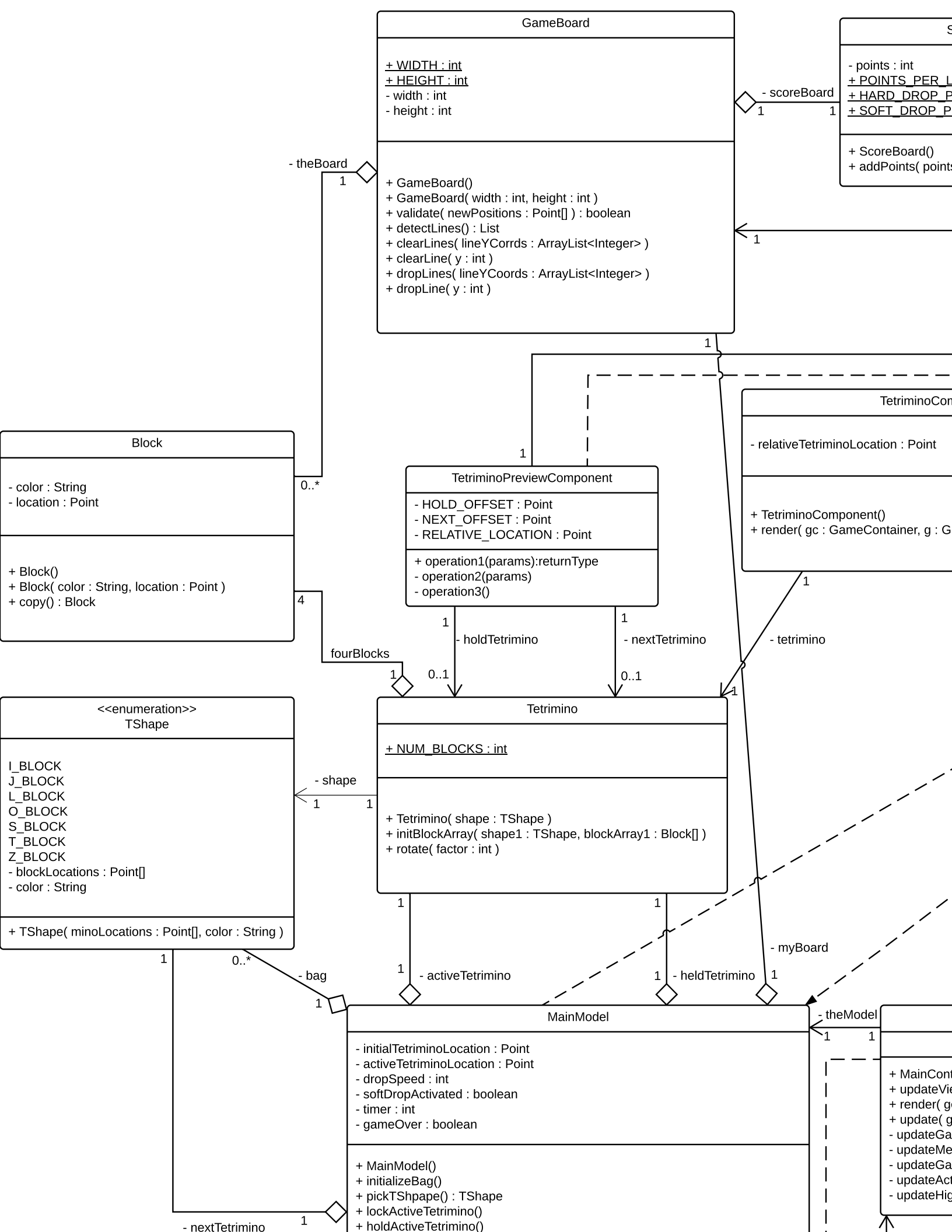
