****

Yarmouk University

**Hijjawi Faculty for Engineering Technology  
Computer Engineering Department  
CPE-350: Object Oriented Modeling and Applications**

**Tennis Game Project**

**Name : Rana Walied Abu-Alhaija'a**

**ID : 2011980034**

**Submitted To: Dr. Hussein Al-Zoubi**

**Task #1: Domain Class Model**

* Preparing the problem statement;
* Identifying objects and classes;
* Developing a data dictionary;
* Identifying associations between classes;
* Identifying attributes of classes and association classes;
* Structure object classes using inheritance;
* Verify access paths for likely queries; and
* Iteratively refine the model.

**Step 1: Preparing the problem statement**

"The game is played by two players. Each of whom wields a racket, indicated by the black vertical lines on the short sides. Each player can move his or her racket up or down .The player on the right will use the key ↑ and ↓ and the player on the left will use A and Z keys as "up" and "down" keys. A red ball bounces back and forth (and up and down) in the playing area, and each player has to hit the ball with the racket so that the ball does not touch the player's short side. If this happened, the opponent gets a point. The point is displayed at the side of the playing area.

When play begins, the ball travels relatively slowly from the left but every time a player is able to hit the ball, its speed increases, making it more difficult to hit. The speed of the ball increases until one of the players misses it and it bounces against one of the short sides. The ball then returns to the speed it had all the beginning of the game.

The game begins when we click on the "New game" button. A game is interrupted automatically when one of the players has reached 10 points. A game can be temporarily interrupted if the "Pause" button is clicked and continued when the "continue" button is clicked. The "Exit" button terminates the game. The actual game played can be interrupted and a new game begun instead, by click on the "New game" button.

When the "New game" button is clicked , a new window pops up with three buttons, one with the text "One player", one with text "Two Players ", in addition to the "cancel" button. When the user clicks on the button "Two Player", the game will functions as before but when the user clicks on the button "One Player", the player on the left will be replaced by the computer. The program will then automatically move the left-hand racket so that it always hit the ball. Because the human player will not get any points , it will not matter that these are not shown; instead , the right-hand label should be allowed to display the number of times the human player has succeeded in hitting the ball before he or she loses by 10:0. A player's skills can then be judged by the number of hits made. The cancel button cancels starting a new game and returns to the previous state.

There is an aspect of the program that is not obvious from the figure. The playing area's width and length can be changed at any time by dragging on the window. So the game can be made easier or more difficult and play will be adjusted automatically to continue on the altered "court"."

**Step 2: Identifying objects and classes**

**-all objects and classes**

|  |  |
| --- | --- |
| Racket (tangible thing) | Red ball(attribute) |
| Ball (tangible thing) | Playing area (tangible) |
| Players (roles played) | speed(attribute) |
| New game(event) | Ball (tangible) |
| End game(event) | exit (event) |
| pause(event) | Continue (event) |
| Short side(concept) | Computer (roles played) |
| Width and length (attribute) | Position (attribute) |
| Black vertical line(attribute) | Control key (concept) |
| Points (concept) | Right player (roles played) |
| Left player (roles played) | Number of hits (concept) |
| Game (concept) | Moving (concept) |

**-The Generalization Relationship**

|  |  |
| --- | --- |
| player(role played) | Racket (concept) |
| Ball (concept) | Playing area(concept) |
| Control key (concept) | Game (concept) |

**-Inappropriate Classes**

|  |  |
| --- | --- |
| Black vertical line (attribute) | Red ball(attribute) |
| Position (attribute) | speed(attribute) |
| New game(operation) | Number of hits (redundant) |
| End game(operation) | exit (operation) |
| pause(operation) | Continue (operation) |
| Width and length (attribute) | Playing area (redundant) |

-**Candidate Classes**

|  |  |
| --- | --- |
| Racket (tangible thing) | Game (concept) |
| Ball (tangible thing) | Control key (concept) |
| Players (roles played) | Short side(concept) |
| Points (concept) | Right player (roles played) |
| Left player (roles played) | Computer (roles played) |
| Moving (concept) |  |

