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 mode: either set up program or set up function frog. 3. The player picks a card. 4. Steps 2-3 are repeated, although step 2 can be skipped. 5. The player presses the write program button. 6. GameBoardModel moves the player's Robot Turtle in the correct position depending on what card the GameBoardModel picked and changes the state of the game board if other tile objects were changed. 7. If the player succeeded in obtaining a Robot Jewel, the player wins and is done the game. If the player failed, then the player uses the bug card to revert their Robot Turtle back to their original position. 8. It is the next player's turn 9. Steps 2-7 are repeated until all players acquire a Jewel. 10. All players acquire a jewel.
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..

6) In the case the obstacle is an Ice Wall (that is not melted):
   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.

.

7) In the case the obstacle is an Ice Wall(that is melted; puddle)
   a) The GameBoardModel moves the Robot Turtle onto the tile in which the puddle is located.

8) In the case the obstacle is a portal:
   a) The GameBoardModel moves the Robot Turtle onto the tile in which the portalis located.
   b) The GameBoardModel moves the Robot Turtle onto the corresponding portal tile to the portal tile the Robot Turtle is standing on.

9)

B. 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. 2)In the case that the Robot Turtle rotated while it was on top of a portal tile: a) The GameBoardModel rotates the Robot Turtle. b) The GameBoardModel moves the Robot Turtle onto the corresponding portal tile to the portal tile the Robot Turtle is standing on. C. Player picks the laser card a) The GameBoardMoodel checks if there are any Ice Walls in front of the turtle with no objects between them b) If an Ice Wall was detected, change its tileType to Puddle and make it able to be moved on by Robot Turtles.
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: 1. The GameBoardModel generates a funny noise. 2. The player plays the Bug card. B. The GameBoardModel moves Robot Turtle 1. In the case that the GameBoardModel has to move the Robot Turtle and a Crate: 1. The GameBoardModel moves the Robot Turtle onto the tile in which the crate was located. 2. The crate is then pushed forward onto the tile that was originally behind the crate. 2. In the case where the GameBoardModel has to move the turtle either forward or rotate and the Robot Turtle ends up on a portal tile. a. The GameBosrdModel moves the Robot Turtle correctly. b. Then correctly moves the Robot Turtle onto the corresponding portal tile. 3. In the case where the GameBoardModel has to try to melt an Ice Wall. a.The GameBoardModel checks if there are any Ice Walls in front of the Robot Turtle that do not have a tile object in between the Robot Turtle and Ice Wall b.If there was such an Ice Wall, melt it by changing its tileType to Puddle and make it so Robot Turtles can move onto this tile.

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, Robot Turtles and 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. 5. The gameboard is displayed to the screen by displayGameboard after GameBoardController gets the converted version of the model from GameBoardModel and passes it into displayGameboard. 6. The game begins.

Extensions:

A. The user tries to add too many obstacles of a certain kind

1. The user tries to add more than 2 Stone Walls.
   a. The GameBoardModel will not allow for too many walls to be placed.
2. The user tries to add more than 2 Crates.
   a. The GameBoardModel will not allow for too many crates to be placed.
3. The user tries to add more than 2 Portals.
   a. The GameBoardModel will not allow for too many portals to be placed.
4. The user tries to add more than 2 Ice Walls.
   a. The GameBoardModel will not allow for too many Ice Walls 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 2 Stone Walls on the GameBoard.
   a. The GameBoardModel will not allow the game to begin until all walls are placed.
2. The user tries to start the game with less than 2 crates.
   a. The GameBoardModel will not allow the game to begin until all crates are placed.
3. The user tries to start the game with less than 2 portals.
   a. The GameBoardModel will not allow the game to begin until all portals are placed.

24 The user tries to start the game with less than 2 Ice Walls.

   a. The GameBoardModel will not allow the game to begin until all Ice Walls 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:	GameBoardController
Preconditions:	All Robot Turtles are on a Tile where the tile type is Jewel.
Postconditions:	The program closes and a message saying that everyone has won is displayed to the screen.
Main Success Scenario:	1. All turtles are on a jewel tile 2. The GameBoardController closes the game through the GameBoardDisplay.. 3. A pop-up message is displayed to the screen informing the users that everybody has won.

Extensions:	None, once all Robot Turtles are on a Robot Jewel, the actions the program performs to close the software remain constant. There are no deviations.
Frequency of Use:	Once per game
Status:	Coding completed
Owner:	Marcia and Loryn
Priority:	High