

PROJECT DELIVERABLE 2



PROJECT DELIVERABLE 2

BY –

SOHAIL AHMED (ID 38879972)

EXERCISE 2 USE CASE SPECIFICATION

1) USE CASE – DISPLAY BOARD

DESCRIPTION:

- The use case represents the situation where the system displays the puzzle board

PRECONDITION:

- The game program is started

POSTCONDITION:

- The puzzle is ready to be displayed

BASICPATH:

- The system initializes the board
- the player(user) runs the program

ALTERNATIVEPATH:

- If the system fails to initialize the board , it shows an error message

EXCEPTION PATH:

- If the system fails to initialize the board , it shows an error message

2) USE CASE – START GAME

DESCRIPTION:

- The use case represents the situation where the game is started

PRECONDITION:

- The puzzle board is ready to be displayed

POSTCONDITION:

- The puzzle board is displayed

BASICPATH:

- the user runs the program
- the system displays the puzzle board

ALTERNATIVEPATH:

- If the system fails to initialize the board , it shows an error message

EXCEPTION PATH:

- If the system fails to initialize the board , it shows an error message

3) USE CASE – SET INITIAL POSITION**DESCRIPTION:**

- The use case represents the situation where the initial position of the puzzle pieces is set on the board

PRECONDITION:

- The game program is started

POSTCONDITION:

- The some pieces are displayed on the board

BASICPATH:

- when the user runs the program
- the board is initialized

- the some puzzle pieces are displayed on the board

ALTERNATIVEPATH:

- If the system fails to display the puzzle pieces on the board , it shows an error message

EXCEPTION PATH:

- If the system fails to display the puzzle pieces on the board , it shows an error message

4) USE CASE – OPTION TO SEE INITIAL POSIOTION OF PIECES

DESCRIPTION: The use case represents the situation where the user has the choice to see the position of the all puzzle pieces

PRECONDITION:

- the program is started
- the board Is displayed
- the some puzzle pieces are displayed on the board

POSTCONDITION:

- The entire puzzle image is displayed

BASIC PATH:

- the system initializes the board
- the system places the initial pieces on their respective positions on the board
- the puzzle pieces are shown in the right order

ALTERNATIVEPATH:

- the user does not choose the option and thus only some puzzle pieces are shown on the board

EXCEPTION PATH:

- If the system fails to initialize the board or display the complete puzzle , it shows an error message

5) USE CASE – SELECT THE RIGHT PIECE

DESCRIPTION:

- The use case represents the situation where the user selects the right puzzle piece positions

PRECONDITION:

- The game program is started
- the board displays the incomplete puzzle
- option to see the full position of the puzzle

POSTCONDITION:

- The player chooses the right piece position
- the piece is added to the board
- new piece is generated

BASICPATH:

- the player selects a piece and places it on the board
- the system checks if the piece placed is correct
- if it is correct the system generates a new piece

ALTERNATIVEPATH:

- if the player chooses the wrong position for the piece than the system returns the piece back
- the player has to choose the position of the piece again

EXCEPTION PATH:

- If the system fails to initialize the board or check the position of image , it shows an error message

6) USE CASE – GENERATE NEW PIECE

DESCRIPTION:

- The use case represents the situation where the system generates a new puzzle piece

PRECONDITION:

- The player chooses the right piece

POSTCONDITION:

- A new piece is generated

BASICPATH:

- The system initializes the board
- Some puzzle pieces are displayed on the board
- The player chooses the right piece position
- The system checks the position
- New puzzle piece is generated

ALTERNATIVEPATH:

- The player chooses the wrong position and the piece is returned back to the player and
- No new piece is generated

EXCEPTION PATH:

- If the system fails to initialize the board or display the puzzle piece or generate a new piece , it shows an error message

7) USE CASE – END GAME

DESCRIPTION:

- The use case represents the situation where the player completes the puzzle

PRECONDITION:

- The system initializes the board
- Some puzzle pieces are displayed on the board
- the player chooses the right puzzle piece
- a new piece is generated
- the player then again chooses the right position till the picture puzzle is completed

POSTCONDITION:

- The game is over and the system displays a congratulations.

