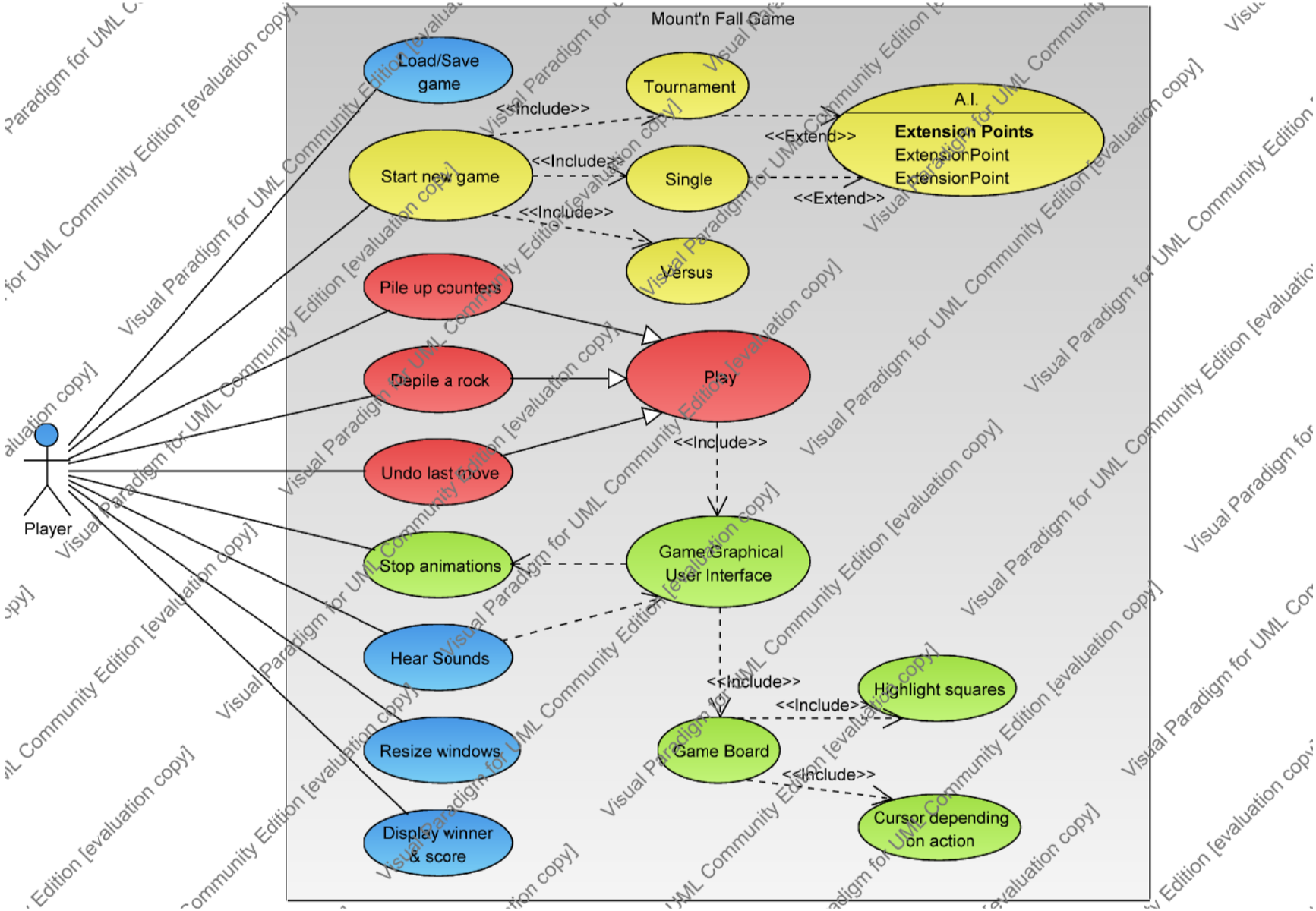
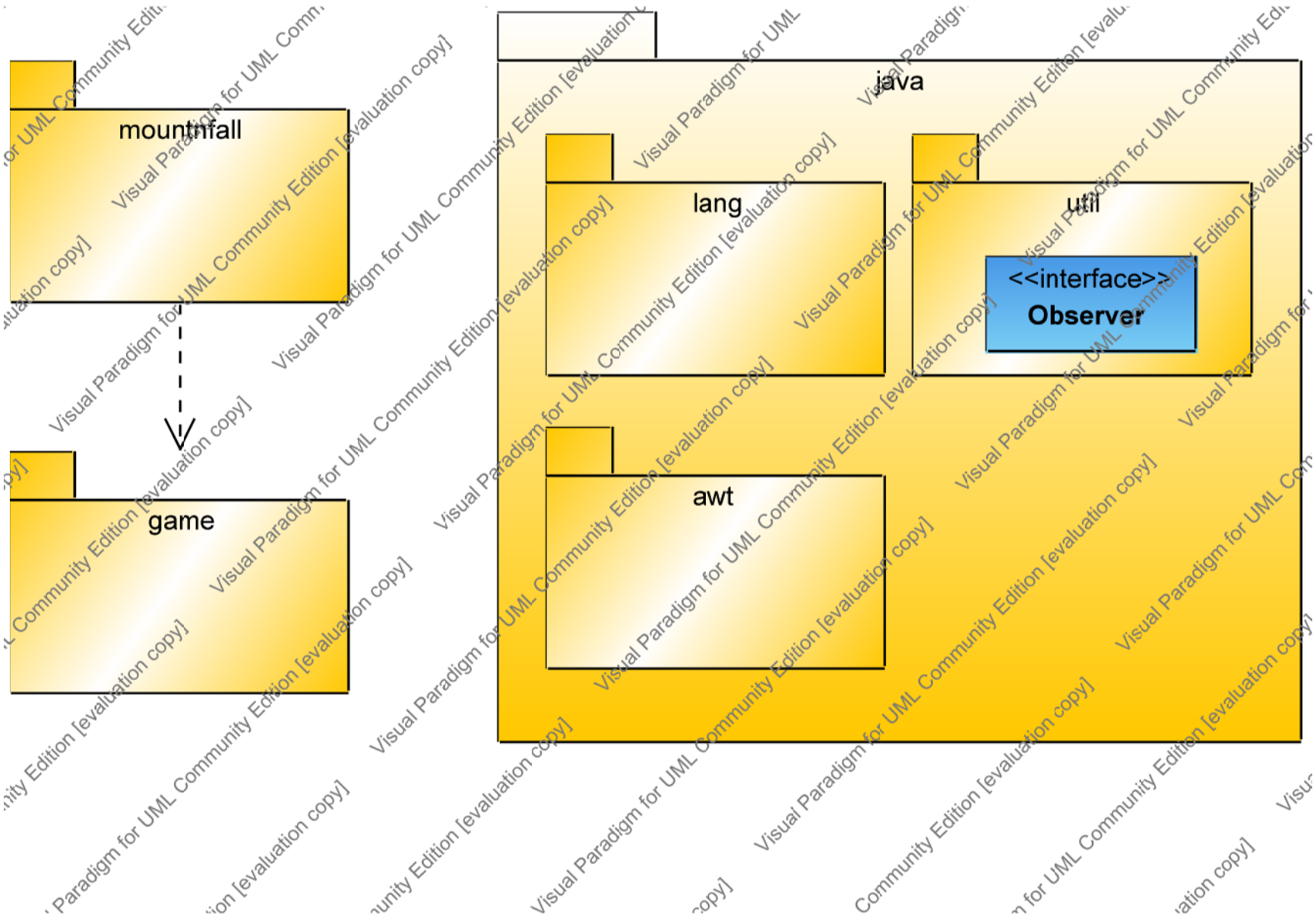
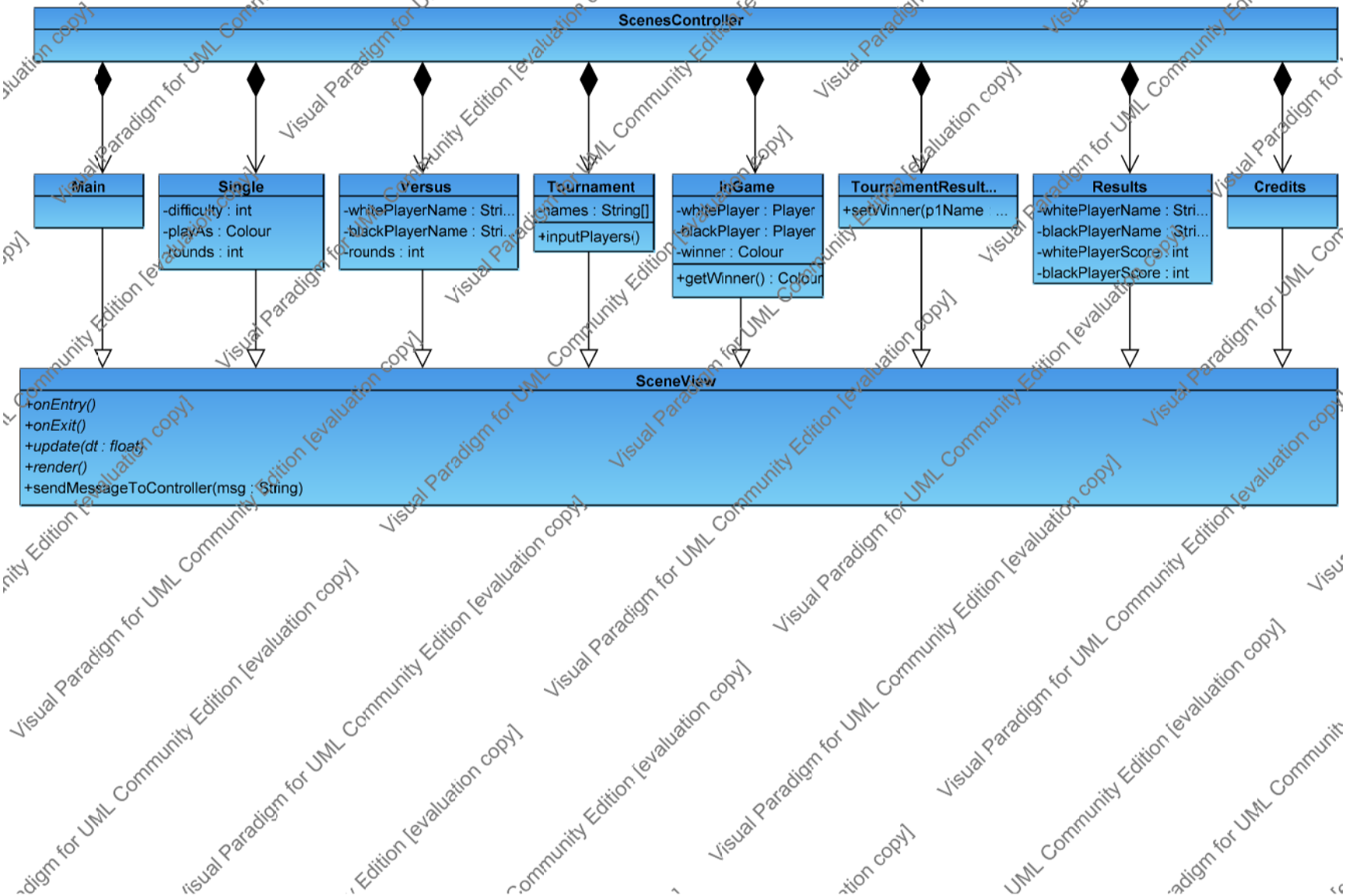
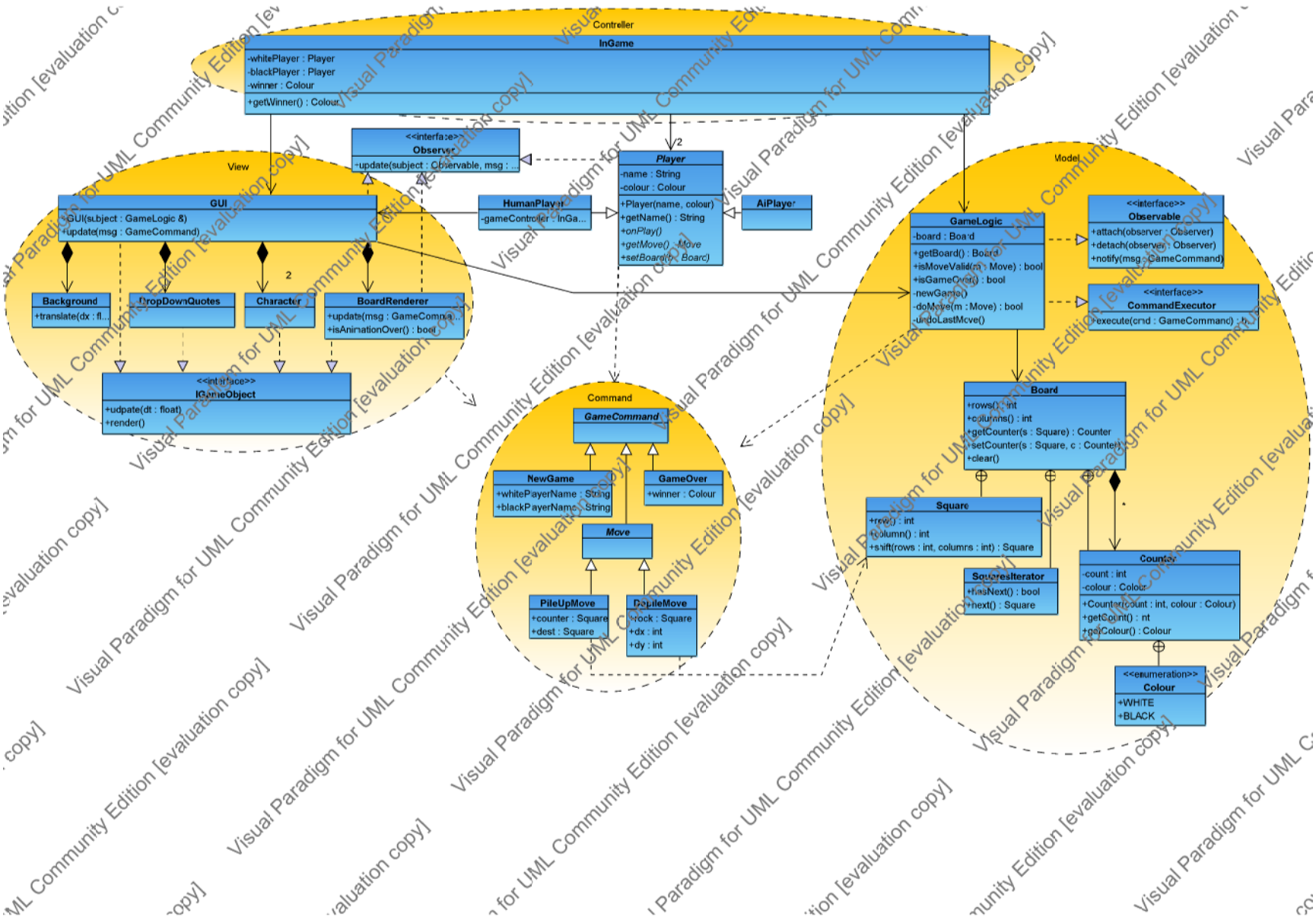
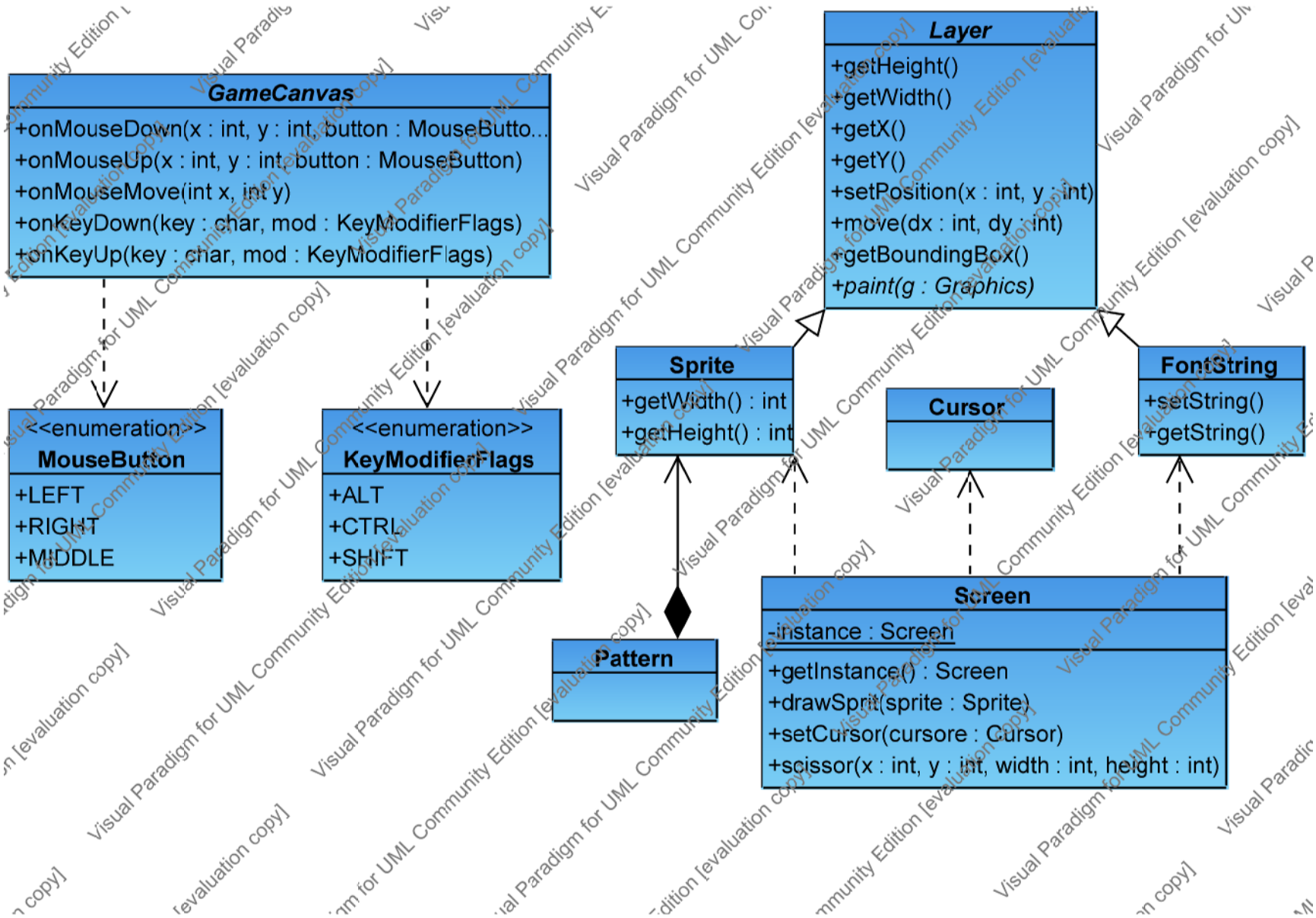

Mount'n Fall Game
Load/Save game
Tournament
A.I.
Extension Points
Extension Point
