CS 2340 Spring 2020 - Milestone 6: Project Iteration 4

NPC Encounters and Interactions

BACKGROUND: For this iteration of the project, your team is going to build features and interactions around NPCs (Non-Playable Characters). A non-player character (NPC) is any character in a game which is not controlled by a player. In video games, this usually means a character controlled by the computer via algorithmic, predetermined or responsive behavior, but not necessarily true artificial intelligence.

PURPOSE: This milestone is designed to effectively introduce you to the process and aspect of designing major software features. Design and making design choices are the key point and takeaway of this project iteration. This project iteration is heavily focused on design and implementation of iterations and features, and hence there is **no diagramming portion for this assignment.**

As mentioned above, this milestone will focus on creating and implementing NPC Encounters. Design choices for interactions should be made such that different newly added entities, representing different aspects of an *NPC Encounter*, are implemented according to object-oriented design patterns. Think of how you can apply SOLID and GRASP to ensure your code is readable and maintainable. Learning to produce clean code now will save you countless hours over your career.

TASK: This milestone has **no diagrams required**; however, you can still draw diagrams to more effectively make design choices. When a player encounters an NPC, a new screen or pop-up should display the details of the encounter. After the player takes some action, they should be notified of the outcome, and then be able to continue playing the game. There are three types of NPCs that need to be added to the game: **bandit**, **police**, and **trader**. The player can only encounter an NPC while in the process of travelling -- as in interruption that stops the player during travel. During any NPC encounter, the player will be prompted to take some action. Each action that the player can take has a potential benefit and/or consequence. NOTE: Be prepared to force encounters during demo to allow efficient grading.

Implementation Requirements

NPC Descriptions

Bandit: A bandit encounters the player to rob credits (money) from the player. They will demand a certain amount of the player's credits before the player can continue travelling to the desired destination. The player then has *3 options*:

1. *Pay the bandit's demand and continue to the desired destination.* If the player cannot afford the bandit's demands, then the player must give the bandit all the items in their inventory. If the player has no items, the bandit will damage the ship's health. Then the player continues to the target destination.

2. *Try to flee back to the previous region.* The success of fleeing is dependent on the player's Pilot skill **(higher Pilot level, higher chance of escape)**. If the player successfully flees back to the original region, they should still lose the fuel required to travel initially, but they keep all their credits & items and they are safe. If the player fails to flee, the bandit will take all their credits and damage the health value of the player's ship.

3. *Try to fight off the bandit.* The success of defeating the bandit is dependent on the player's fighter skill **(higher fighter level, higher chance of winning)**. Successfully fighting off the bandit will allow the player to travel as intended to the desired destination without any new consequences. Additionally, success will grant the player some of the bandit's credits as a reward for winning the fight. Failing to fight off the bandit will cost the player all their credits and should damage the health of the player's ship.

Trader: A trader will ask the player if they would like to buy a few of a certain item. The player can choose to:

- *Buy the items.* After buying the items, the player continues travelling to the target destination.

- *Ignore the trader and continue travelling to the target destination.*

- *Try to rob the trader for items.* The success of robbing the trader is dependent on the player's Fighter skill **(the higher the Fighter skill, the higher the chance of robbing the trader)**. If the player successfully robs the trader, they get some of the trader's items. If the player fails, the trader will damage the player's ship health.

- *Negotiate for a better price.* Negotiating with the trader will either lower the price of the trader's items (success) or upset the trader (failure). The success of negotiation is dependent on the player's Merchant skill **(the higher the Merchant skill, the higher the chance of winning the negotiation)**. If the player succeeds, the trader's price should be significantly reduced. If a trader gets upset, they will demand a higher price for their items. After choosing to negotiate once, the player cannot do so again, and must decide to buy an item, ignore the trader, or rob the trader.

Police: Police encounter the player because they believe the player has acquired stolen items. Perhaps the player has stolen items from NPCs, or perhaps they purchased a stolen item unknowingly. Whether the items are truly stolen is unknown, but the police will make their demands regardless. If the police encounter the player, they will identify some items in the player's inventory they believe to be stolen and try to confiscate the stolen items. **A police encounter should only occur if the player has items in their inventory**. The player can choose to:

- *Forfeit the items to the Police and continue to the desired destination.*

- *Try to flee back to the previous region.* The success of fleeing is dependent on the player's Pilot skill **(higher Pilot level, higher chance of escape)**. If the player successfully

flees back to the original region, they should still lose the fuel required to travel initially, but they keep all their items and they are safe. If the player fails to flee, the Police will confiscate the stolen items, damage the health value of the player's ship, and force the player to pay a fine for evasion. Then the player returns to the previous region.

- *Try to fight off the police.* The success of defeating the police is dependent on the player's Fighter skill **(higher Fighter level, higher chance of winning)**. Successfully fighting off the police will allow the player to travel as intended to the desired destination, keeping the stolen items in their inventory.