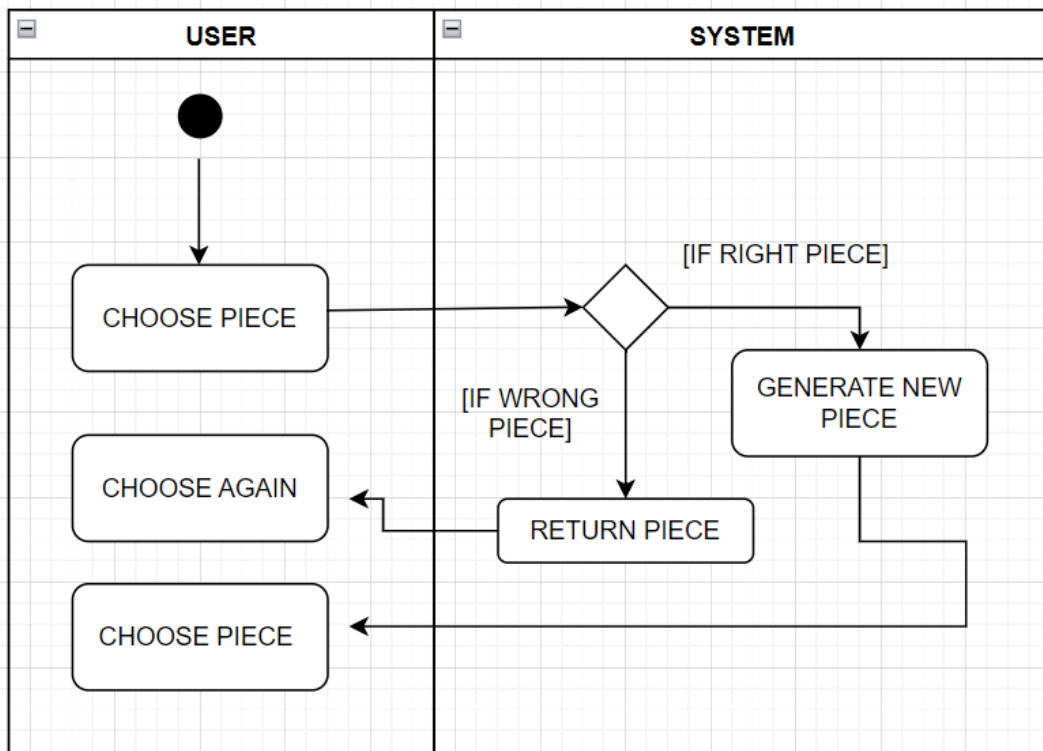
ALTERNATIVEPATH:

- if the player wishes to continue the game then the player can rerun the program and begin from the starting again

EXCEPTION PATH:

- If the system fails to initialize the board or display the puzzle pieces , generate new pieces , display congratulations message , it shows an error message

2) UML ACITIVITY DIAGRAM FOR THE SELECT RIGHT PIECE USE CASE



EXERCISE 1 FROM USE-CASE TO CLASSES

The list of all the entities that I can recognize from the use case diagram:-

- **PLAYER** – the player is the main actor who interacts with the picture puzzle game ;
- **Display board** – is a use case that is used to represent the picture puzzle board on which the game occurs
- **START GAME** – is a use case that represents the player's option or action to start playing the game
- **SET INITIAL PIECES** – is a use case that represents some pieces being arranged in the puzzle
- **OPTION TO SEE INITIAL POSITION OF PIECES** – is a use case that represents the option for the player to choose whether or not he or she can view the entire picture puzzle.
- **SELECT THE RIGHT PIECE** – is a use case that represents the action of the user(ACTOR) to choose the right puzzle piece position in the game
- **GENERATE NEW PIECE** – is an included use case that represents the generation of new pieces with each round as the player chooses the right puzzle piece position.
- **END GAME** – is a use case that represents the action of the player ending the puzzle game by completing it

