

InterGalactic

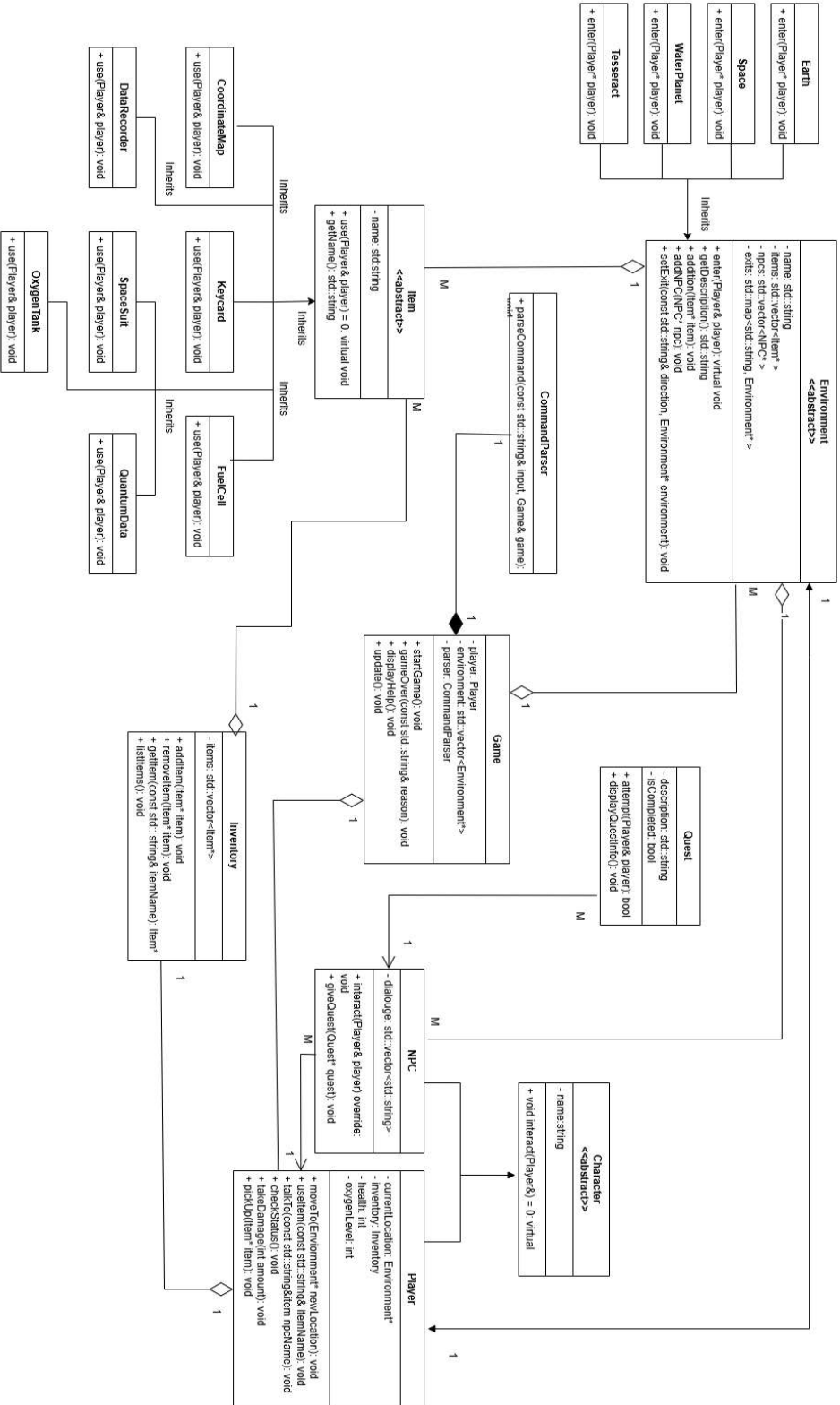


Major Project Group 14

[Suwinya Rajapakse, Japnoor Manku, Nicolas Crespo Cedeno, Moksh Shah]

[November 8, 2024]

UML Diagram



Story Overview:

In this text-based sci-fi adventure game called InterGalactic, which is inspired by the movie *Interstellar*, you play as Cooper, a former NASA pilot turned farmer. Earth is on the brink of collapse due to environmental catastrophes. Strange gravitational anomalies occur on your farm, leading you on a journey to find a new home for humanity. Your adventure will take you from Earth to distant galaxies, navigating through wormholes and black holes, solving puzzles, and making critical decisions that determine the fate of humankind.

Gameplay Summary:

- **Main Objective:** Find a habitable planet to ensure the survival of humanity.
- **Gameplay Mechanics:**
 - **Movement:** Navigate through various interconnected environments.
 - **Interaction with Environments:** Examine surroundings, unlock doors, and solve environmental puzzles.
 - **Interaction with Objects:** Pick up, use, and combine items to progress.
 - **Interaction with NPCs:** Engage in dialogues that influence the story outcome.
 - **Usable Items:** Collect and utilize items essential for survival and mission success.
 - **In-Game Help:** Use the `help` command to get a list of available actions and hints.

