Name: Jason Murphy
Student number: C00172382
Course: Games development year 2
Title: Battleships mini project
Date started: 08/01/2014
Inception deadline date: 27/01/2014

First iteration deadline: 24/03/2014
first iteration:

Vision:

To begin with I will describe the game of battleships itself. It is a well known game as it has been around for a very long time. The first commercial versions (plastic board) appeared in the 1930s although other versions appear to have existed since before the first world war.

source of photo

It is a guessing game for two players. The players place their battleships on their side of the grid. The board is made up of squares to make a grid, which is typically square. The ships are different sizes and so will cover different amounts of squares on the board. The players then take turns trying to guess where the other players battleship is.

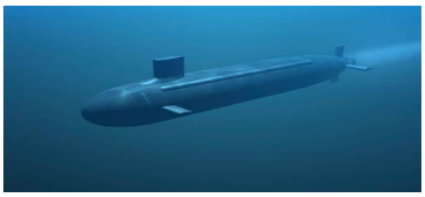
They can guess one square per go and get another go if they manage to hit the other players ship. When a ship is hit a marker is placed on the section of the ship that is hit. When the ship is completely covered it is destroyed. The game continues until one player has no more ships left.

What I want to do is to make a digital version of this game that can be played by one player against an AI opponent. The player should be able to pick up on the rules within the first few minutes of gameplay(assuming they have never played battleships before). If the player has played before they should be able to jump straight in and play. The game will be highly customisable with the players being able to choose how many ships both him/her and the Ai opponent will have, the shape/theme of those ships and also the size of the board/grid(e.g. 10*10, 5*8 etc).

