Milestone 3: Use-Cases COMP 3721
DUE Friday October 30th at 6PM (via GitHub)
Marcia Gallant & Loryn Losier

ID:	**UC1**
Title:	Play Game
Description:	Move your Robot Turtle to a Robot Jewel in order to win the game.
Primary Actor:	GameBoardModel
Preconditions:	The GameBoardController has run the games setup and the GameBoardModel has calculated that it is the first player's turn to make a move.
Postconditions:	The GameBoardModel moves the player's Robot Turtle to the correct Robot Jewel tile depending on what card the user picked. It is the next player's turn.
Main Success Scenario:	1. It is the Player's turn. 2. The player picks a card. 3. GameBoardModel moves the player's Robot Turtle in the correct position depending on what card the GameBoardModel picked. 4. It is the next player's turn 5. Steps 2-4 are repeated until the player acquires a Jewel. 6. The Player acquires the Jewel and wins.
Extensions:	A. The player picks a card that is trying to move the Robot Turtle in a tile where an obstacle is present. 1. In the case the obstacle is a Stone Wall: a. The Robot Turtle fails to move. b. GameBoardModel generates a funny noise. c. The player plays the Bug card. d. The player picks new cards. 2. In the case the obstacle is a Crate (and there is not another obstacle in the tile behind the crate in the direction the Robot Turtle is instructed to move): a. The GameBoardModel moves the Robot Turtle onto the tile in which the crate was located. b. The crate is then pushed forward onto the tile that was originally behind the crate.

	3. In the case the obstacle is a Crate (and there is another obstacle in the tile behind the crate in the direction the Robot Turtle is instructed to move): a. The Robot Turtle fails to move. b. The GameBoardModel generates a funny noise. c. The player plays the Bug card. d. The player picks new cards. 4. In the case the obstacle is another Robot Turtle: a. The Robot Turtle fails to move. b. The GameBoardModel generates a funny noise. c. The player plays the Bug card. d. The player picks new cards. 5. In the case the obstacle is the edge of the gameboard: a. The Robot Turtle fails to move. b. The GameBoardModel generates a funny noise. c. The player plays the Bug card. d. The player picks new cards.. D. player picks a card that rotates their Robot Turtle: 1. In the case that the Robot Turtle was rotated incorrectly: a. The player plays the Bug Card. b. The player picks new cards.
Frequency of Use:	Could be nearly continuous.
Status:	Coding completed
Owner:	Marcia and Loryn
Priority:	High

ID:	**UC2**
Title:	Move Turtle
Description:	The GameBoardModel moves the Robot Turtle to the appropriate tile after checking which card or cards the player has picked.
Primary Actor:	GameBoardModel
Preconditions:	A player has picked a card.
Postconditions:	It is the next player's turn.
Main Success Scenario:	1. It is a player's turn 2. The player picks a card. 3. The GameBoardModel moves the correct Robot Turtle depending on which card the player picked into the correct position. 4. It's the next player's turn
Extensions:	A. The GameBoardModel could not move the Robot Turtle: 1. In the case that the Robot Turtle could not be moved due to an immovable obstacle: a. The GameBoardModel generates a funny noise. b. The player plays the Bug card. c. The player picks new cards. B. The GameBoardModel moves Robot Turtle 1. In the case that the GameBoardModel has to move the Robot Turtle and a Crate: a. The GameBoardModel moves the Robot Turtle onto the tile in which the crate was located. b. The crate is then pushed forward onto the tile that was originally behind the crate.
Frequency of Use:	Could be nearly continuous.
Status:	Coding completed
Owner:	Marcia and Loryn
Priority:	High

ID:	**UC3**
Title:	Start Game
Description:	Setting up the game
Primary Actor:	GameBoardController
Preconditions:	The game has not been set up yet
Postconditions:	The gameboard model is set up and it is displayed to the users. The users can now begin playing the game
Main Success Scenario:	1. The user inputs how many players there are in the game. 2. The default GameBoard consisting of all blank tiles, 4 Robot Turtles and 4 Robot Jewel tiles is set up. 3. The GameBoardController adds the correct number of objects onto the GameBoard and the GameBoardModel changes the tile type accordingly. 4. The GameBoardModel creates a set of cards for each player. 5. The gameboard is displayed to the screen by displayGameboard after GameBoardController gets the model from GameBoardModel and passes it into displayGameboard.

Extensions:	A. The user tries to add too many obstacles of a certain kind 1. The user tries to add more than 20 Stone Walls. a. The GameBoardModel will not allow for too many walls to be placed. 3. The user tries to add more than 8 Crates. a. The GameBoardModel will not allow for too many crates to be placed. B. The user tries to start the game without adding enough obstacles. 1. The user tries to start the game with less than 20 Stone Walls on the GameBoard. a. The GameBoardModel will not allow the game to begin until all walls are placed. 3. The user tries to add more than 8 Crates. a. The GameBoardModel will not allow the game to begin until all crates are placed.

Frequency of Use:	Once per game
Status:	Coding completed
Owner:	Marcia and Loryn
Priority:	High

ID:	**UC4**
Title:	Complete Game
Description:	All players have won the game and the GameBoard will be cleared.
Primary Actor:	GameBoardModel
Preconditions:	All Robot Turtles are on a Tile where the tile type is Jewel.
Postconditions:	The Gameboardmodel is cleared.
Main Success Scenario:	1.All turtles are on a jewel tile 2.The GameBoardModel clears the game. 3.The view of the game returns to the start where the users must input their names.
Extensions:	None, the game clears itself automatically once all Robot Turtles have won.

Frequency of Use:	Once per game
Status:	Coding completed
Owner:	Marcia and Loryn
Priority:	High