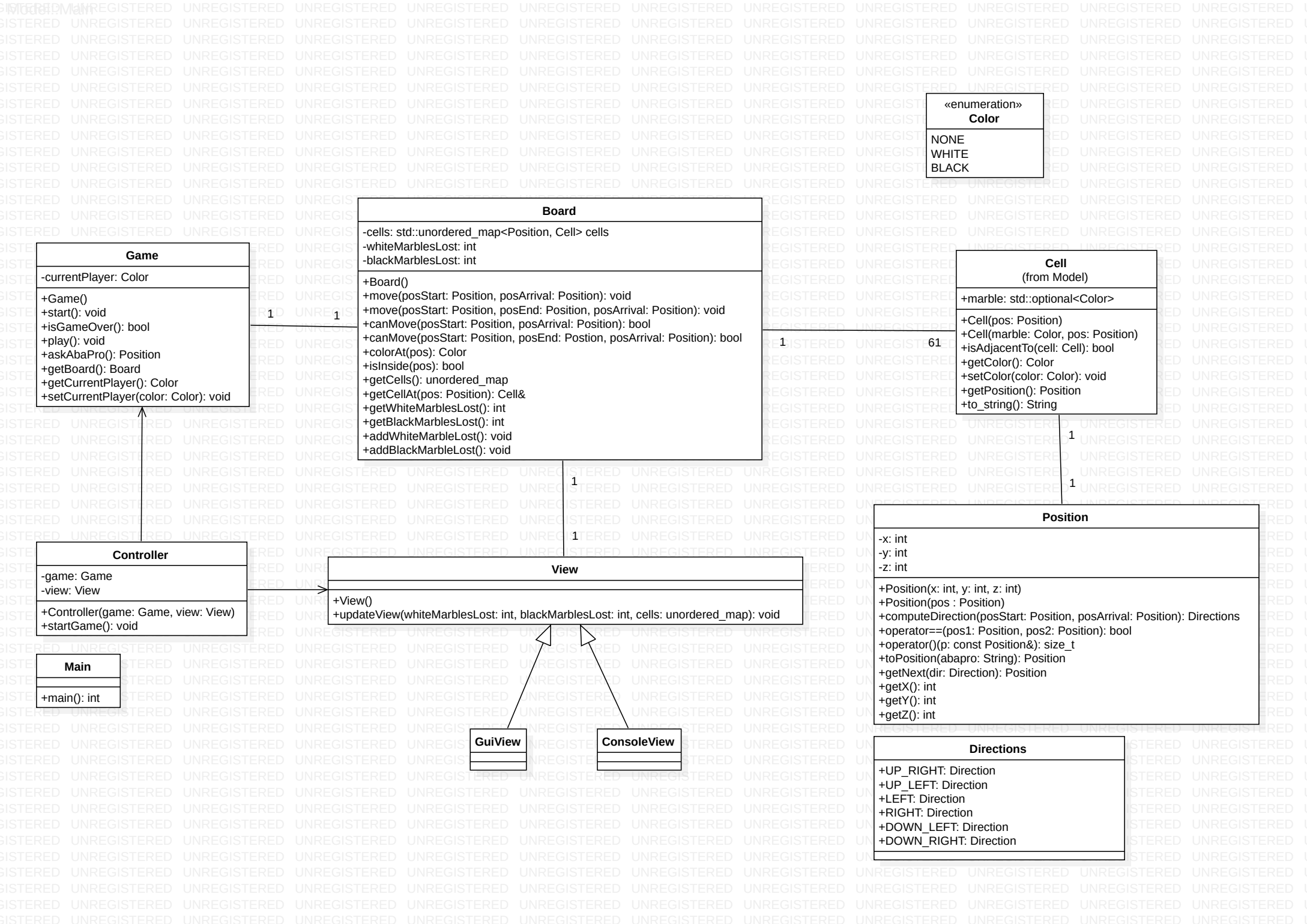
Model.main
«enumeration»
Color
NONE
WHITE
BLACK
Board
-cells: std::unordered_map<Position, Cell> cells
-whiteMarblesLost: int
-blackMarblesLost: int
+Board()
+move(posStart: Position, posArrival: Position): void
+move(posStart: Position, posEnd: Position, posArrival: Position): void
+canMove(posStart: Position, posArrival: Position): bool
+canMove(posStart: Position, posEnd: Postion, posArrival: Position): bool
+colorAt(pos): Color
+isInside(pos): bool
+getCells(): unordered_map
+getCellAt(pos: Position): Cell&
+getWhiteMarblesLost(): int
+getBlackMarblesLost(): int
+addWhiteMarbleLost(): void
+addBlackMarbleLost(): void
Game
-currentPlayer: Color
+Game()
+start(): void
+isGameOver(): bool
+play(): void
+askAbaPro(): Position
+getBoard(): Board
+getCurrentPlayer(): Color
+setCurrentPlayer(color: Color): void
Cell
(from Model)
+marble: std::optional<Color>
+Cell(pos: Position)
+Cell(marble: Color, pos: Position)
+isAdjacentTo(cell: Cell): bool
+getColor(): Color
+setColor(color: Color): void
+getPosition(): Position
+to_string(): String
Controller
-game: Game
-view: View
+Controller(game: Game, view: View)
+startGame(): void
View
+View()
+updateView(whiteMarblesLost: int, blackMarblesLost: int, cells: unordered_map): void
Position
-x: int
-y: int
-z: int
+Position(x: int, y: int, z: int)
+Position(pos : Position)
+computeDirection(posStart: Position, posArrival: Position): Directions
+operator==(pos1: Position, pos2: Position): bool
+operator()(p: const Position&): size_t
+toPosition(abapro: String): Position
+getNext(dir: Direction): Position
+getX(): int
+getY(): int
+getZ(): int
Main
+main(): int
GuiView
ConsoleView
Directions
+UP_RIGHT: Direction
+UP_LEFT: Direction
+LEFT: Direction
+RIGHT: Direction
+DOWN_LEFT: Direction
+DOWN_RIGHT: Direction
1
1
1
61
1
1
1
1