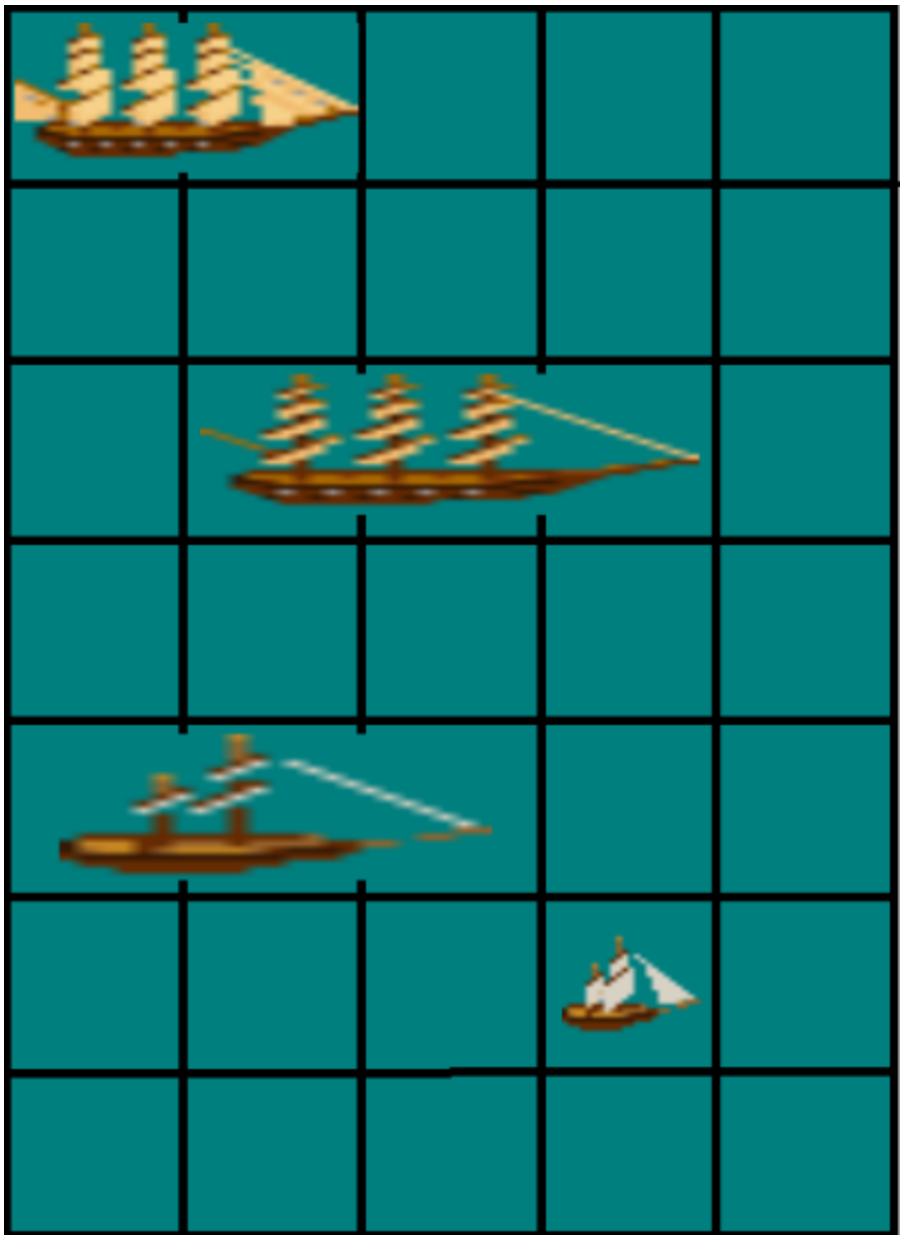
Play

Options

Quit



possible menu screen ↑

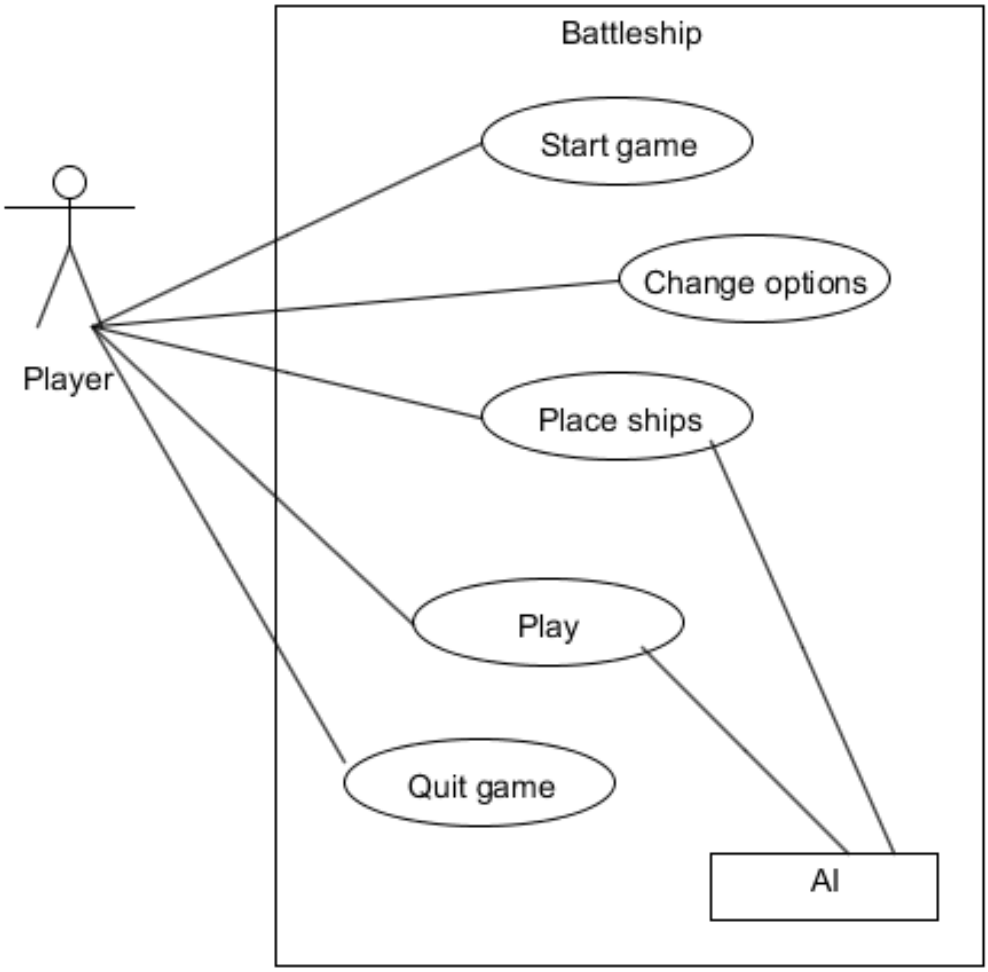


possible board with pirate themed ships ↑

My target audience would be casual gamers who just want a simple game to play in order to relax. It will not be mentally draining. It will be a game where people play a few games each day to pass time. There will not be an age limit on it per say but it would probably be best to supervise young children while they are playing at a computer.

My target platform is a downloadable executable for computers running windows, written in c#/xna.

Use case diagram:



Brief Use Cases:

(i) Use Case: Start game

Actors: Player

Description: This use case begins when the player decides to play a game of battleships. They start the application and the game loads up to the main menu where they are presented with the choice play and quit.

(ii) Use Case: Change options

Actors: Player

Description: This use case begins after the player has selected play game from the main menu(this is shown before the player gets to play the game). The player is presented with the options that they can change. They can choose how many ships the player and AI opponent can have(e.g. 4,5,6,7), the theme/appearance of ships(e.g. WW", modern, pirate) and the size of the board(e.g. 5*5, 5*8(in terms of squares)). The use case ends once the player has selected the options they want and press play.

(iii) Use Case: Place ships

Actors: Player, AI

Description: This use case begins when the player has finished selecting the options they want for the game. The player is prompted to place their ships. They are shown which ship (size and type) they are currently placing. The player can rotate the ship 90 degrees clockwise or anti clockwise. Each ship will take up a certain amount of squares depending on their size. This continues with the player placing all their ships. The AI also places its ships at this time as well. When the player has their ships ready they press ready and this use case ends.

(iv) Use Case: Play

Actors: Player, AI

Description: This use case begins when the both the player and the AI have finished placing their ships. They then take it in turns to pick a square on the board. If the square chosen contains a part of the other players ship then a hit is scored and the player gets to pick another square. When a ship is completely destroyed it is removed from the board. This continues until either the player or the AI have no more ships left. At this point the game ends and a winner is declared.

(v) Use Case: Quit game

Actors: Player

Description: This use case begins when the player decides they want to quit battleships. They select quit from the menu and are asked to confirm that they want to quit battleships. The game then closes.

Supplementary Specification:

Usability: Novice user's should be able to finish their first game in less than 10 minutes and get used to the controls in that time. They should be confident after the first game. Experienced users should not find the game too easy either, there must be a degree of challenge involved.

Reliability: The game should not crash more than once every 48 hours at a minimum.

Performance: The game should start and load up in less than 30 seconds and not experience any major lag. Lag during gameplay should be kept to a minimum.

Supportability: The game will be available in 3 languages: English,Irish and French. This will be an option when starting the application for the player to choose. Code should be clearly commented so as to make it easy to refactor or change.

Constraints: The game will be developed in xna which leaves a possible opportunity to port it to windows phone at some point as not much refactoring will be needed. It will be targeted towards windows computers as an executable. It will use a mouse as an input device.

Detailed Use Cases:

(i) Name: Start game

Actors: Player
Description: This use case begins when the player decides to play a game of battleships. They start the application and the game loads up to the main menu where they are presented with the choice play and quit.
Main Success Scenario:
1. Player decides to play a game of battleships
2. Player starts the application
3. The system displays Main menu
Alternatives:
3a. Player decides to close the application

(ii) Name: Change options

Actors: Player
Description: This use case begins after the player has selected play game from the main menu(this is shown before the player gets to play the game). The player is presented with the options that they can change. They can choose how many ships the player and AI opponent can have(e.g. 4,5,6,7), the theme/appearance of ships(e.g. WW1", modern, pirate) and the size of the board(e.g. 5*5, 5*8). The use case ends once the player has selected the options they want and press play.
Main Success Scenario:
1. Player selects the play option from the main menu
2. Options menu is displayed
3. Player selects how many ships each player is going to have
4.The system sets the max number of ships to the amount selected by the player
5. Player selects the theme/style of the ships
6.The system sets each ships theme(e.g. WW1, modern, pirate) to the one selected by the player
7. Player selects the size of the board
8.The system sets the size of the board to the size selected by the player
9. Player starts the game with the chosen options
10.The system then starts a game of battleships
Alternatives:
within steps 3-5, the player may decide to not change any options in which case the game starts with the default settings.

(iii) Name: Place ships

Actors: Player, AI

Description: This use case begins when the player has finished selecting the options they want for the game. The player is prompted to place their ships. They are shown which ship (size and type) they are currently placing. The player can rotate the ship 90 degrees clockwise or anti clockwise. Each ship will take up a certain amount of squares depending on their size. This continues with the player placing all their ships. The AI also places its ships at this time as well. When the player has their ships ready they press ready and this use case ends.

Main Success Scenario:

1. System prompts the player to place their ships
2. The system displays which ship is next to be placed on the board
3. The player then places this ship
4. AI places its ships as well
5. When the player has all their ships placed they press ready

Alternatives:

3a, the Player may rotate the current ship in order to place it horizontally or vertically on the board.