**Step 3: Developing the Data Dictionary**

|  |  |
| --- | --- |
| Class | Definition |
| Racket | Hits the ball |
| Ball | Ball is a basic element in this game, moving from one player to another |
| Players | It's the basic roles played in this game . it will use the racket to hit ball |
| Points | Counting number of points will the player get. |
| Game | the main application class |
| Left player | The player that will be in the left of playing area and use A , Z keys , it can replaced by computer |
| Right player | The player that will be in the right of playing area and use ↑ ,↓ keys |
| Computer | Its role as a player it will be in the place of the left player if "one player" option will be chosen |
| Control key | Its specifies the control keys of the racket according to the type of player |
| Short side | It's important part of the game. if the ball touch the short side of one player , the opponent will get a point |
| moving | specifies the initial position and the color and shape for moving object(racket and ball) |

**Step 4: Identifying associations between classes**

|  |  |
| --- | --- |
| **Verb phrase** | **Association** |
| The game is played by two players. | Played by |
| The game is played by two players. Each of whom wields a racket | Wields, control |
| a racket that indicated by the black vertical lines on the short sides | Place in |
| Each player can move his or her racket up or down .The player on the right will use the key ↑ and ↓ and the player on the left will use A and Z keys as "up" and "down" keys | Using |
| player has to hit the | Hit |
| the ball with the racket so that the ball does not touch the player's short side. If this happen, the opponent gets a point. | Touch, get |

# class diagram.jpg

**Task #2: Use Case Model**

* Identify the major actors;
* Write description to define the roles of each actor;
* Examine the roles of each actor and identify the use cases;
* Draw initial use case diagram;
* Write initial descriptions for the use cases;
* Perform a textual analysis to identify candidate business (domain) objects;
* Develop the base use case descriptions; and
* Iteratively elaborate the base use case descriptions and determine the <<extend>>, <<include>> and generalization relationships. Refine the use case diagram and the use case description to reflect the use case relationships.

**Step 1: Identifying the major actors and use cases**

* **There are two main actors:**

1. **Human player**

**- Left player**

**- Right player**

1. **Computer**

**Step** 2: Write description to define the roles of each actor

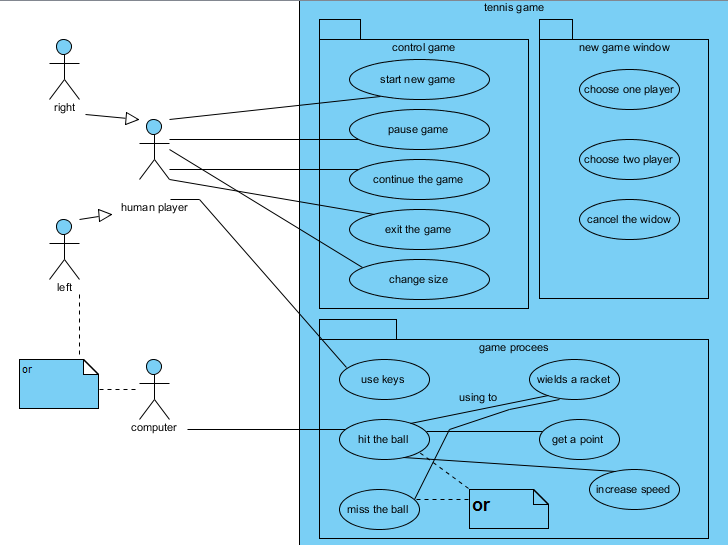
|  |  |
| --- | --- |
| Human player | **Actor name** |
| This actor can control the game by start, pause, exit, continue and change the size the window of the game. She/he can choose one player, two players or cancel button. And hit the ball by wield a racket using the keys | **Description** |

|  |  |
| --- | --- |
| computer | **Actor name** |
| This system actor can only hit the ball automatically by implementing a code for that. | **Description** |

**Step 3: Identify the use cases**

1. **Start new game**
2. **Pause game**
3. **Continue the game**
4. **Exit game**
5. **Choose one player**
6. **Choose two player**
7. **Cancel the window**
8. **Change size**
9. **Hit the ball**
10. **Wields a racket**
11. **Use keys**
12. **Increase speed**
13. **Get a point**
14. **Miss the ball**

