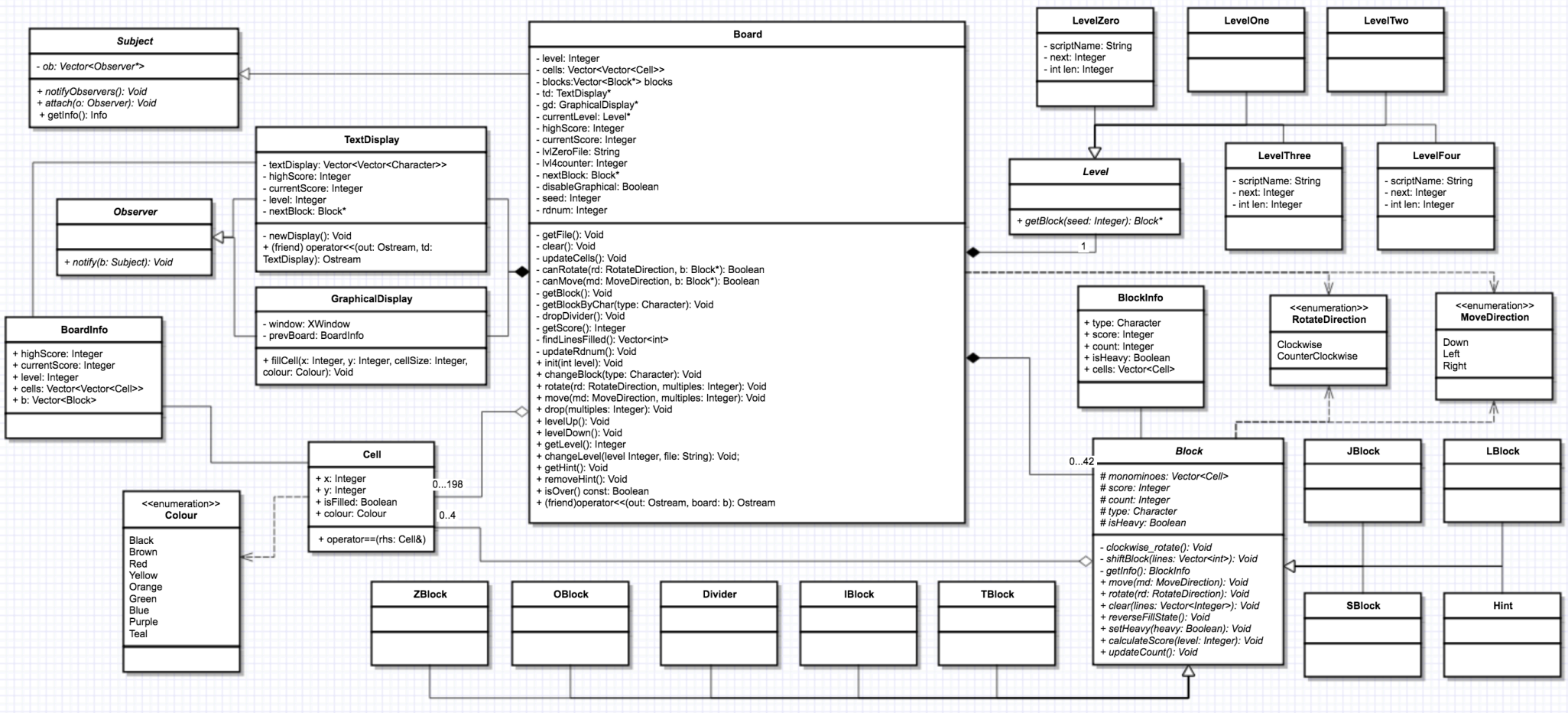
Subject
- ob: Vector<Observer*>
+ notifyObservers(): Void
+ attach(o: Observer): Void
+ getInfo(): Info
Board
- level: Integer
- cells: Vector<Vector<Cell>>
- blocks:Vector<Block*> blocks
- td: TextDisplay*
- gd: GraphicalDisplay*
- currentLevel: Level*
- highScore: Integer
- currentScore: Integer
- lvlZeroFile: String
- lvl4counter: Integer
- nextBlock: Block*
- disableGraphical: Boolean
- seed: Integer
- rdnum: Integer
- getFile(): Void
- clear(): Void
- updateCells(): Void
- canRotate(rd: RotateDirection, b: Block*): Boolean
- canMove(md: MoveDirection, b: Block*): Boolean
- getBlock(): Void
- getBlockByChar(type: Character): Void
- dropDivider(): Void
- getScore(): Integer
- findLinesFilled(): Vector<int>
- updateRdnum(): Void
+ init(int level): Void
+ changeBlock(type: Character): Void
+ rotate(rd: RotateDirection, multiples: Integer): Void
+ move(md: MoveDirection, multiples: Integer): Void
+ drop(multiples: Integer): Void
+ levelUp(): Void
+ levelDown(): Void
+ getLevel(): Integer
+ changeLevel(level Integer, file: String): Void;
+ getHint(): Void
+ removeHint(): Void
+ isOver() const: Boolean
+ (friend)operator<<(out: Ostream, board: b): Ostream
LevelZero
- scriptName: String
- next: Integer
- int len: Integer
LevelOne
LevelTwo
Level
+ getBlock(seed: Integer): Block*
LevelThree
- scriptName: String
- next: Integer
- int len: Integer
LevelFour
- scriptName: String
- next: Integer
- int len: Integer
1
TextDisplay
- textDisplay: Vector<Vector<Character>>
- highScore: Integer
- currentScore: Integer
- level: Integer
- nextBlock: Block*
- newDisplay(): Void
+ (friend) operator<<(out: Ostream, td: TextDisplay): Ostream
Observer
+ notify(b: Subject): Void
GraphicalDisplay
- window: XWindow
- prevBoard: BoardInfo
+ fillCell(x: Integer, y: Integer, cellSize: Integer, colour: Colour): Void
BoardInfo
+ highScore: Integer
+ currentScore: Integer
+ level: Integer
+ cells: Vector<Vector<Cell>>
+ b: Vector<Block>
BlockInfo
+ type: Character
+ score: Integer
+ count: Integer
+ isHeavy: Boolean
+ cells: Vector<Cell>
<<enumeration>>
RotateDirection
Clockwise
CounterClockwise
<<enumeration>>
MoveDirection
Down
Left
Right
Cell
+ x: Integer
+ y: Integer
+ isFilled: Boolean
+ colour: Colour
+ operator==(rhs: Cell&)
0...198
0..4
0...42
<<enumeration>>
Colour
Black
Brown
Red
Yellow
Orange
Green
Blue
Purple
Teal
Block
monominoes: Vector<Cell>
score: Integer
count: Integer
type: Character
isHeavy: Boolean
- clockwise_rotate(): Void
- shiftBlock(lines: Vector<int>): Void
- getInfo(): BlockInfo
+ move(md: MoveDirection): Void
+ rotate(rd: RotateDirection): Void
+ clear(lines: Vector<Integer>): Void
+ reverseFillState(): Void
+ setHeavy(heavy: Boolean): Void
+ calculateScore(level: Integer): Void
+ updateCount(): Void
JBlock
LBlock
SBlock
Hint
ZBlock
OBlock
Divider
IBlock
TBlock