3b, Player tries to place the ship with part of it off the board, system does not allow them to place the ship there.

(iv) Name: Play

Actors: Player, AI

Description: This use case begins when the both the player and the AI have finished placing their ships. They then take it in turns to pick a square on the board. If the square chosen contains a part of the other players ship then a hit is scored and the player gets to pick another square. When a ship is completely destroyed it is removed from the board. This continues until either the player or the AI have no more ships left. At this point the game ends and a winner is declared.

Main Success Scenario:

1. Player picks a square on the AI's board
2. The system checks if the the selected square is occupied by an AI ship
3. AI picks a square on the board
4. The system checks if the selected square is occupied by a Player ship

5. System checks if any ships have been destroyed and removes these destroyed ships from the board
6. The system checks if either the player or the AI have no ships left. The game ends and the player with ships left is declared the winner

Alternatives:

2a, If the square contains a ship then score a hit and allow the player another guess
2b, if there is not a ship then change turn to the AI
4a, if the square chosen by the AI contains a ship, then score a hit and allow the AI another guess
4b, if there is not a ship then change turn to the player

(v) Name: Quit game

Actors: Player

Description: This use case begins when the player decides they want to quit battleships. They select quit from the menu and are asked to confirm that they want to quit battleships. The game then closes.

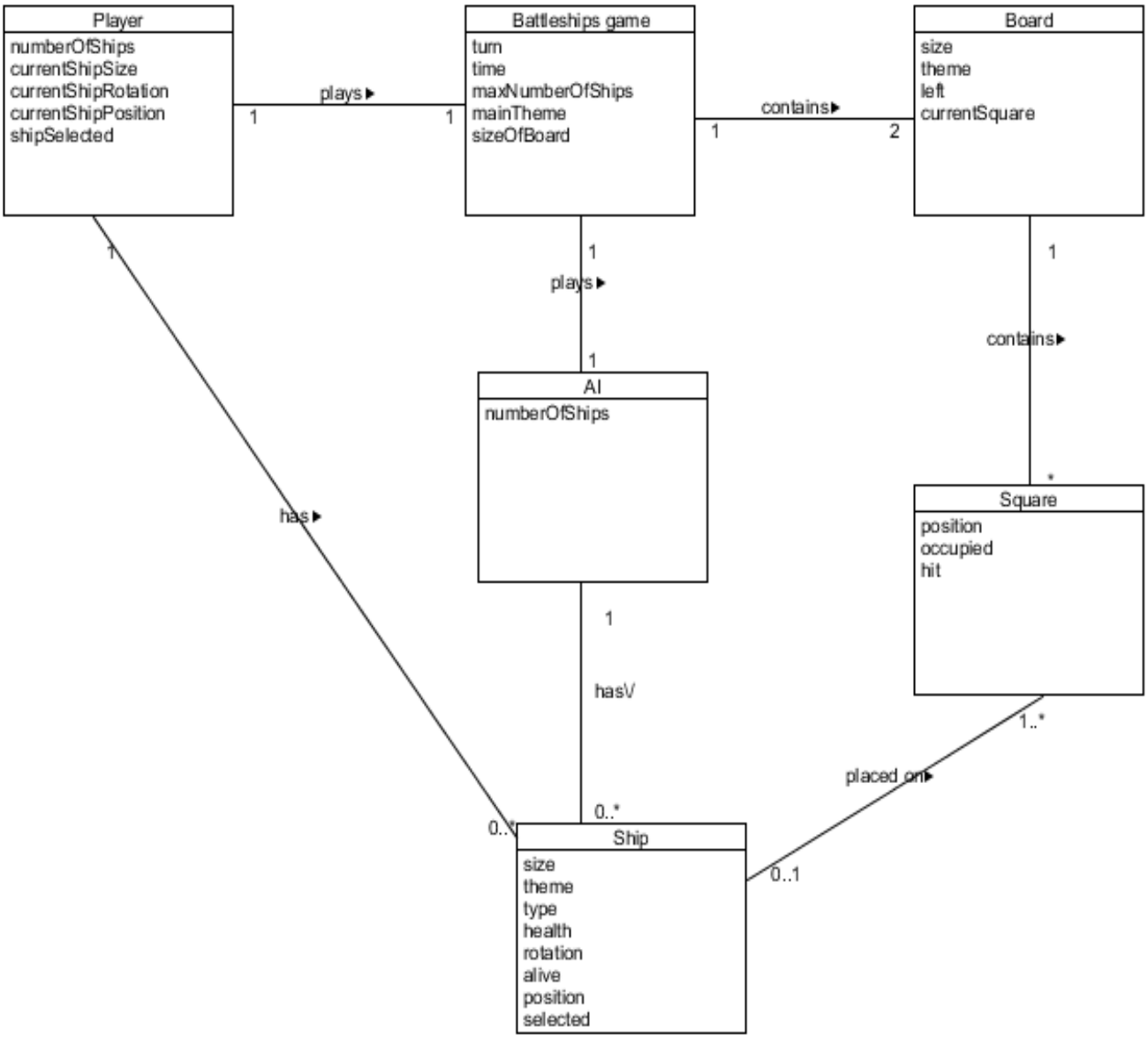
Main Success Scenario:

1. Player decides to quit battleships
2. Player selects quit from the menu
3. The player confirms they want to quit
4. The application closes

Alternatives:

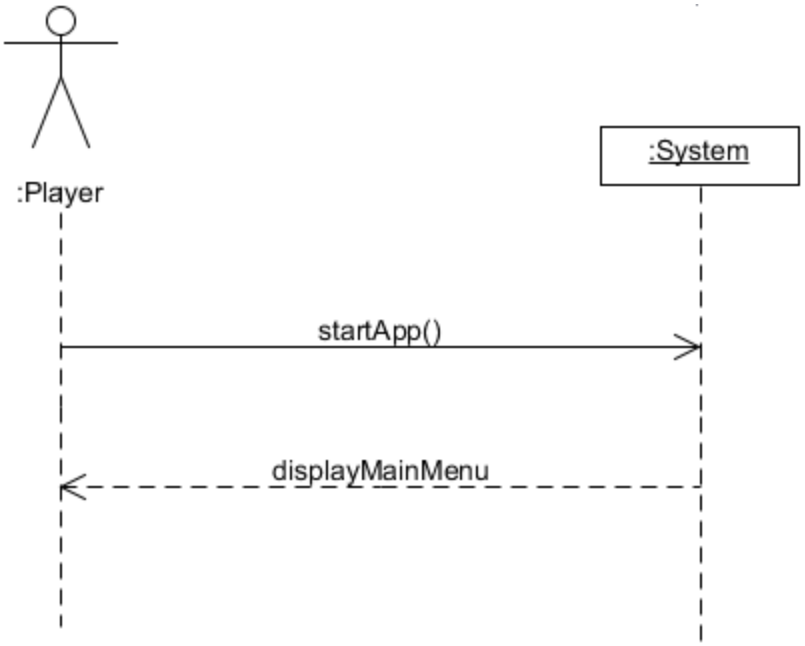
3a, The player changes their mind and does not confirm that they want to quit battleships, so the game continues from where they left off.