**Step 4: creating an initial use case diagram**

****

**Step 5: initial description of the use cases**

|  |
| --- |
| **Use case:** start new game |
| **Use case ID:** UC01 |
| **Actor:** human player |
| **Description:** the human player clicks on new game button, then a window will appear with three options: one player, two players and cancel. she/ he can choose any one of them. |

|  |
| --- |
| **Use case:** pause game |
| **Use case ID:** UC02 |
| **Actor:** human player |
| **Description:** the human player clicks on pause button. This button will interrupt the game and everything will stop. |

|  |
| --- |
| **Use case:** continue the game |
| **Use case ID:** UC03 |
| **Actor:** human player |
| **Description:** the human player clicks on continue button. This button will continue the interrupted game and everything will continue from the last state. |

|  |
| --- |
| **Use case:** exit the game |
| **Use case ID:** UC04 |
| **Actor:** human player |
| **Description:** the human player clicks on exit button. This button will exit from the game and everything will not be saved for the next game the player will start |

|  |
| --- |
| **Use case:** choose one player |
| **Use case ID:** UC05 |
| **Actor:** - |
| **Description:** this is a choice will appear if the human player click on "new game" button. When click on "one player" button the player on the left will be replaced by the computer. The right player will use his/her keys |

|  |
| --- |
| **Use case:** choose two players |
| **Use case ID:** UC06 |
| **Actor:** - |
| **Description:** this is a choice will appear if the human player click on "new game" button. When click on "two players" button two human players will play using their keys. |

|  |
| --- |
| **Use case:** cancel the window |
| **Use case ID:** UC07 |
| **Actor:** - |
| **Description:** this is a choice will appear if the human player click on "new game" button. When click on "cancel" button it cancels starting a new game and returns to the previous state. |

|  |
| --- |
| **Use case:** wields a racket |
| **Use case ID:** UC08 |
| **Actor:**  - |
| **Description:** human player use keys to wields a racket (up or down)to hit the ball (keys depend on player name) |

|  |
| --- |
| **Use case:** use keys |
| **Use case ID:** UC09 |
| **Actor:**  human player |
| **Description:** human player use keys to control the movement of the racket |

|  |
| --- |
| **Use case:** hit the ball |
| **Use case ID:** UC10 |
| **Actor:**  - |
| **Description:** when using the keys to control the racket if the racket touches the ball then this action called hitting the ball. And the ball will travel to the opponent side |

|  |
| --- |
| **Use case:** get a point |
| **Use case ID:** UC11 |
| **Actor:**  - |
| **Description:** when using the keys to control the racket if the racket doesn’t touch the ball (i.e. ball touch the short side) the opponent gets a point. Its important to know that happen if two players option was chosen. |

|  |
| --- |
| **Use case:** miss the ball |
| **Use case ID:** UC12 |
| **Actor:**  - |
| **Description:** when using the keys to control the racket if the racket doesn’t touch the ball (i.e. ball touch the short side) this action called ball missing. |

|  |
| --- |
| **Use case:** change size |
| **Use case ID:** UC13 |
| **Actor:**  human player |
| **Description:** The playing area's width and length can be changed at any time by dragging on the window |

|  |
| --- |
| **Use case:** increase speed |
| **Use case ID:** UC14 |
| **Actor:**  - |
| **Description:** When play begins, the ball travels relatively slowly from the left but every time a player is able to hit the ball, its speed increases |

**Step 6: Perform a textual analysis to identify candidate business (domain) objects**

* **It's done before in task #1.**

**Step 7: Developing the base use case description**

|  |  |
| --- | --- |
| **Use case name** | Start new game |
| **Use case ID** | UC01 |
| **Super use case** | - |
| **Actor(s)** | Human player |
| **Brief description** | The human player clicks on new game button. |
| **Preconditions** | Human player must run the application game |
| **Post-conditions** | A window will appear with three options |
| **Flow of events** | 1. Human player will run the tennis game 2. Click new game button 3. Window will appear with three choices 4. Extend(choose one player), (choose two player),(cancel the widow) |
| **Alternative flows and exceptions** | At any time the human player can choose it even that a game currently run |
| **Priority** | High |

|  |  |
| --- | --- |
| **Use case name** | pause game |
| **Use case ID** | UC02 |
| **Super use case** | - |
| **Actor(s)** | Human player |
| **Brief description** | The human player clicks on pause button. This button will interrupt the game and everything will stop. |
| **Preconditions** | Human player must run the application game |
| **Post-conditions** | Interrupt the game and everything will stop. |
| **Flow of events** | 1. Human player will run the tennis game 2. Click pause button |
| **Alternative flows and exceptions** | At any time the human player can choose it even that a game currently run |
| **Priority** | High |

