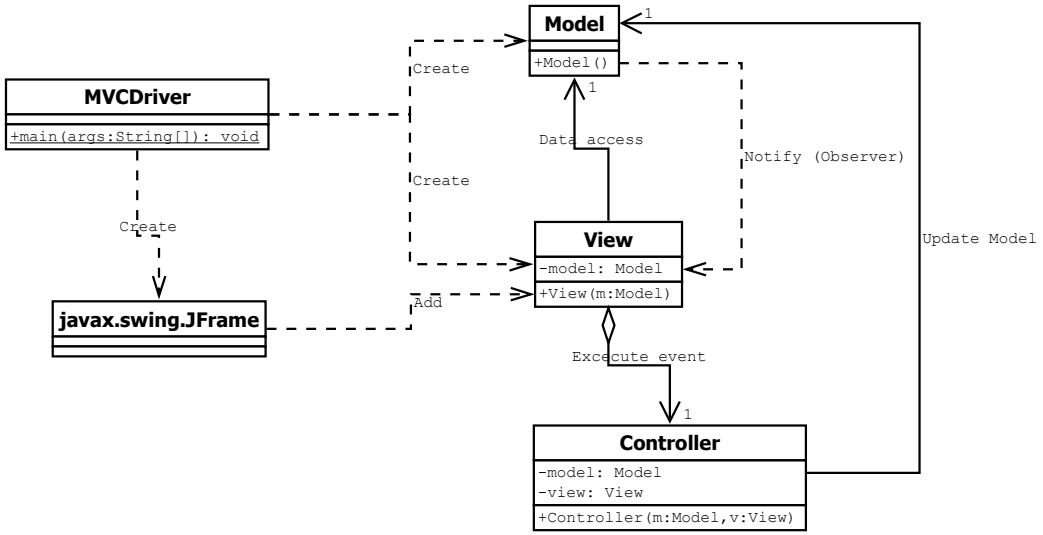
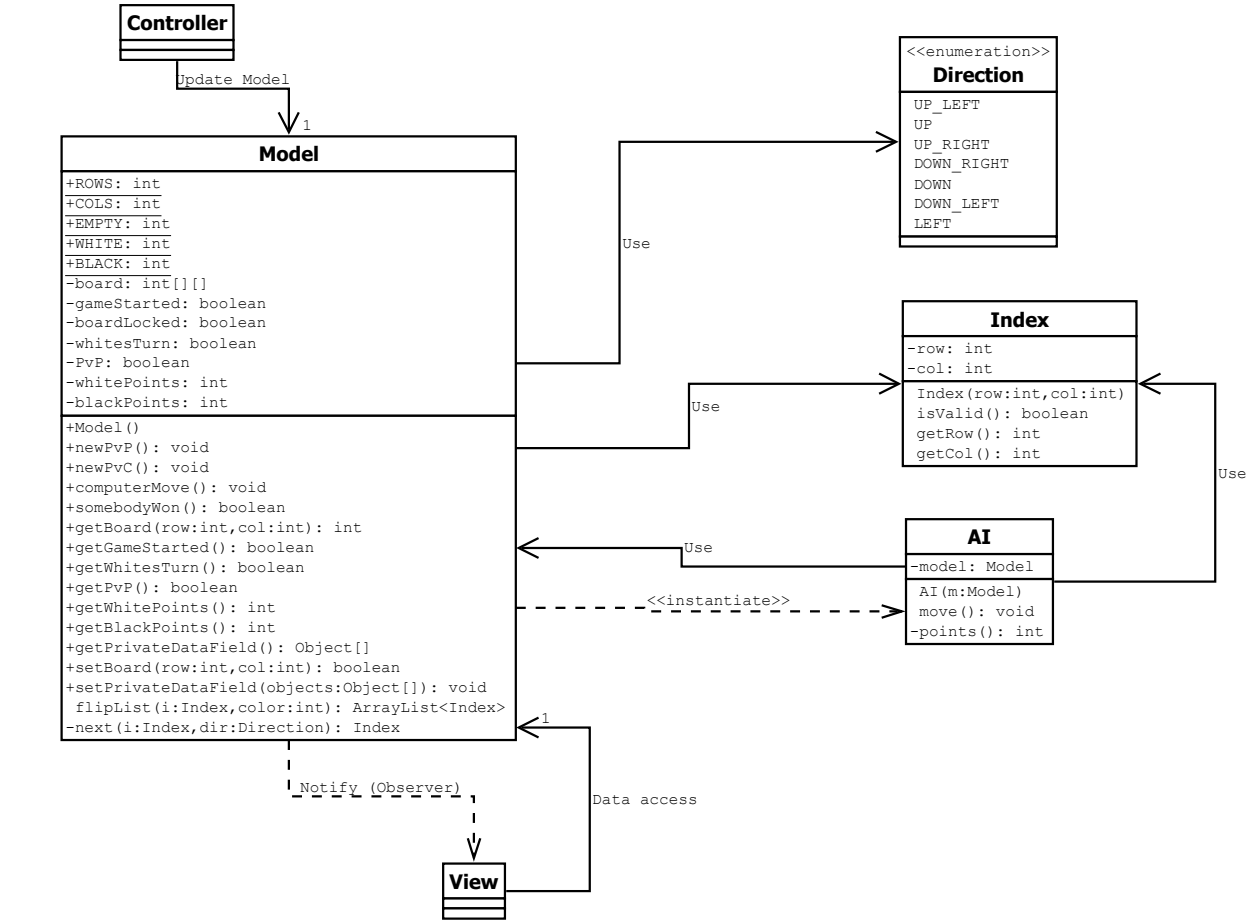