Domain Model

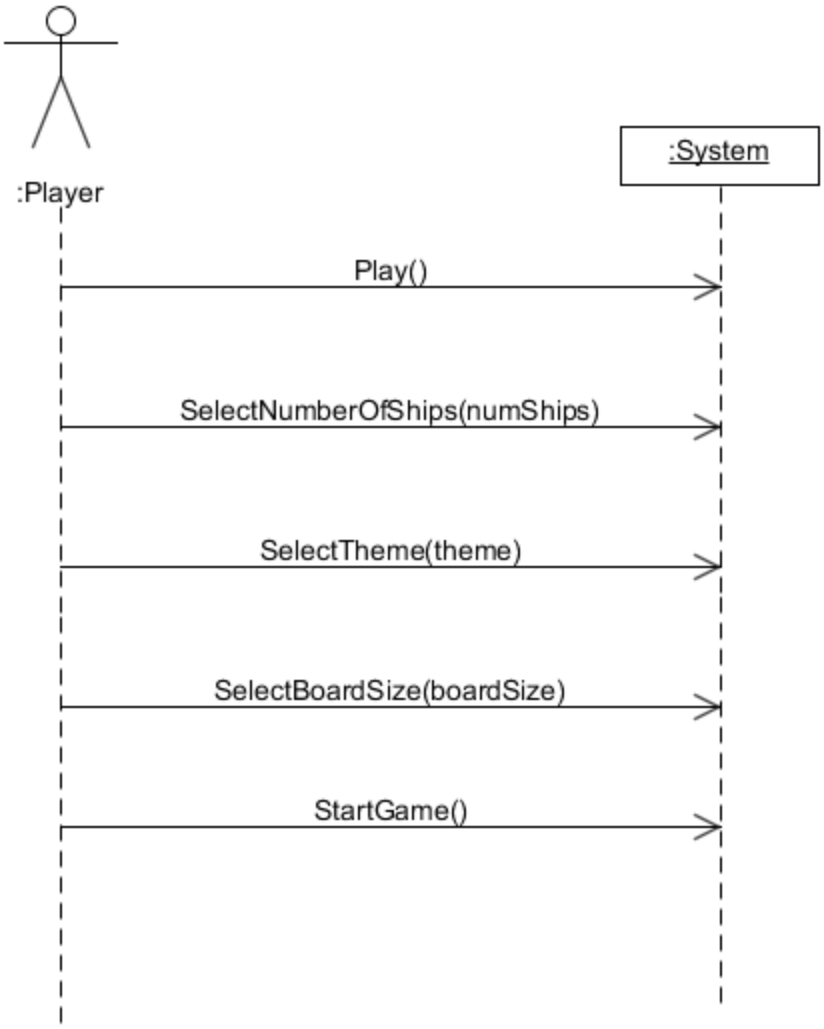


System Sequence Diagrams:

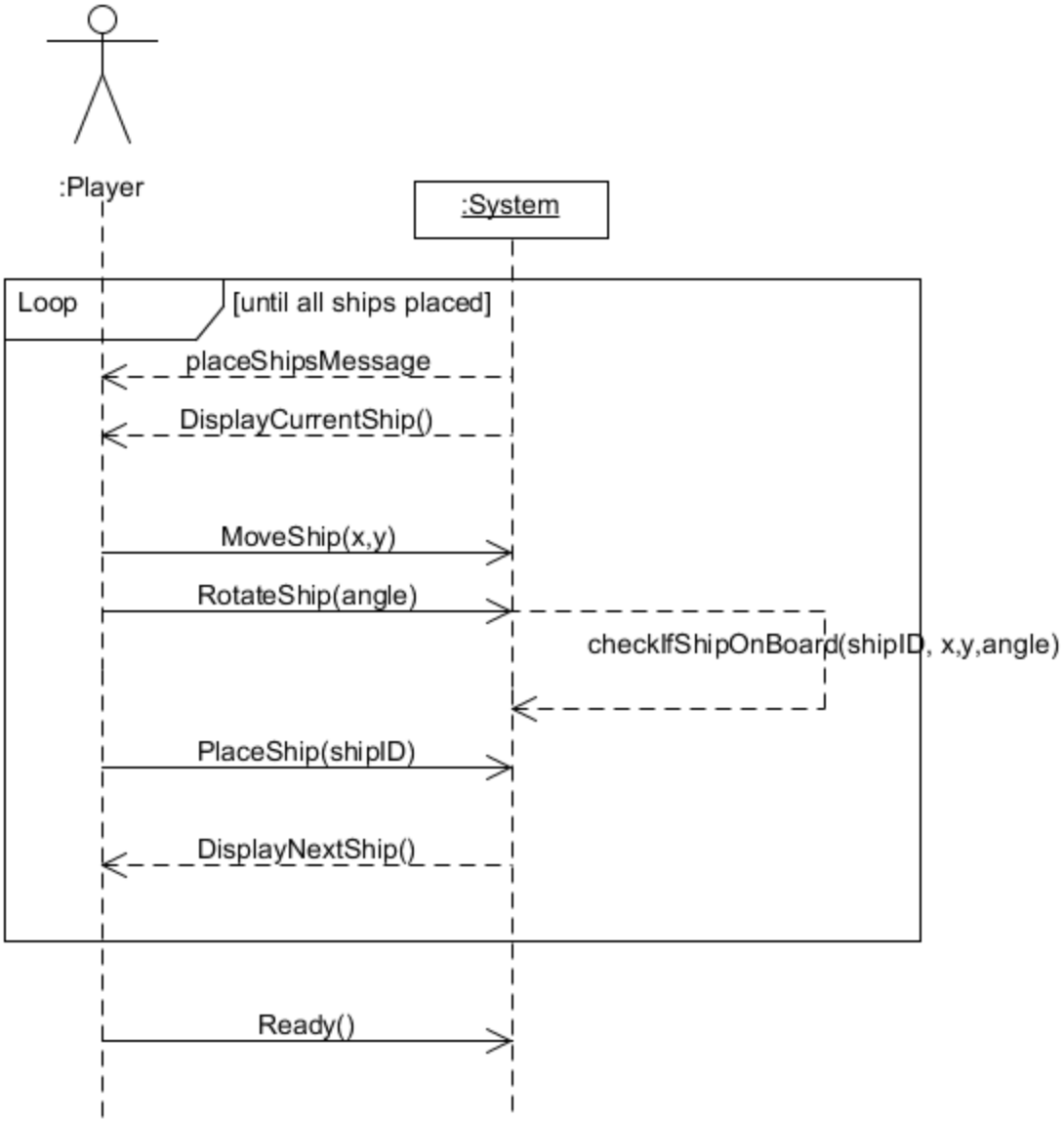
start game:



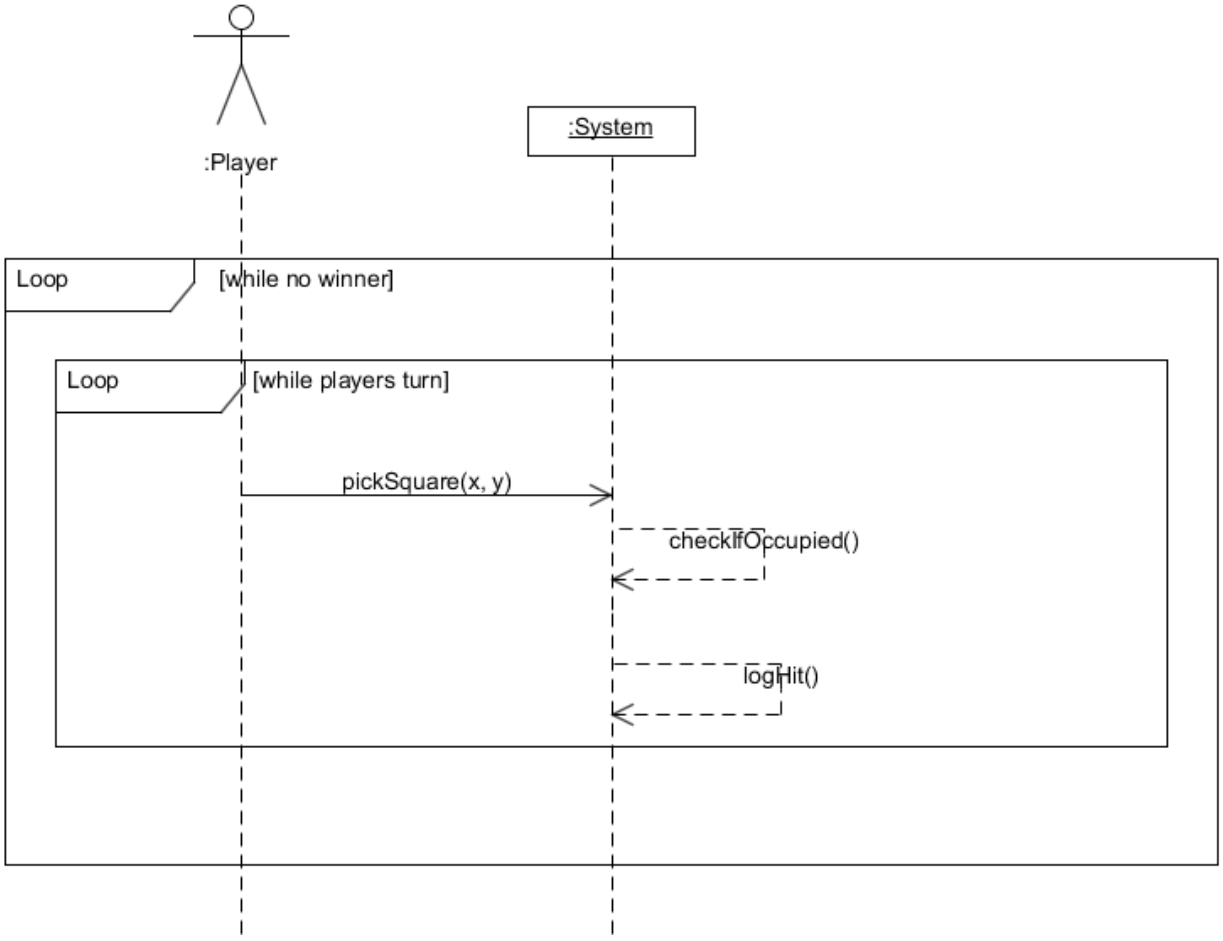
Change options:



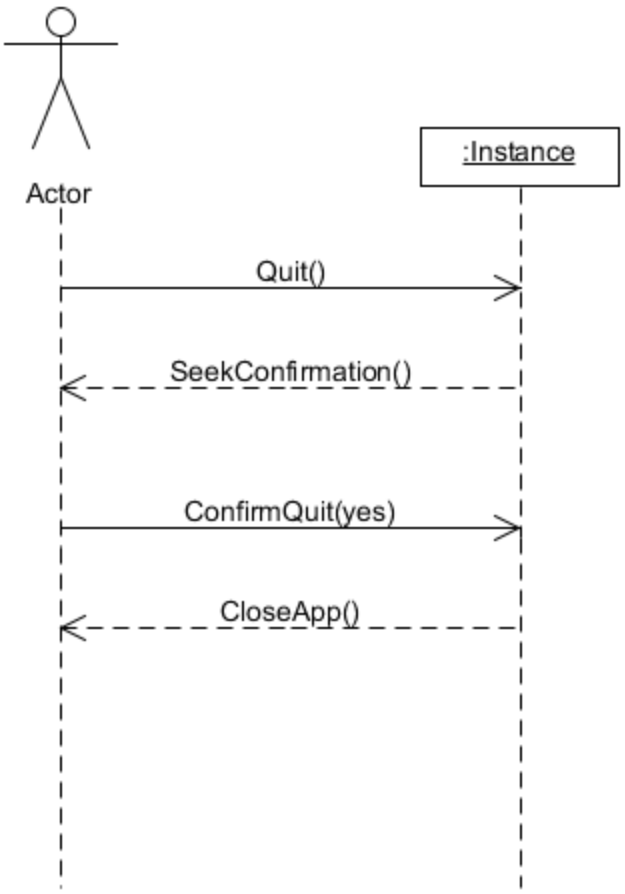
place ships:


:Player
:System
Loop
[until all ships placed]
placeShipsMessage
DisplayCurrentShip()
MoveShip(x,y)
RotateShip(angle)
checkIfShipOnBoard(shipID, x,y,angle)
PlaceShip(shipID)
DisplayNextShip()
Ready()

Play:


:Player
:System
Loop
[while no winner]
Loop
[while players turn]
pickSquare(x, y)
checkIfOccupied()
logHit()

Quit game:


Actor
:Instance
Quit()
SeekConfirmation()
ConfirmQuit(yes)
CloseApp()