|  |  |
| --- | --- |
| **Use case name** | continue the game |
| **Use case ID** | UC03 |
| **Super use case** | - |
| **Actor(s)** | Human player |
| **Brief description** | The human player clicks on continue button. This button will continue the interrupted game and everything will continue from the previous state. |
| **Preconditions** | Human player must run the application game |
| **Post-conditions** | Continue the interrupted game and everything will continue from the previous state. |
| **Flow of events** | 1. Human player will run the tennis game 2. Click continue button |
| **Alternative flows and exceptions** | This button used after click "pause" button , otherwise it will not affect any thing |
| **Priority** | High |

|  |  |
| --- | --- |
| **Use case name** | exit the game |
| **Use case ID** | UC04 |
| **Super use case** | - |
| **Actor(s)** | Human player |
| **Brief description** | The human player clicks on exit button. This button will exit from the game . |
| **Preconditions** | Human player must run the application game |
| **Post-conditions** | This button will exit from the game and everything will not be saved for the next game the player will start |
| **Flow of events** | 1. Human player will run the tennis game 2. Click exit button |
| **Alternative flows and exceptions** | - |
| **Priority** | High |

|  |  |
| --- | --- |
| **Use case name** | choose one player |
| **Use case ID** | UC05 |
| **Super use case** | - |
| **Actor(s)** | Human player |
| **Brief description** | This is a choice will appear if the human player click on "new game" button. When click on "one player" button the player on the left will be replaced by the computer. The right player will use his/her keys |
| **Preconditions** | Human player must run the application game and click "new game button" |
| **Post-conditions** | The player on the left will be replaced by the computer. The right player will use his/her keys |
| **Flow of events** | 1. Human player will run the tennis game 2. Click "new game" button 3. Choose "one player" button |
| **Alternative flows and exceptions** | - |
| **Priority** | low |

|  |  |
| --- | --- |
| **Use case name** | choose two players |
| **Use case ID** | UC06 |
| **Super use case** | - |
| **Actor(s)** | Human player |
| **Brief description** | This is a choice will appear if the human player click on "new game" button. When click on "two players" button two human players will play using their keys. |
| **Preconditions** | Human player must run the application game and click "new game button" |
| **Post-conditions** | two human players will play using their keys |
| **Flow of events** | 1. Human player will run the tennis game 2. Click "new game" button 3. Choose "two players" button |
| **Alternative flows and exceptions** | - |
| **Priority** | low |

|  |  |
| --- | --- |
| **Use case name** | cancel the window |
| **Use case ID** | UC07 |
| **Super use case** | - |
| **Actor(s)** | Human player |
| **Brief description** | this is a choice will appear if the human player click on "new game" button. When click on "cancel" button it cancels starting a new game and returns to the previous state. |
| **Preconditions** | Human player must run the application game and click "new game button" |
| **Post-conditions** | Cancels starting a new game and returns to the previous state. |
| **Flow of events** | 1. Human player will run the tennis game 2. Click "new game" button 3. Choose "cancel" button |
| **Alternative flows and exceptions** | - |
| **Priority** | high |

|  |  |
| --- | --- |
| **Use case name** | wields a racket |
| **Use case ID** | UC08 |
| **Super use case** | Use keys |
| **Actor(s)** | Human player |
| **Brief description** | human player use keys to wields a racket (up or down)to hit the ball (keys depend on player name) |
| **Preconditions** | Human player must run the application game and click "new game button" and use the keys |
| **Post-conditions** | Moving the racket up and down to hit the ball |
| **Flow of events** | 1. Human player will run the tennis game 2. Click "new game" button 3. Control the keys that Inherits from use keys |
| **Alternative flows and exceptions** | - |
| **Priority** | High |

|  |  |
| --- | --- |
| **Use case name** | use keys |
| **Use case ID** | UC09 |
| **Super use case** | - |
| **Actor(s)** | Human player |
| **Brief description** | human player use keys to control the movement of the racket |
| **Preconditions** | Human player must run the application game and click "new game button" and use the keys |
| **Post-conditions** | Moving the racket up and down to hit the ball |
| **Flow of events** | 1. Human player will run the tennis game 2. Click "new game" button 3. Control the keys |
| **Alternative flows and exceptions** | - |
| **Priority** | High |

|  |  |
| --- | --- |
| **Use case name** | hit the ball |
| **Use case ID** | UC10 |
| **Super use case** | - |
| **Actor(s)** | Human player |
| **Brief description** | When using the keys to control the racket if the racket touches the ball then this action called hitting the ball. And the ball will travel to the opponent side |
| **Preconditions** | Human player must run the application game and click "new game button" and use the keys to move the racket |
| **Post-conditions** | Hit the ball and send it to the opponent side |
| **Flow of events** | 1. Human player will run the tennis game 2. Click "new game" button 3. Include (wield a racket) 4. Send the ball to opponent side |
| **Alternative flows and exceptions** | - |
| **Priority** | High |