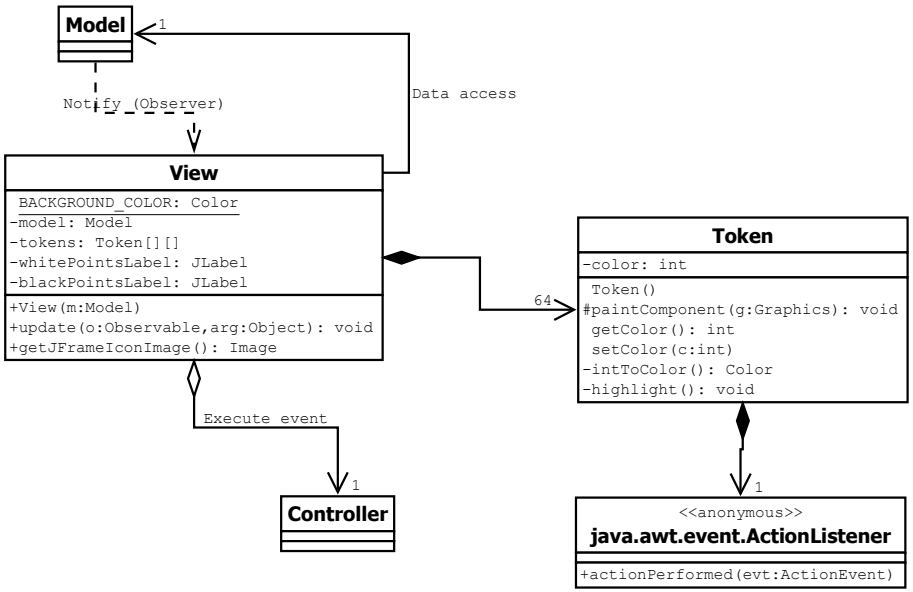
MVCDriver