Contracts

Contract C01: startApp()
Cross references: use case's: start game
Postconditions:
-An instance of BattleShipGame was created(instance creation)
-An instance of Player was created(instance creation)
-An instance of AI was created(instance creation)

Contract C02: Play()
Cross references: use case's: change options
Postconditions:
-options menu was displayed

Contract C03: SelectNumberOfShips(numShips)
Cross references: use case's: change options
Postconditions:
-maxNumberOfShips was given the value of numShips(attribute modification)

Contract C04: SelectTheme(theme)
Cross references: use case's: change options
Postconditions:
-mainTheme was given the value of theme(attribute modification)

Contract C05: SelectBoardSize(boardSize)
Cross references: use case's: change options
Postconditions:
-sizeOfBoard was given the value of boardSize(attribute modification)

Contract C06: StartGame()
Cross references: use case's: change options
Postconditions:
-Game was started
-many instances of ship were created(instance creation) and associated with Player and AI(association formed)
-board object b1 and b2 were created(instance creation)
-many square objects were created and associated with b1 and b2(instance creation and associations formed)

Contract C07: RotateShip(angle)
Cross References: use case's: Place ships
Postconditions:
-The angle of the ship object is changed(attribute modification)

Contract C08 PlaceShip()
Cross References: use case's: Place ships
Postconditions:
-The ship object was associated with the squares it was placed on(association formed)

Contract C09: Ready()
Cross References: use case's: Place ships
Postconditions:
-The game has now begun

Contract C10: PickSquare(x, y)
Cross References: use case's: Play()
Postconditions:
-The player selected a square to shoot at. This square will be referred to in the next 2 contracts

Contract C11: checkIfOccupied()
Cross References: use case's: Play()
Postconditions:
-If there was a ship associated with the square in question it had some of its health taken away(attribute modification).
-If the ship was left with zero health then its "alive" attribute was set to false(attribute modification)

Contract C12: logHit()
Cross References: use case's: Play()
Postconditions:
-The square selected by the player had it's "hit" attribute set to true(attribute modification)

Contract C13: Quit()
Cross References: use case's: Quit Game
Postconditions:
-The player decided to quit the game

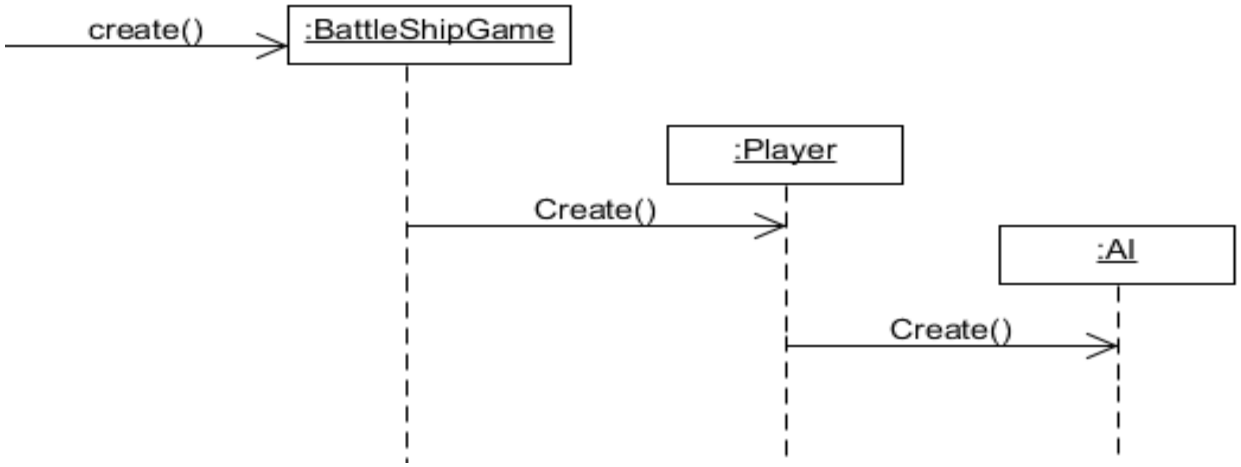
Contract C14: seekConfirmation()
Cross References: use case's: Quit Game
Postconditions:
-The system asked the player to confirm that they wanted to quit

Contract C15: Confirm(yes)
Cross References: use case's: Quit Game
Postconditions:
-The player confirmed that they wanted to quit

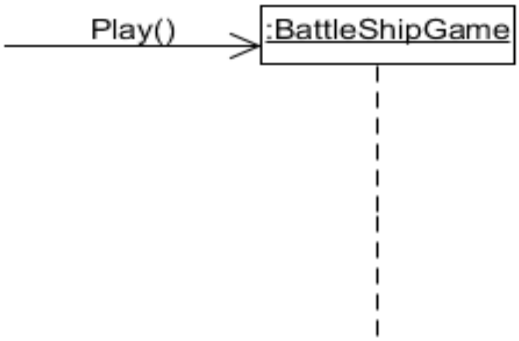
Contract C16: closeApp()
Cross References: use case's: Quit Game
Postconditions:
-The app was closed

Sequence Diagrams:

Sequence for C01

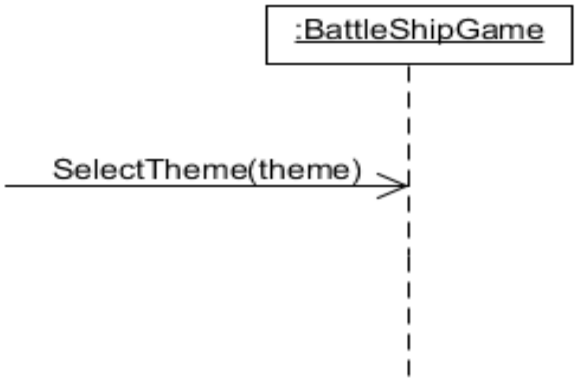


Sequence for C02

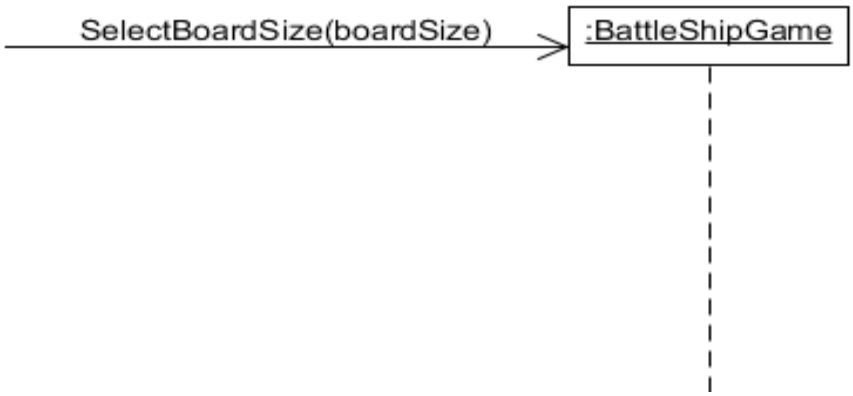


Sequence for C03

Sequence for C04

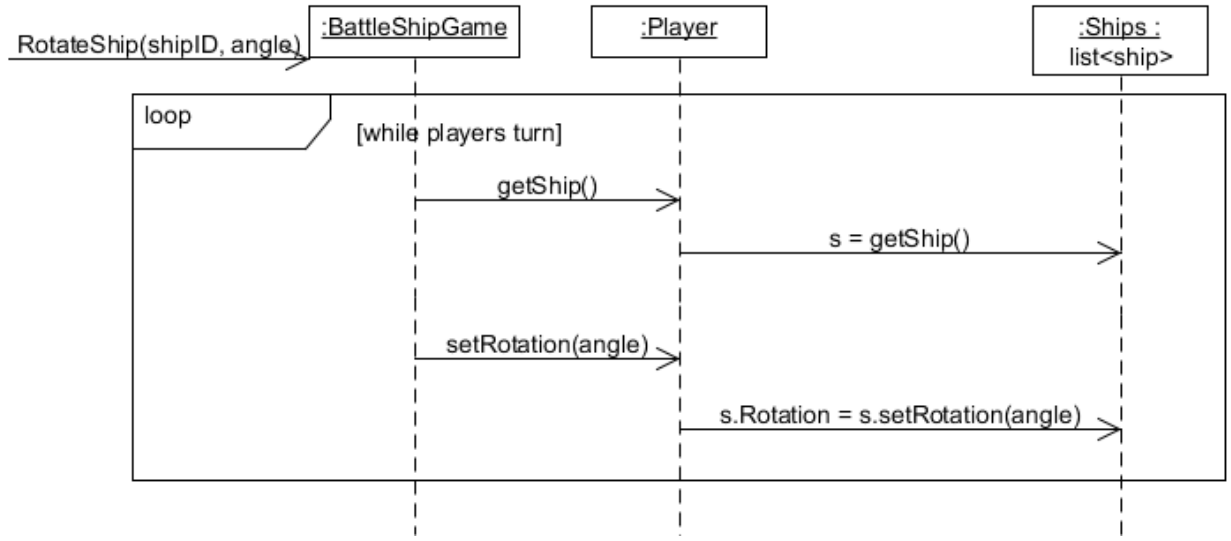


Sequence for C05

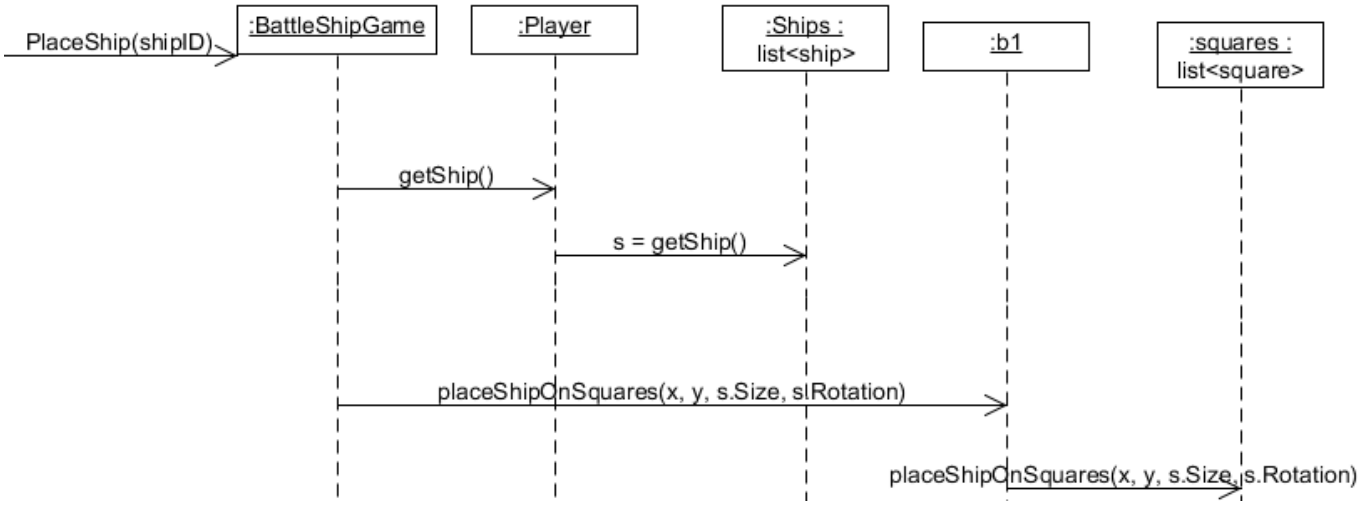


Sequence for C06

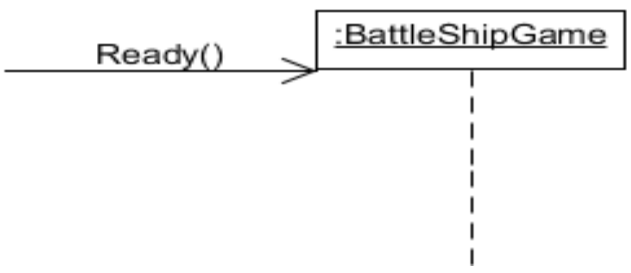
Sequence for C07



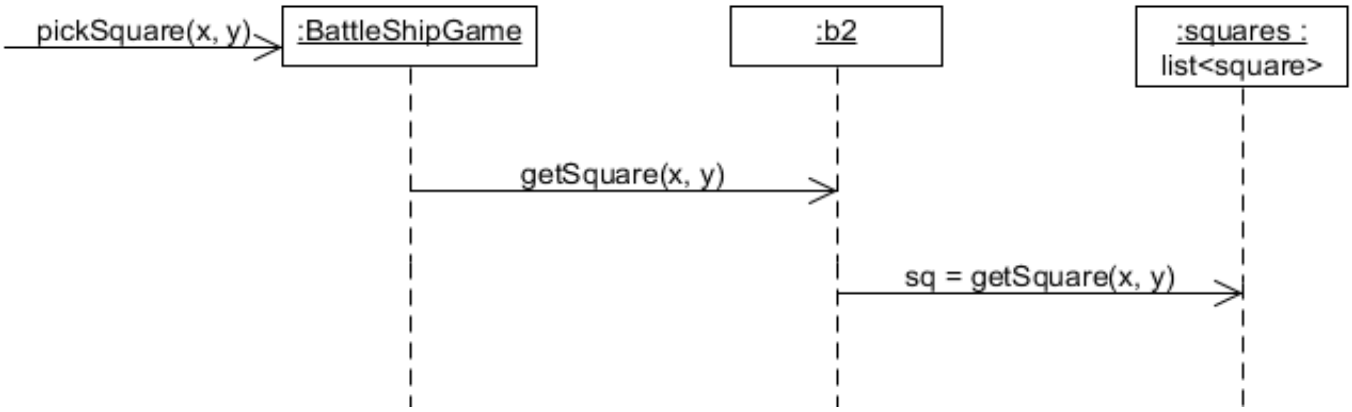
Sequence for C08



Sequence for C09

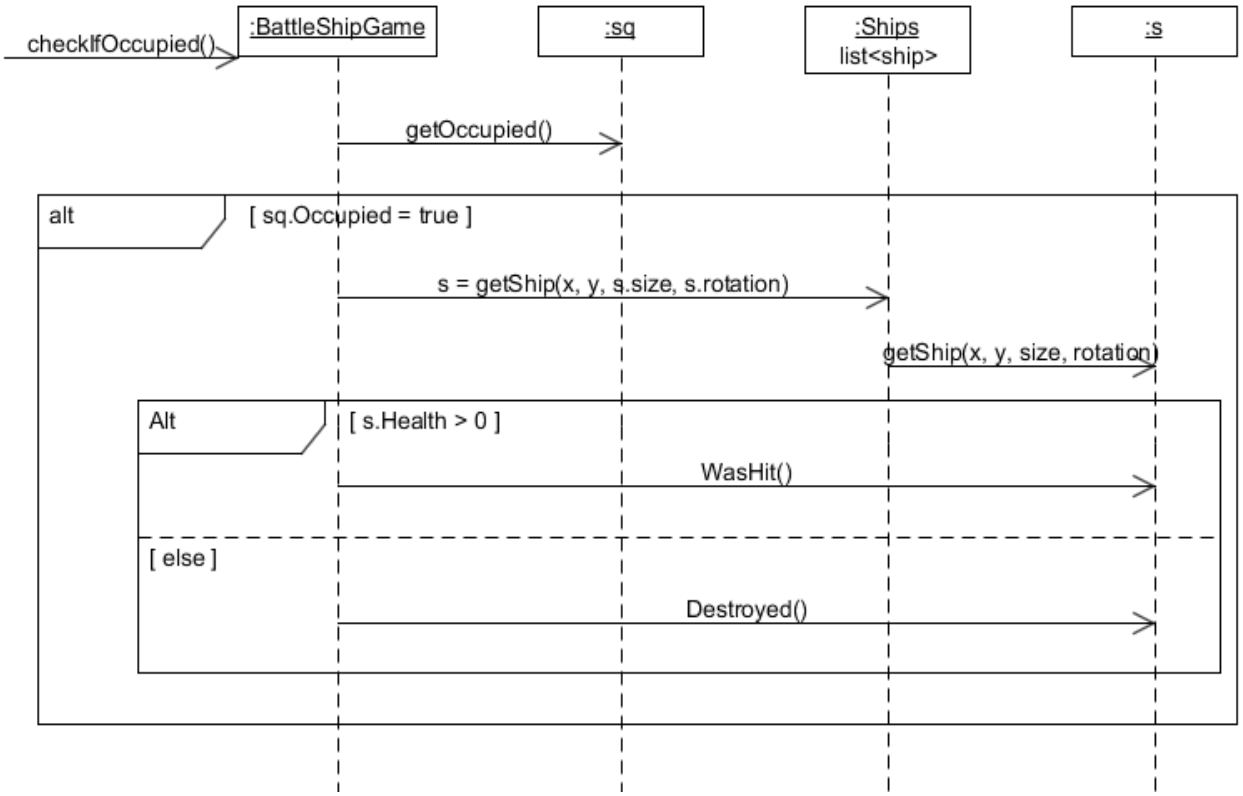


Sequence for C10



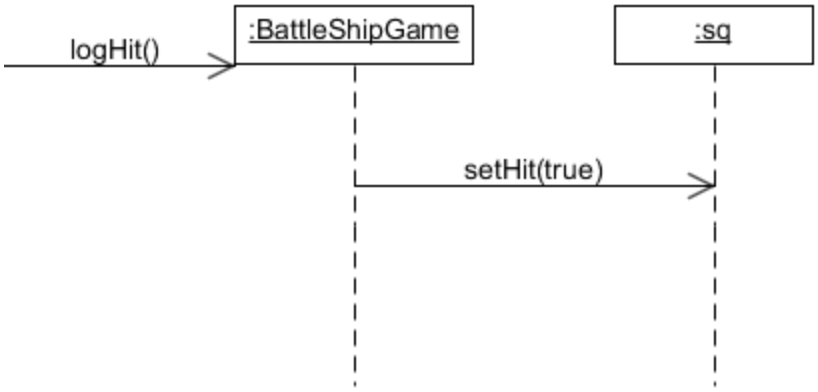
Sequence for C11