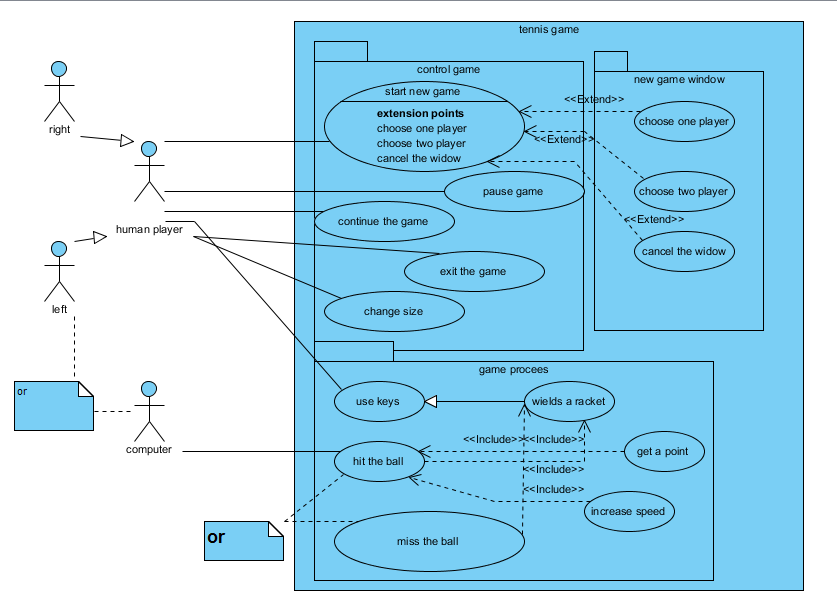
|  |  |
| --- | --- |
| **Use case name** | get a point |
| **Use case ID** | UC11 |
| **Super use case** | - |
| **Actor(s)** | - |
| **Brief description** | When using the keys to control the racket if the racket doesn’t touch the ball (i.e. ball touch the short side) the opponent gets a point. It's important to know that happen if two players option was chosen. |
| **Preconditions** | Human player must run the application game and click "new game button" choose two players and use the keys to move the racket and hit the ball successfully |
| **Post-conditions** | A point will be added to the total. |
| **Flow of events** | 1. Human player will run the tennis game 2. Click "new game" button 3. Include (hit the ball) 4. Increase the score |
| **Alternative flows and exceptions** | A game is interrupted automatically when one of the players has reached 10 points |
| **Priority** | High |

|  |  |
| --- | --- |
| **Use case name** | miss the ball |
| **Use case ID** | UC12 |
| **Super use case** | - |
| **Actor(s)** | - |
| **Brief description** | When using the keys to control the racket if the racket doesn’t touch the ball (i.e. ball touch the short side) this action called ball missing. |
| **Preconditions** | Human player must run the application game and click "new game button" choose two players and use the keys to move the racket. |
| **Post-conditions** | A point will be added to the opponent |
| **Flow of events** | 1. Human player will run the tennis game 2. Click "new game" button 3. Include (wield a racket) 4. Increase the score |
| **Alternative flows and exceptions** | - |
| **Priority** | High |

|  |  |
| --- | --- |
| **Use case name** | Change size |
| **Use case ID** | UC13 |
| **Super use case** | - |
| **Actor(s)** | Human player |
| **Brief description** | The playing area's width and length can be changed at any time by dragging on the window |
| **Preconditions** | Human player must run the application drag the window |
| **Post-conditions** | Change the size of the window. So the game can be made easier or more difficult |
| **Flow of events** | 1. Human player will run the tennis game 2. Drag the window |
| **Alternative flows and exceptions** | - |
| **Priority** | Low |

|  |  |
| --- | --- |
| **Use case name** | increase speed |
| **Use case ID** | UC14 |
| **Super use case** | - |
| **Actor(s)** | - |
| **Brief description** | the ball travels relatively slowly from the left but every time a player is able to hit the ball, its speed increases |
| **Preconditions** | Hitting the ball successfully |
| **Post-conditions** | Increase the speed |
| **Flow of events** | 1. Include (hit the ball) 2. Increase the speed |
| **Alternative flows and exceptions** | - |
| **Priority** | high |