2) IDENTIFYING THE MOST IMPORTANT RELATIONSHIPS BETWEEN THESE ENTITIES

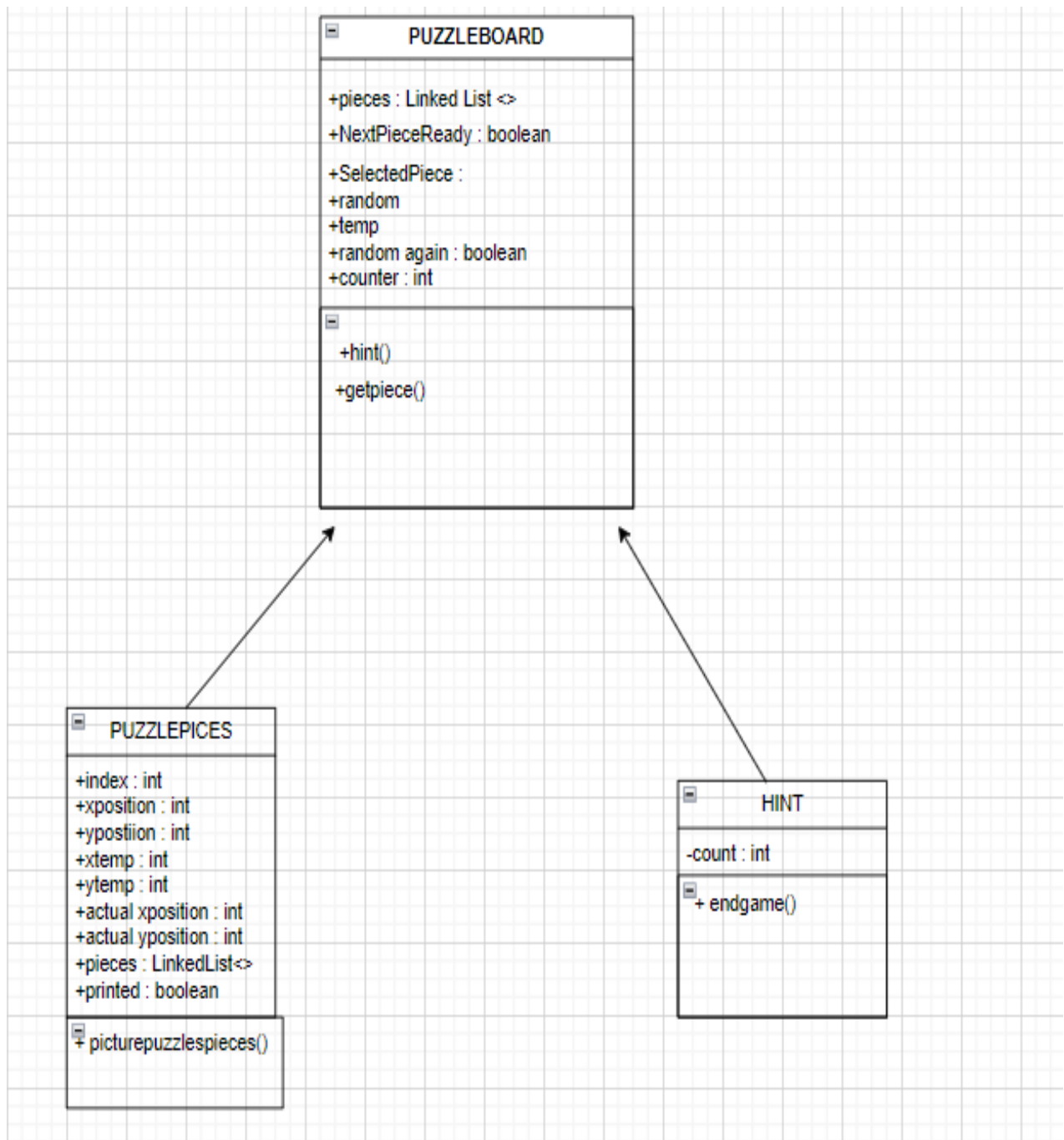
The entities of the use case diagram are not related by relation such as aggregation and inheritance.

However there is an <<includes>> relation between the “CHOOSE RIGHT PIECE” and “GENERATE NEW PIECE”.

The <<includes>> relation is used when one use case requires another use case for it to be functional. Here “CHOOSE RIGHT PIECE” is needed to generate a new piece which ultimately dictates the success of the game

Thus the most important relation I can identify from the entities is the <<includes>> relation CHOOSE RIGHT PIECE” and “GENERATE NEW PIECE”

3) UML CLASS DIAGRAM



// note: could not define these variables in draw.io so I have declared them down here//

+SelectedPiece : PUZZLEPIECES

+random : PUZZLEPIECES +temp : PUZZLEPIECES

EXERCISE 2: FROM CLASSES TO SEQUENCES

1) the most important use case that was identified from the uml activity diagram was the “CHOOSE RIGHT PIECE” use case. The following is a list of all the objects that is involved in the use case

- i. **The player object:** object that refers to the user (player).
- ii. **The display board object :** object that refers to the picture puzzle board that will display the pieces
- iii. **Puzzle board piece object :** object that refers to the puzzle pieces of the game
- iv. **Puzzle piece check object:** object that refers to the system that checks and validates the right puzzle piece positions.

2) The timely order of events and interactions between the objects are as follows

- The player object interacts with the display board object to view the puzzle board
- The puzzle board gets the puzzle pieces from the puzzle board piece object and displays on the board
- The player object then interacts with the puzzle piece object and selects the puzzle piece and places it on the board

- The puzzle piece check object checks if the puzzle piece placed was in the right position.
- If it was in the right position a new piece is generated for the player to choose.
- if it was wrong then the piece is returned back to the player to try again

3) UML SEQUENCE DIAGRAM