Extension Point
<<Include>>
<<Extend>>
Start new game
<<Include>>
Single
<<Extend>>
<<Include>>
Versus
Pile up counters
Depile a rock
Play
Undo last move
<<Include>>
Player
Stop animations
Game Graphical User Interface
Hear Sounds
<<Include>>
Highlight squares
<<Include>>
Resize windows
Game Board
<<Include>>
Cursor depending on action
Display winner & score

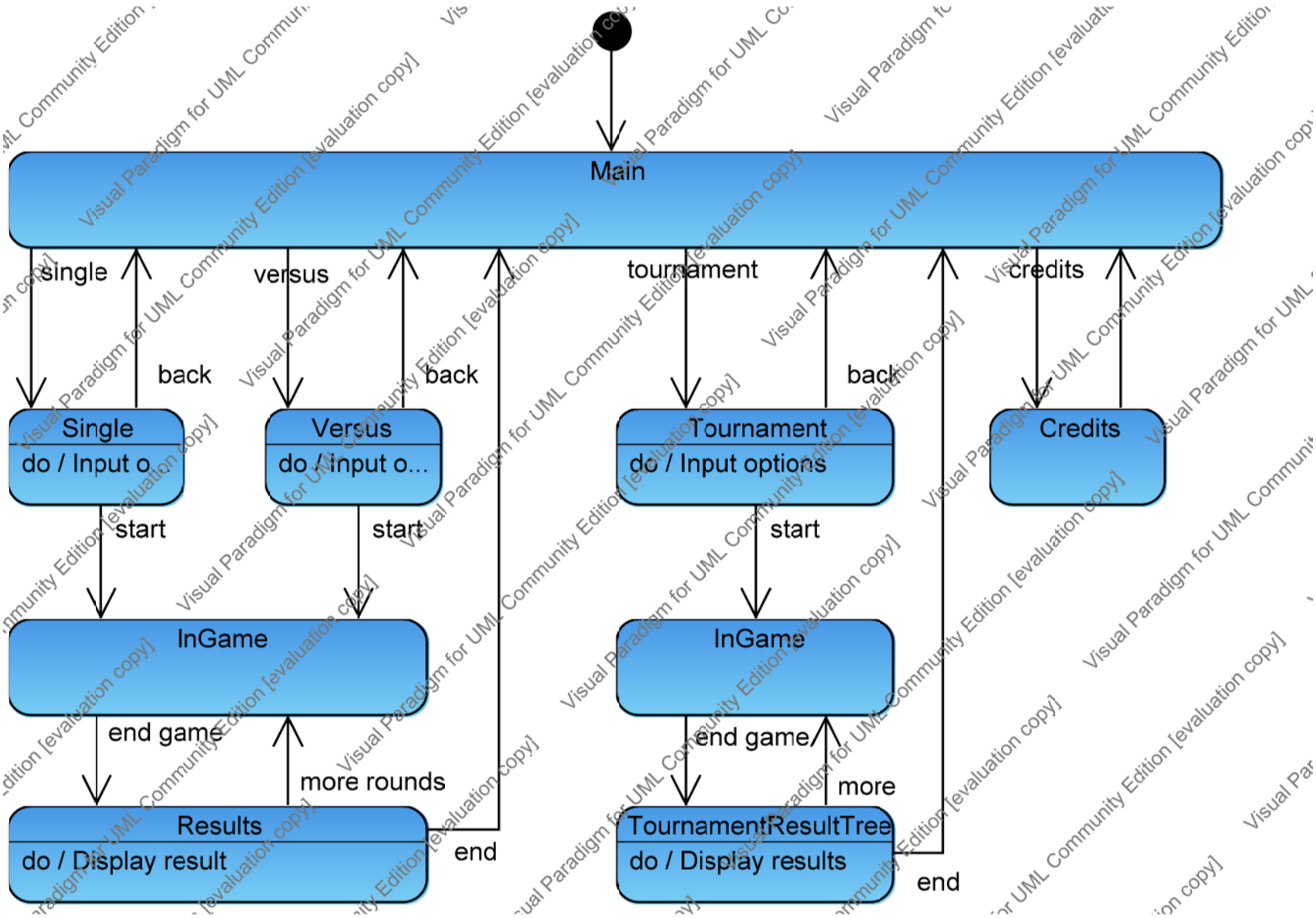

mountfall
game
java
lang
util
<<interface>>
Observer
awt

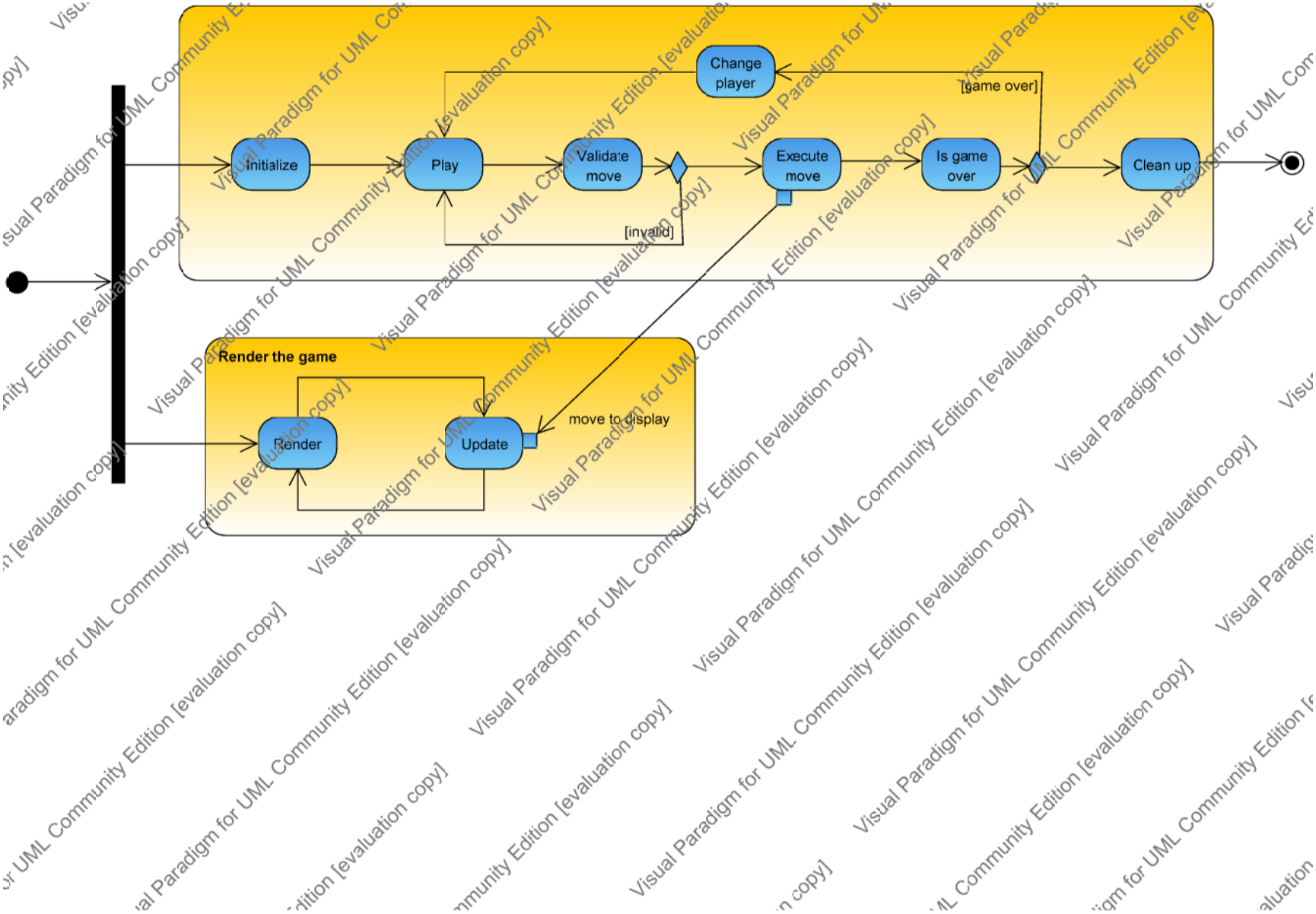


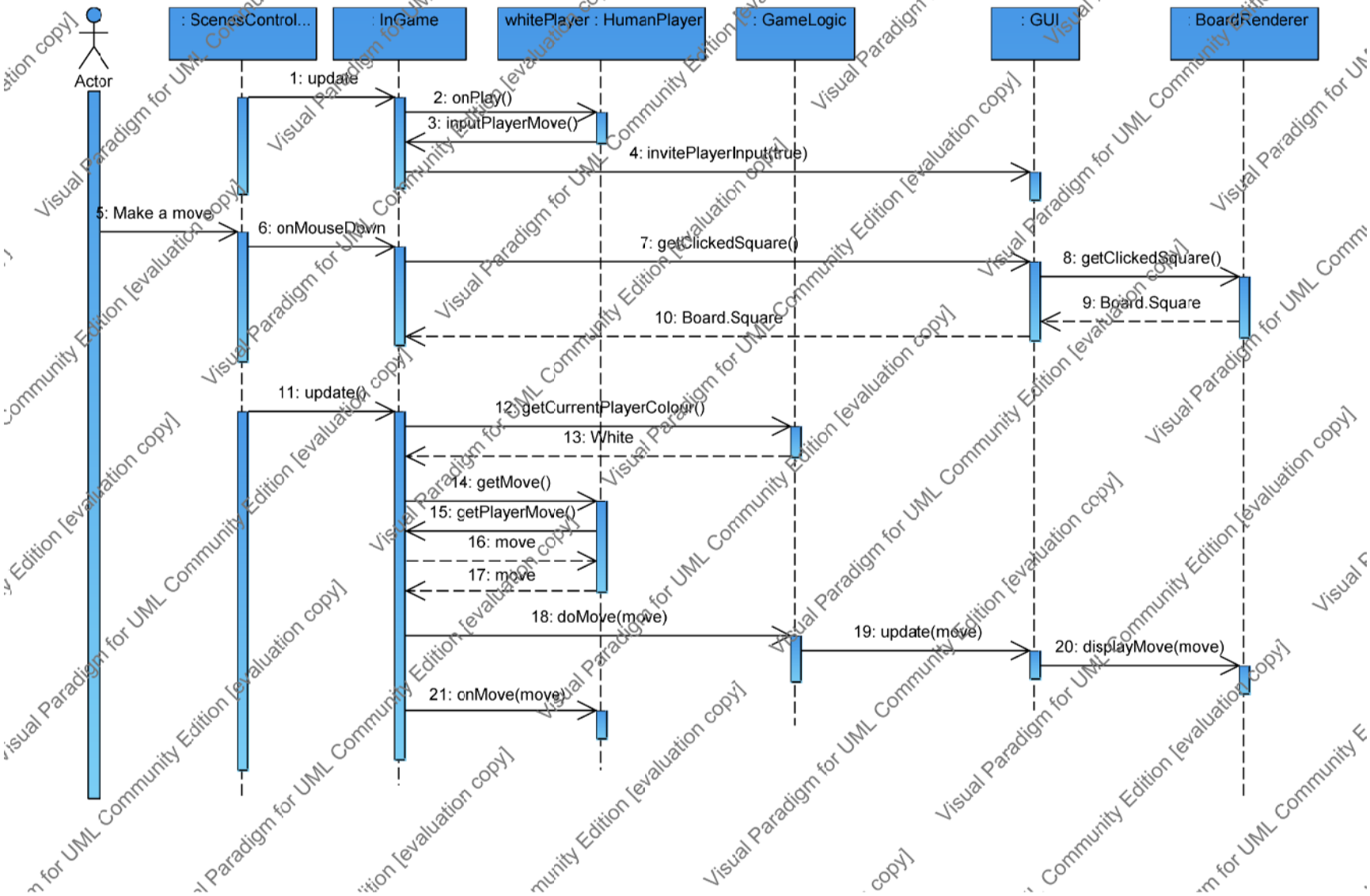
Visual Paradigm for UML Community Edition [evaluation copy]