# Step 8: Elaborate the base use case description



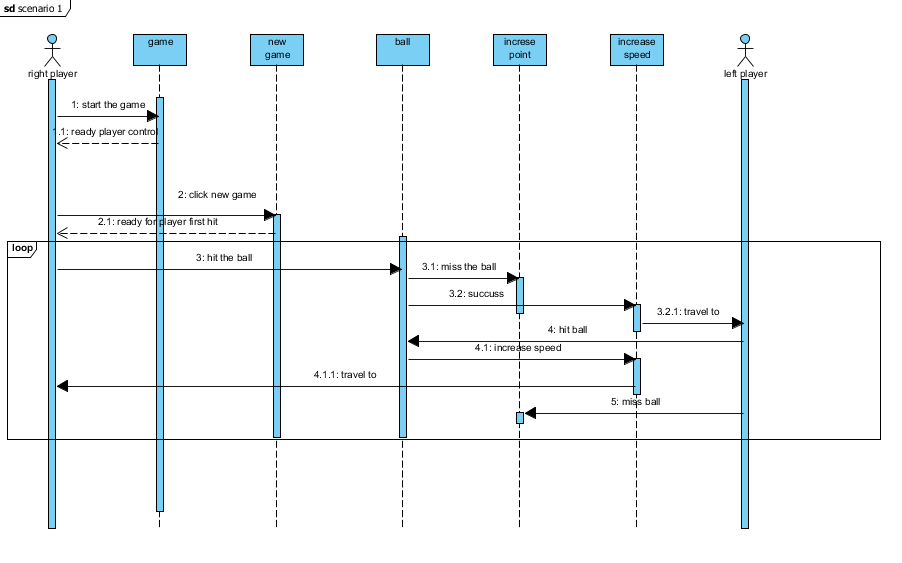
**Task #3: Dynamic Modeling and Analysis**

* Developing system- level sequence diagrams for use case scenarios.
* Developing three –tier sequence diagrams for the use case scenarios.
* Developing three – tier collaboration diagrams for the use case scenario (optional) for the use case scenarios.
* Developing a state chart diagram for each of the control objects.
* Refining the class diagram that you have developed in task #1 by using the result of steps 2 to 4 above.

**Step 1:- Developing system-level sequence diagram**

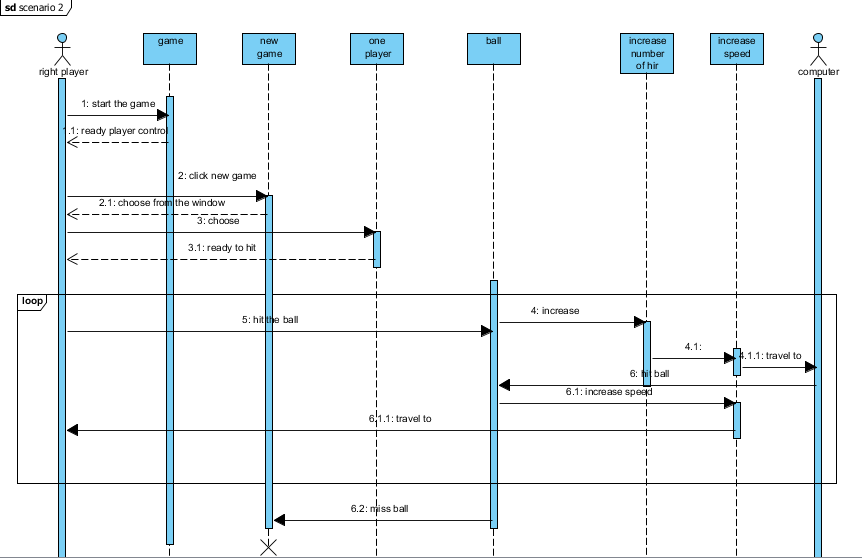
**- Scenario #1:**

When the user clicks "new game" button for the first time. The game is played by two players. Each player can move his or her racket up or down using keys to hit the ball and for each successful hit ball's speed increase. When the one of the player gets 10 point the game ended.



