has in game music: 5pts
If your view of the universe is a map accurate relative to region coordinates: 5pts

Checkstyle
During demo your team will be required to run the checkstyle script (located under files>checkstyle>Java Guide.pdf). This script will give your project a score out of 10 and will account for 10 points of your M3 final grade. Be sure to run the checkstyle script prior to submission to avoid unforeseen deductions.

Milestone Tagging
Tags are a way of marking a specific commit and are typically used to mark new versions of software. To do this, use "`git tag`" to list tags and "`git tag –a tag_name –m description`". You are required to tag the latest commit before the deadline which is to be graded during demo. You will be required to pull this commit during demo.

Submission Requirements
In addition to your diagrams, ensure that you include a link to your GitHub repository in your submission. Also, ensure that you have added your grading TA(s) as collaborators so that they

may view your private repository. **Repositories must be located on the Georgia Tech GitHub and must be set to private!** Points may be deducted if these guidelines are not followed!

CRITERIA: You will be graded according to the rubrics on the final pages of this assignment document. Please note, 60 points of the 100 total points are dependent on your group's demo which you will pull the tagged git commit for the milestone to be run on your machine. **Groups are required to demo in order to receive credit for the features they have implemented. If you would like to demo changes made after the milestone due date, you will receive a flat 20-point penalty on the milestone's grade.**

User Stories Rubric (15 points)

5 User Stories Present.	10 points. (2 point for each present user story)
User stories are correctly formatted.	5 points. (1 point for each correctly formatted user story) *Note: While user stories should follow the above given template, it is up to TA discretion whether a user story is correctly formatted/appropriate.*

Sequence Diagram Rubric (25 points)

Sequence diagrams formatted correctly.	15 points. (no points if it doesn't look like a sequence diagram, otherwise -1 point for each formatting error with a max of -3 per diagram)
Sequence of events makes contextual sense with the user story and project.	5 points. (-5/(# team members) per diagram that doesn't make logical sense)
ALT/LOOP Included	5 points. (-5/(# team members) per diagram that doesn't include ALT/LOOPs)

Implementation / Demo Rubric (70 points)

A link to the group's project private GaTech GitHub repository is provided in the submission and all grading TAs have been added.	10 points.
Previous functionality is still functional.	10 points.
Front end displays 10 or more regions in the universe. In any format.	5 points.
Front end displays each region's coordinates and distance from current region	5 points.
Front end displays {name, tech level, and bio} for regions if and only they have been visited	5 points.
Player can travel to other regions	5 points.
Universe is generated unique each new game	5 points.
Player's starting region is random each game	5 points.
Extra Credit: Region display is in map format with regions displayed in relationship to each other's location	5 points.
Extra Credit: In-game music	5 points.
Checkstyle score generated by python script	10 points.