Model
+Model()
1
1
MVCDriver
+main(args:String[]): void
Create
Create
Create
Data access
Notify (Observer)
Update Model
View
-model: Model
+View(m:Model)
Add
javax.swing.JFrame
Execute event
1
Controller
-model: Model
-view: View
+Controller(m:Model,v:View)

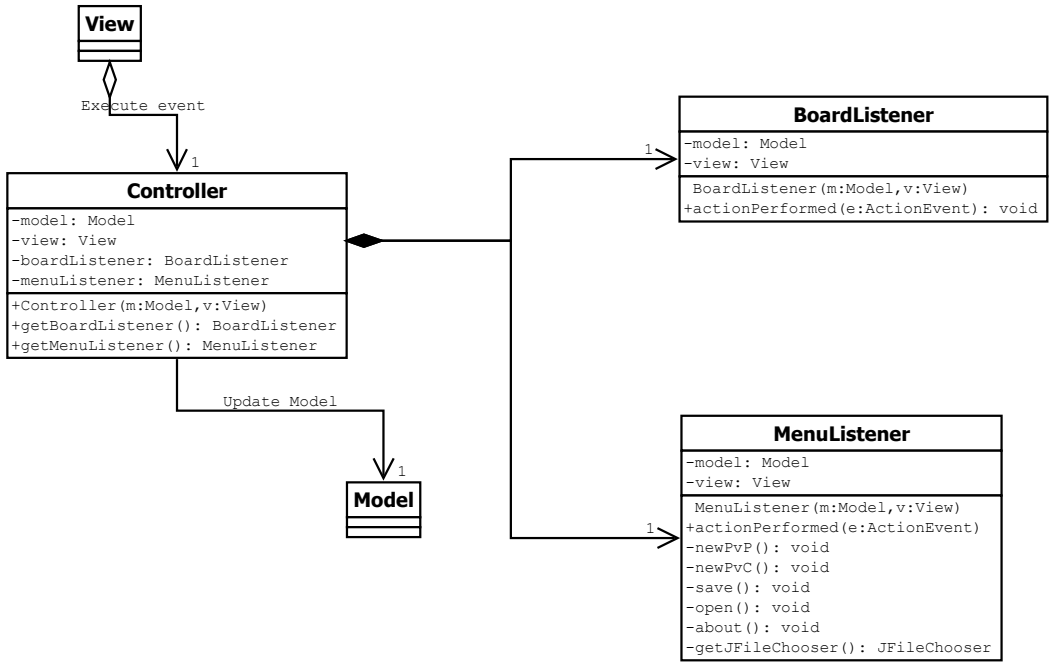
Model


Controller
Update Model
1
Model
+ROWS: int
+COLS: int
+EMPTY: int
+WHITE: int
+BLACK: int
-board: int[][]
-gameStarted: boolean
-boardLocked: boolean
-whitesTurn: boolean
-PvP: boolean
-whitePoints: int
-blackPoints: int
+Model()
+newPvP(): void
+newPvC(): void
+computerMove(): void
+somebodyWon(): boolean
+getBoard(row:int,col:int): int
+getGameStarted(): boolean
+getWhitesTurn(): boolean
+getPvP(): boolean
+getWhitePoints(): int
+getBlackPoints(): int
+getPrivateDataField(): Object[]
+setBoard(row:int,col:int): boolean
+setPrivateDataField(objects:Object[]): void
flipList(i:Index,color:int): ArrayList<Index>
-next(i:Index,dir:Direction): Index
Use
<<enumeration>>
Direction
UP_LEFT
UP
UP_RIGHT
DOWN_RIGHT
DOWN
DOWN_LEFT
LEFT
Use
Index
-row: int
-col: int
Index(row:int,col:int)
isValid(): boolean
getRow(): int
getCol(): int
Use
Use
AI
-model: Model
AI(m:Model)
move(): void
-points(): int
<<instantiate>>
1
Notify (Observer)
Data access
View

View


Model
1
Notify (Observer)
Data access
View
BACKGROUND_COLOR: Color
-model: Model
-tokens: Token[][]
-whitePointsLabel: JLabel
-blackPointsLabel: JLabel
+View(m:Model)
+update(o:Observable,arg:Object): void
+getJFrameIconImage(): Image
64
Token
-color: int
Token()
#paintComponent(g:Graphics): void
getColor(): int
setColor(c:int)
-intToColor(): Color
-highlight(): void
Execute event
1
Controller
1
<<anonymous>>
java.awt.event.ActionListener
+actionPerformed(evt:ActionEvent)

Controller


View
Execute event
1
Controller
-model: Model
-view: View
-boardListener: BoardListener
-menuListener: MenuListener
+Controller(m:Model,v:View)
+getBoardListener(): BoardListener
+getMenuListener(): MenuListener
Update Model
1
Model
BoardListener
-model: Model
-view: View
BoardListener(m:Model,v:View)
+actionPerformed(e:ActionEvent): void
1
MenuListener
-model: Model
-view: View
MenuListener(m:Model,v:View)
+actionPerformed(e:ActionEvent)
-newPvP(): void
-newPvC(): void
-save(): void
-open(): void
-about(): void
-getJFileChooser(): JFileChooser
1