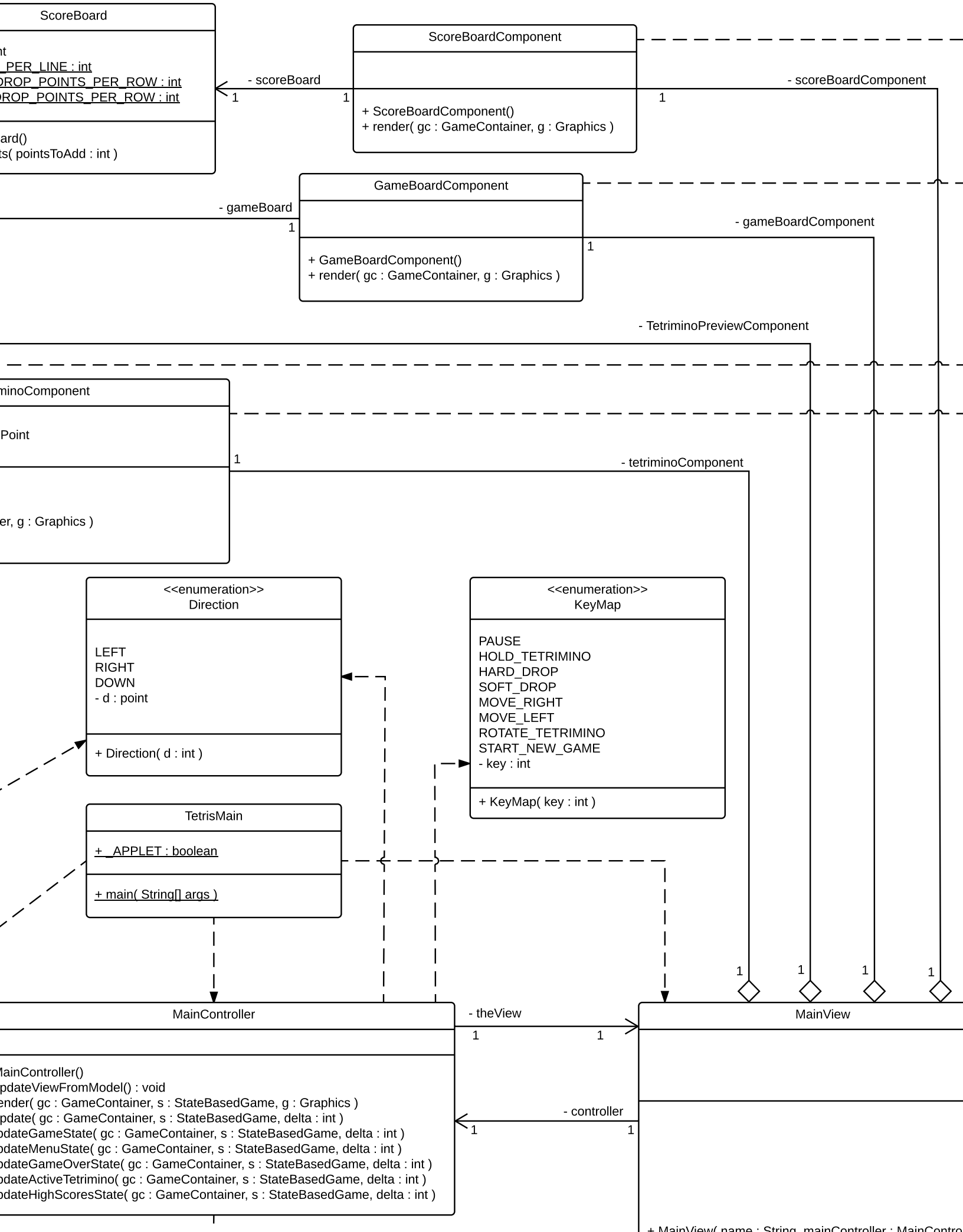
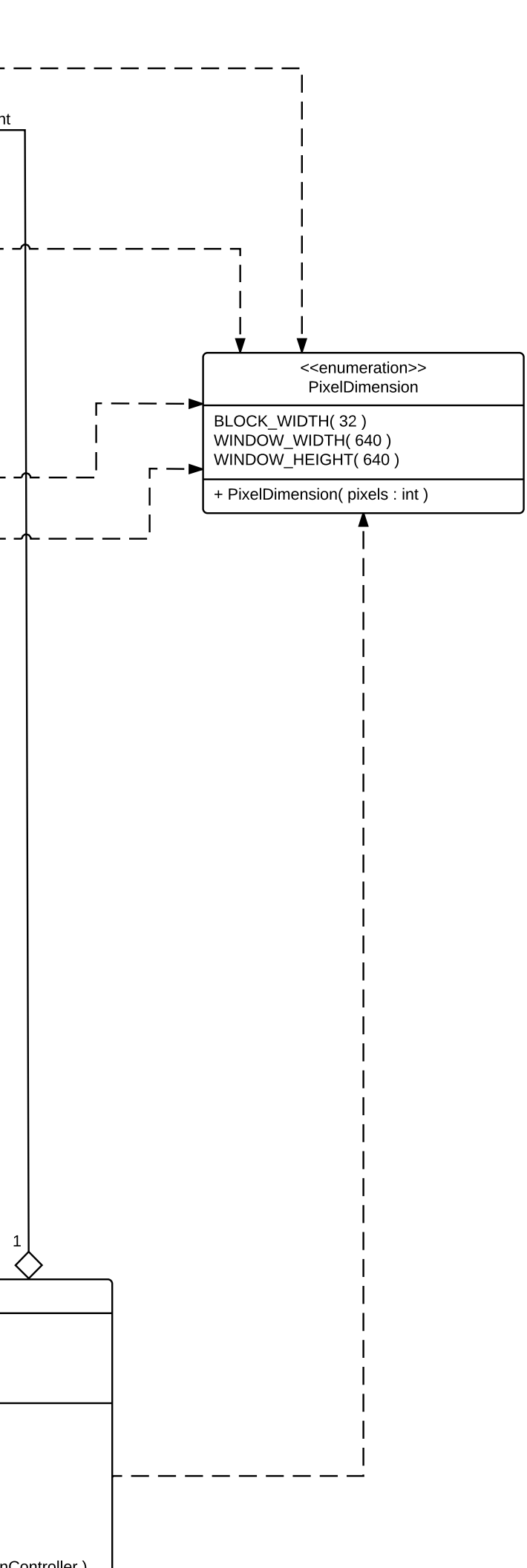