Ship and Player Attributes Update

Remember to update the ship's fuel, cargo, and health accordingly while travelling, trading, and undergoing encounters. Additionally, ship fuel for traveling between regions should be implemented during this milestone. Pilot skill should affect fuel cost for traveling between regions.

NOTES TO CONSIDER:

1. As explained at the bottom of the **TASK** section above, the above functionality requirements may take too much time to demonstrate for all the different cases (such as different skill levels from configuration). Groups should be prepared to show TAs in their code where they implemented each requirement for full points if the requirements are not obviously demonstrated in the front-end of the app.

2. *Reminder:* All implementation details not specified by the rubric are up to you and any functionality not specifically defined above is open to your choosing for implementation.

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 M6 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

There are no 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, *100* 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.**

Criteria	Pts
[General encounters] In game, NPC encounters occur during travel, interrupting travel between regions	3.0 pts
[Bandit encounter] Game difficulty increases bandit encounters Group can demonstrate in game or explain the code for how game difficulty increases bandit encounters (the higher the difficulty, the more frequent bandit encounters are)	3.0 pts
[Bandit encounter] Bandit should demand a number of credits	3.0 pts
[Pay Bandit option] When paying bandit, player should lose the specified number of credits and continue to intended destination	3.0 pts
[Pay Bandit option] If player can't afford Bandit demands but chooses not to flee or fight, then player loses entire inventory, cargo space updated, then continues to intended destination	3.0 pts

[Pay Bandit option] If player can't afford the demand and has no items (and chooses not to flee or fight), their ship health is lowered then they continue to intended destination	3.0 pts
[Flee Bandit option] Pilot skill: Group can explain the code for how Pilot skill affects likelihood of success of fleeing bandit (the higher the Pilot skill, the more likely success)	3.0 pts
[Flee Bandit option] Successful fleeing: Player should be returned to original region. Player will maintain all credits and items, but lose fuel	3.0 pts
[Flee Bandit option] Unsuccessful fleeing: Player should lose all credits and ship health should be lowered	3.0 pts
[Fight Bandit option] Fighter skill: Group can explain the code for how Fighter skill affects likelihood of success of fighting bandit (the higher the Fighter skill, the more likely success)	3.0 pts
[Fight Bandit option] Successful fighting: Player granted additional credits and continues traveling to intended destination	3.0 pts
[Fight Bandit option] Unsuccessful fighting: Player should lose all credits and ship's health should be lowered	3.0 pts
[Space-police encounter] Game difficulty increases space-police encounters (the higher the difficulty, the more frequent space-police encounters are)	3.0 pts

[Space-police encounter] SP demand item(s) for forfeiture	3.0 pts
[Forfeit to SP option] Specified item(s) removed from player's inventory, cargo space updated, and player continues to intended destination	3.0 pts
[Flee SP option] Pilot skill: Group can explain the code for how Pilot skill affects likelihood of success of fleeing (the higher the Pilot skill, the more likely success)	3.0 pts
[Flee SP option] Successful fleeing: Player should return to original region and lose fuel	3.0 pts
[Flee SP option] Unsuccessful fleeing: Player should have specified items confiscated from inventory, cargo space updated, lose additional credits as a fine, and have ship health lowered. Return to the original region.	3.0 pts
[Fight SP option] Fighter skill: Group can explain the code for how fighter skill affects likelihood of success of fighting space-police (the higher the fighter skill, the more likely success)	3.0 pts
[Fight SP option] Successful fighting: Player should keep all inventory items and continue to intended destination	3.0 pts
[Trader encounter] The trader will ask the player if they would like to buy a few of a certain item	3.0 pts
[Ignore Trader option] Player continues to intended destination	3.0 pts

[Buy Trader option] After buying the items, the player continues travelling to the target destination.	3.0 pts
[Rob Trader option] Fighter skill: Group can explain the code for how fighter skill affects likelihood of success of robbing trader (the higher the fighter skill, the more likely success)	3.0 pts
[Rob Trader option] Successful robbing: Some of trader’s items added to player inventory, cargo space updated, and player continues to intended destination	3.0 pts
[Rob Trader option] Unsuccessful robbing: Ship health is lowered and player continues to intended destination	3.0 pts
[Negotiate Trader option] Merchant skill: Group can explain the code for how merchant skill affects likelihood of success of negotiating with trader (the higher the merchant skill, the more likely success)	3.0 pts
[Negotiate Trader option] Successful negotiation: the trader's price is significantly reduced	3.0 pts
[Negotiate Trader option] Unsuccessful negotiation: trader's price should increase	3.0 pts
[Negotiate Trader option] Can only negotiate once per encounter (after negotiating once, the players only options should be to buy from, ignore, or rob the trader)	3.0 pts

[Checkstyle] Code Style Guide (see Files > TA Lecture Slides > Code Review > Java/Python Guide.pdf)	10.0 pts