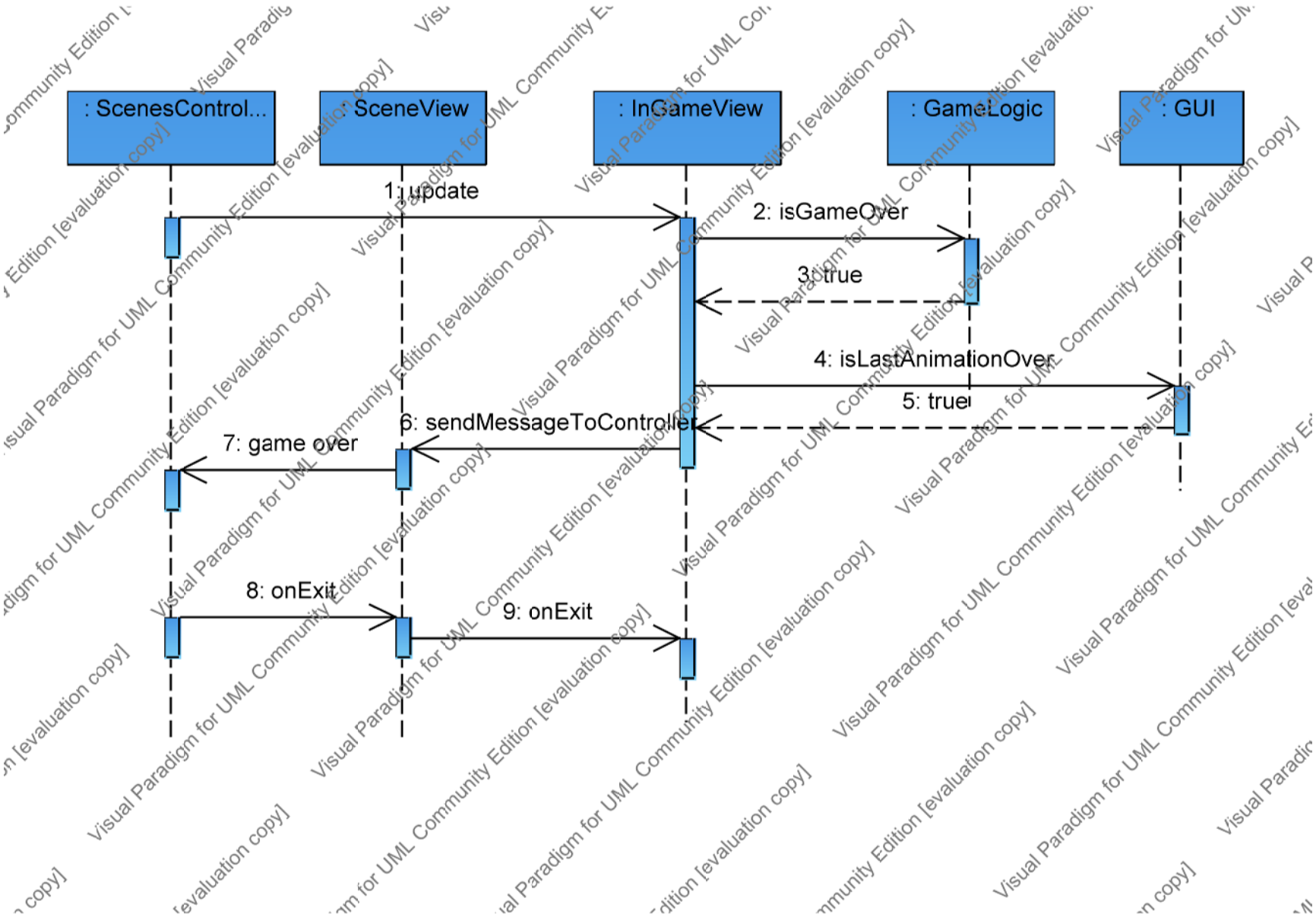

Controller
InGame
-whitePlayer : Player
-blackPlayer : Player
-winner : Colour
+getWinner() : Colour
<<interface>>
Observer
+update(subject : Observable, msg : ...
Player
-name : String
-colour : Colour
+Player(name, colour)
+getName() : String
+onPlay()
+getMove() : Move
+setBoard(b : Board)
HumanPlayer
-gameController : InGa...
AiPlayer
View
GUI
+GUI(subject : GameLogic &)
+update(msg : GameCommand)
Background
+translate(dx : fl...
DropDownQuotes
Character
BoardRenderer
+update(msg : GameCommand...
+isAnimationOver() : bool
<<interface>>
IGameObject
+udpate(dt : float)
+render()
Command
GameCommand
NewGame
+whitePlayerName : String
+blackPlayerName : String
GameOver
+winner : Colour
Move
PileUpMove
+counter : Square
+dest : Square
DoubleMove
+rock : Square
+dx : int
+dy : int
Model
GameLogic
-board : Board
+getBoard() : Board
+isMoveValid(m : Move) : bool
+isGameOver() : bool
-newGame()
-doMove(m : Move) : bool
-undoLastMove()
<<interface>>
Observable
+attach(observer : Observer)
+detach(observer : Observer)
+notify(msg : GameCommand)
<<interface>>
CommandExecutor
+execute(cmd : GameCommand) : b...