Class Descriptions

Class Name	Method Name	Description
Game	<ul style="list-style-type: none">+ startGame(): void+ gameOver(const std::string& reason): void+ displayHelp(): void+ parseCommand(const std::string& input): void	<ul style="list-style-type: none">- Initializes the game and presents the opening scenario.- Ends the game with a given reason.- Shows available commands and hints.- Processes player input commands.
Character <<abstract>>	<ul style="list-style-type: none">+ void interact(Player&) = 0: virtual	<ul style="list-style-type: none">- Defines interaction for the derived classes
Player (inherits from Character)	<ul style="list-style-type: none">+ moveTo(Enviornment* newLocation): void+ useItem(const std::string& itemName): void	<ul style="list-style-type: none">- Moves the player to a new environment.- Uses an item from the inventory.

	+ talkTo(const std::string&item npcName): void + checkStatus(): void + takeDamage(int amount): void + pickUp(Item* item): void	- Initiates dialogue with an NPC. - Displays the player's current status. - Takes the damage by the amount of damage to the equipment - Picks up item
Environment Class <<abstract>>	+virtual void enter(Player& player)	Actions upon entering the environment
	+std::string getDescription()	Provides a description of the environment
	+void addItem(Item* item)	Adds an item to the environment
	+void addNPC(NPC* npc)	Adds an NPC to the environment.
	+void setExit(const std::string& direction, Environment* environment)	Sets movement options
Item Class<<abstract>>	+virtual void use(Player& player)	Defines how an item affects the player.
	+std::string getName()	Returns the item's name.
NPC Class (Inherits from Character)	+void interact(Player& player)	Defines NPC interaction.
	+void giveQuest(Quest* quest)	Assigns a quest to the player.
Quest Class	+bool attempt(Player& player)	Attempts to complete the quest.
	+void displayQuestInfo()	Shows quest details.
CommandParser Class	+void parseCommand(const std::string& input, Game& game)	Parses and executes command.
Inventory Class	+void addItem(Item* item)	Adds an item to the inventory.

	Void removeItem(Item* item)	Removes an item from the inventory.
	+void listItems()	Displays all the items in the inventory.

Tasks (Puzzles):

1. Task 1: Decode the Sand Lines

- **Objective:** Interpret the mysterious sand patterns in Murphy's room to find NASA's coordinates.
- **Mechanic:** Choose between interpreting the patterns as Morse code or binary coordinates.
- **Completion:** Selecting "binary coordinates" reveals the location of the secret NASA base.
- **Failure:** Misinterpreting the patterns leads to missing the opportunity to find NASA, resulting in game over.

2. Task 2: Convince NASA

- **Objective:** Persuade Dr. Brand to allow you to pilot the Endurance mission.
- **Mechanic:** Engage in a dialogue with multiple-choice responses that showcase your skills and commitment.
- **Completion:** Choosing responses that highlight your experience secures your place on the mission.
- **Failure:** Poor dialogue choices result in being denied the mission, leading to game over.

3. Task 3: Find Fuel for the Spaceship

- **Objective:** Locate and collect sufficient fuel cells to power the Endurance spacecraft.
- **Mechanic:** Explore the NASA base, search different areas, and collect hidden fuel cells.
- **Completion:** Collecting the required number of fuel cells ensures a successful launch.
- **Failure:** Insufficient fuel cells cause a malfunction during launch, resulting in game over.

4. Task 4: Navigate the Wormhole

- **Objective:** Successfully pilot the Endurance through a wormhole near Saturn.
- **Mechanic:** Input the correct sequence of navigational commands based on provided data.
- **Completion:** Correct maneuvers lead to safe passage to the new galaxy.
- **Failure:** Incorrect commands cause the ship to be lost in the wormhole, leading to game over.

5. Task 5: Collect Planetary Data Before Tidal Wave

- **Objective:** On Miller's planet, gather crucial planetary data and return before a tidal wave strikes.
 - **Mechanic:** Navigate the planet's surface, use the Data Recorder, and monitor the timer representing the approaching wave.
 - **Completion:** Returning to the ship in time with the data advances the mission.
 - **Failure:** Delay results in the tidal wave destroying the lander, leading to game over.
6. **Task 6: Send Data from the Tesseract**
- **Objective:** Transmit critical quantum data back to Earth from within the Tesseract.
 - **Mechanic:** Arrange data sequences correctly to send a Morse code message to Murphy.
 - **Completion:** Successfully sending the message allows humanity to solve gravity and evacuate Earth.
 - **Failure:** Incorrect data arrangement leads to humanity's doom, resulting in game over.

Death Factors (Ways to Lose):

1. **Misinterpreting Sand Lines:** Failure in Task 1 leads to inability to find NASA.
2. **Denied by NASA:** Poor dialogue choices in Task 2 result in mission denial.
3. **Launch Failure:** Not collecting enough fuel cells in Task 3 causes a critical malfunction.
4. **Lost in Wormhole:** Incorrect navigation in Task 4 leads to being lost in space.
5. **Perishing on Miller's Planet:** Failing to return before the tidal wave in Task 5 results in death.
6. **Humanity's Extinction:** Failing Task 6 prevents saving humanity.