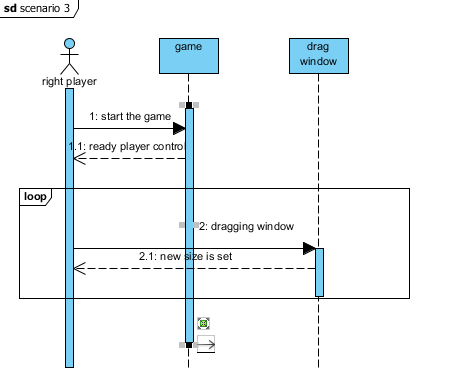
**- Scenario #2:**

When the user clicks on the button "One Player", then the right player will play with the computer. The program will then automatically move the left-hand racket so that it always hit the ball. Because the human player will not get any points; player's skills can then be judged by the number of hits made. The game ended when the right player miss the ball.

****

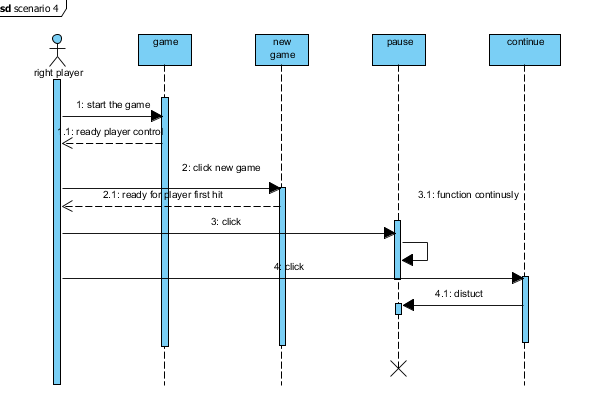
**- Scenario #3:**

The playing area's width and length can be changed at any time by dragging on the window. So the game can be made easier or more difficult and play will be adjusted automatically to continue on the altered "court".

****

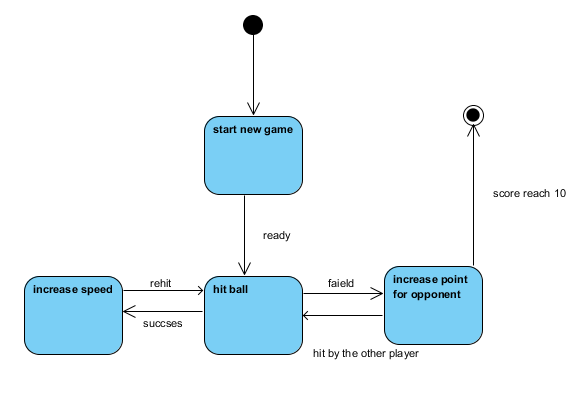
**- Scenario #4:**

After starting a new game, "Pause" button is clicked. Then the game will wait for player to click "continue" button. After this the game will continued from the previous state

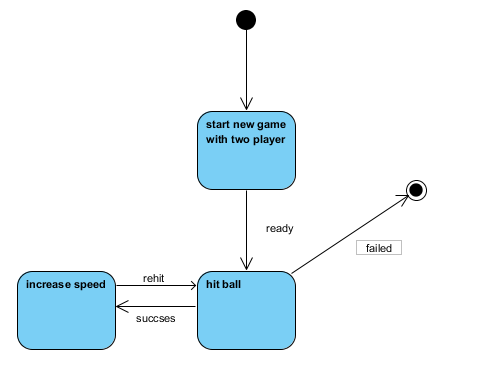
****

**Step2: Developing a state chart diagram for each of the control objects.**

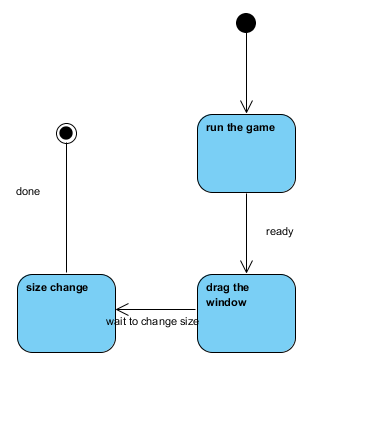
* **Scenario #1**

****

* **Scenario #2**

****

* **Scenario #3**

****

* **Scenario #4**