Board
+rows() : int
+columns() : int
+getCounter(s : Square) : Counter
+setCounter(s : Square, c : Counter)
+clear()
Square
+row() : int
+column() : int
+shift(rows : int, columns : int) : Square
SquaresIterator
+hasNext() : bool
+next() : Square
Counter
-count : int
-colour : Colour
+Counter(count : int, colour : Colour)
+getCount() : int
+getColour() : Colour
<<enumeration>>
Colour
+WHITE
+BLACK

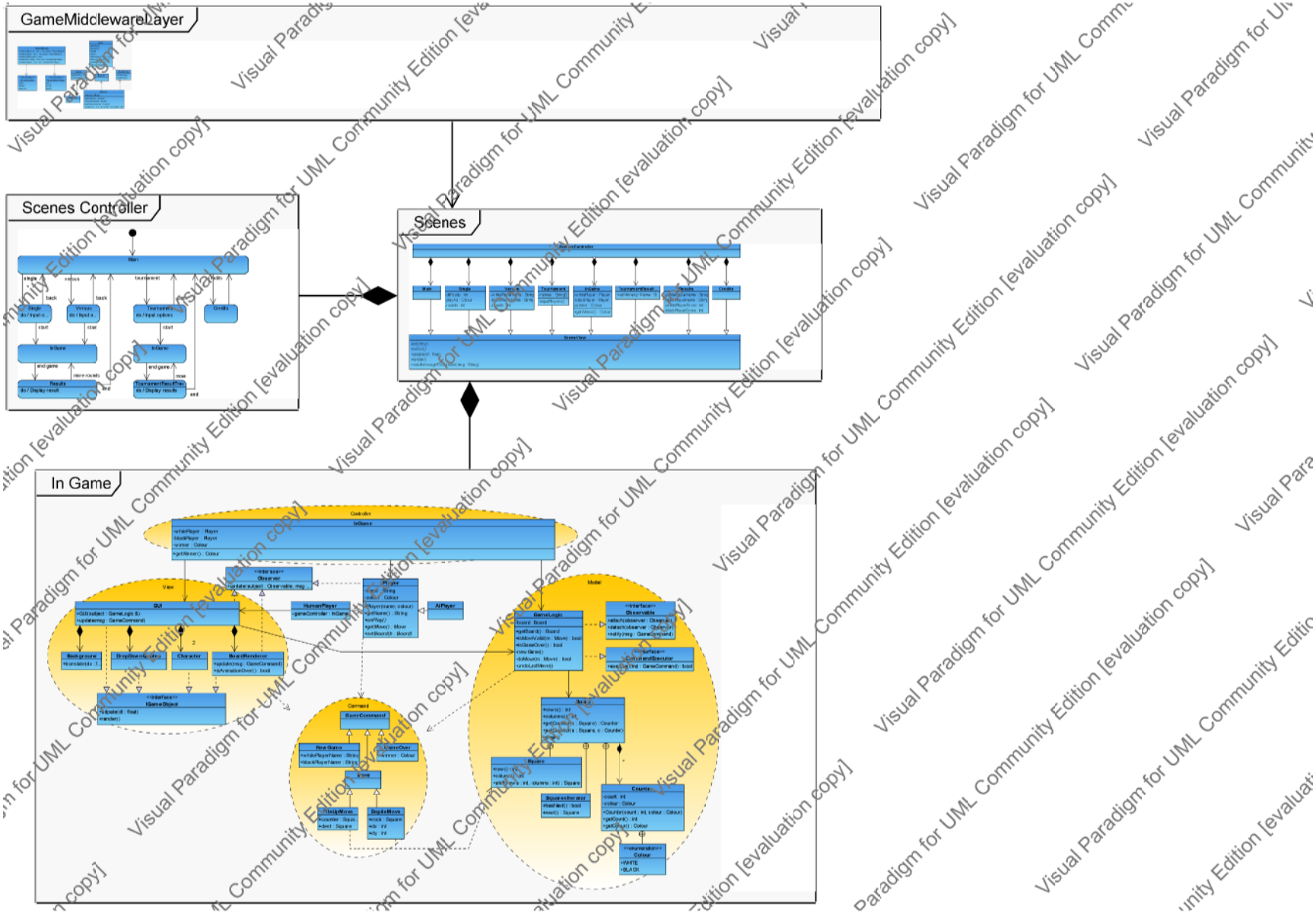

GameCanvas
+onMouseDown(x : int, y : int, button : MouseButto...
+onMouseUp(x : int, y : int, button : MouseButton)
+onMouseMove(int x, int y)
+onKeyDown(key : char, mod : KeyModifierFlags)
+onKeyUp(key : char, mod : KeyModifierFlags)
<<enumeration>>
MouseButton
+LEFT
+RIGHT
+MIDDLE
<<enumeration>>
KeyModifierFlags
+ALT
+CTRL
+SHIFT
Layer
+getHeight()
+getWidth()
+getX()
+getY()
+setPosition(x : int, y : int)
+move(dx : int, dy : int)
+getBoundingBox()
+paint(g : Graphics)
Sprite
+getWidth() : int
+getHeight() : int
Cursor
FontString
+setString()
+getString()
Pattern
Screen
-instance : Screen
+getInstance() : Screen
+drawSprite(sprite : Sprite)
+setCursor(cursore : Cursor)
+scissor(x : int, y : int, width : int, height : int)


Main
single
versus
tournament
credits
back
back
back
Single
do / Input o...
Versus
do / Input o...
Tournament
do / Input options
Credits
start
start
start
InGame
InGame
end game
end game
more rounds
more
Results
do / Display result
end
TournamentResultTree
do / Display results
end


Initialize
Play
Validate move
Change player
[game over]
[invalid]
Execute move
Is game over
Clean up
Render the game
Render
Update
move to display


Actor
: ScenesControl...
: InGame
whitePlayer : HumanPlayer
: GameLogic
: GUI
: BoardRenderer
1: update
2: onPlay()
3: inputPlayerMove()
4: invitePlayerInput(true)
5: Make a move
6: onMouseDown
7: getClickedSquare()
8: getClickedSquare()
9: Board.Square
10: Board.Square
11: update()
12: getCurrentPlayerColour()
13: White
14: getMove()
15: getPlayerMove()
16: move
17: move
18: doMove(move)
19: update(move)
20: displayMove(move)
21: onMove(move)


: ScenesControl...
: SceneView
: InGameView
: GameLogic
: GUI
1: update
2: isGameOver
3: true
4: isLastAnimationOver
5: true
6: sendMessageToController
7: game over
8: onExit
9: onExit


GameMidclewareLayer
Scenes Controller
Scenes
In Game