Group 1
Use Case Specifications

ID:	startGame()
Title:	Initialize game and setup view
Description:	Initializes the gamestate class variable and prepares the view to display the data, this involves laying out the planets, deck initialization, and players being given their cats.
Primary Actor:	GameController
Preconditions:	There is a player to start the game.
Postconditions:	The game is started and ready to be interacted with by the first player.
Main Success Scenario:	1.GameController creates a new GameState variable, which initializes all of its data in its constructor. 2. GameController's action count gets set to 3 actions, for the first player to take.
Extensions:	None.
Frequency of Use:	At the start of each game.
Status:	Implemented.
Owner:	Development Team
Priority:	High - Essential

ID:	actionTaken()
Title:	An action has been taken.
Description:	Performs variable changes whenever a valid action has been taken, such as lowering the player's action count, and performing end of turn functions when their action count reaches 0.
Primary Actor:	Player
Preconditions:	A valid action has been taken
Postconditions:	The action counter has been decreased, and potentially the player's turn is over, with fascist tokens being updated, and cats being scratched according to a dice roll.
Main Success Scenario:	1.The action counter is decreased. 2. End function. 3. Change the player's turn to the next player in line. 4. Calculate the amount of dice to be rolled based on the current fascism scale and tokens removed. 5. For each dice, roll a dice, and add a fascist token to each planet according to their ID. 6.
Extensions:	2a. If remaining actions are 0, proceed to 3. 5a. If a cat is on the planet that was rolled, add a scratch to it and proceed.
Frequency of Use:	Whenever a valid action has been taken.
Status:	Implemented.
Owner:	Development Team
Priority:	High - Essential

ID:	playCard
Title:	Play a card.
Description:	Plays an inputted card, performing said card’s unique CardAction.
Primary Actor:	Player
Preconditions:	A player has a card, and chooses to play it.
Postconditions:	The effects of the card played have been applied.
Main Success Scenario:	1. Player selects a card. 2. The card has its effect played, unique to said card. 3. The card is removed from the player’s hand.
Extensions:	2a. The card’s effect is invalid, and the card is refunded.
Frequency of Use:	Whenever a card is played.
Status:	Implemented.
Owner:	Development Team
Priority:	High - Essential

ID:	travel
Title:	Travels to a planet.
Description:	Takes an inputted direction, and travels to the planet in that direction if it exists.
Primary Actor:	Player
Preconditions:	It is a player’s turn, and they have remaining actions, and choose to travel.
Postconditions:	The player moves in said direction.
Main Success Scenario:	1. The player inputs a direction. 2. The player’s cat moves in that direction. 3. actionTaken() is called.
Extensions:	2a. The direction is invalid, move is invalid.
Frequency of Use:	Whenever a travel button is selected.
Status:	Implemented.
Owner:	Development Team
Priority:	High - Essential

ID:	rollDice
Title:	Rolls dice
Description:	Takes a random number from a random number generator then returns it.
Primary Actor:	GameController
Preconditions:	The method is called.
Postconditions:	The method returns an int.
Main Success Scenario:	1. The method is called. 2. The method rolls a number, then returns it.
Extensions:	None

Frequency of Use:	Whenever a turn is ended and dice need to be rolled.
Status:	Implemented.
Owner:	Development Team
Priority:	High - Essential

ID:	fight
Title:	Fight Fascism
Description:	Fights fascism at the player’s current planet, removing a fascism token at it if there is one.
Primary Actor:	Player
Preconditions:	The player has an action left, and chooses the fight action.
Postconditions:	The planet the player is on has one less fascism token.
Main Success Scenario:	1. The player chooses to fight fascism. 2. The planet loses a fascism token. 3. The amount of fascism tokens removed in a turn gets incremented. 4. Call actionTaken(). 5. End function.
Extensions:	2a. There are no fascism tokens on the planet, end function.
Frequency of Use:	Whenever the fight action is chosen on a player's turn.
Status:	Implemented
Owner:	Development Team
Priority:	High - Essential

ID:	restock

Title:	Restock Cards
Description:	Restocks the players cards until their hand is full.
Primary Actor:	Player
Preconditions:	The player’s hand is less than full, and they choose the restock action.
Postconditions:	The player’s hand is full of cards.
Main Success Scenario:	1. Player calls the function. 2. Function adds cards from the deck to the player's hand until the player’s hand is full. 3. Call actionTaken() 4. End function.
Extensions:	2a. The player’s hand is already full, end function.
Frequency of Use:	Whenever the restock command is chosen.
Status:	Implemented
Owner:	Development Team
Priority:	High - Essential