Refer back to C11. The correct square was got there. Here I will refer directly to this square

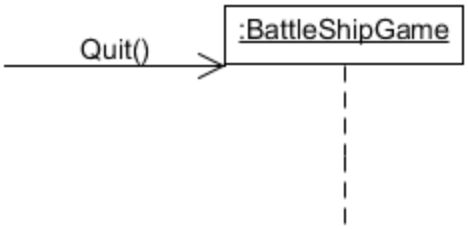


Sequence for C12

Refer back to C11. The correct square was got there. Here I will refer directly to this square



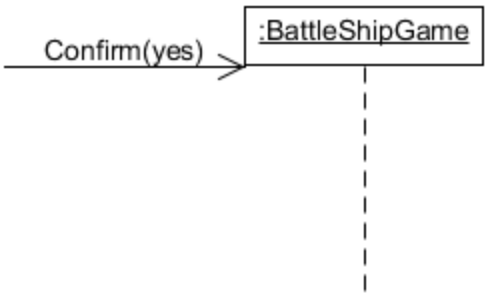
Sequence for C13



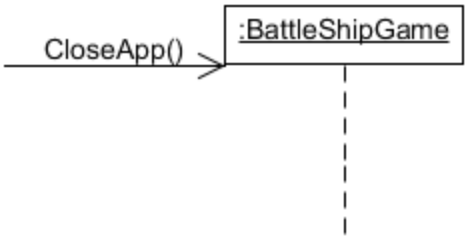
Sequence for C14

nothing done here. Gui shows a message that asks the player to confirm whether or not they want to quit the app.

Sequence for C15



Sequence for C16



Design Class Diagram