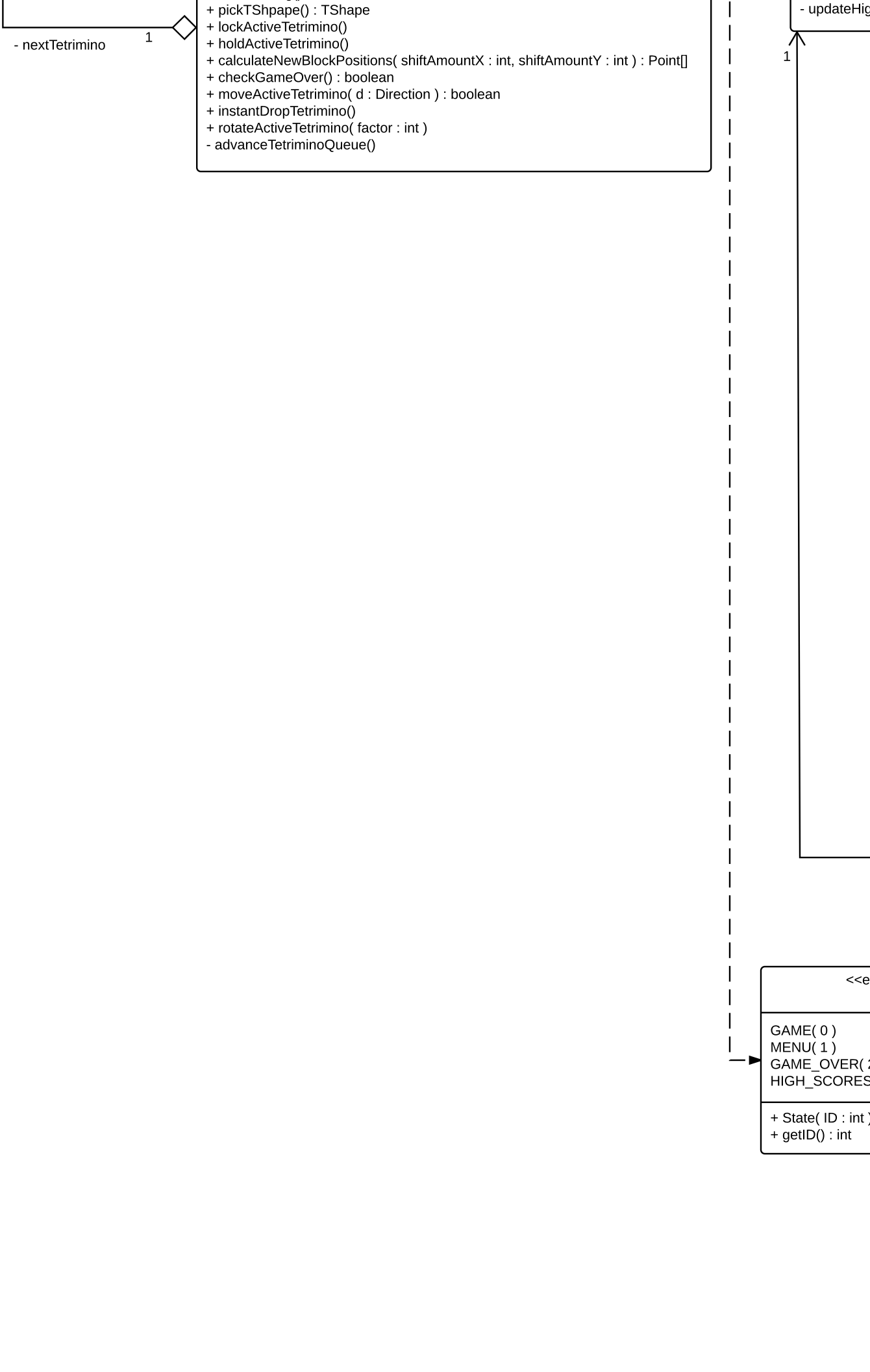
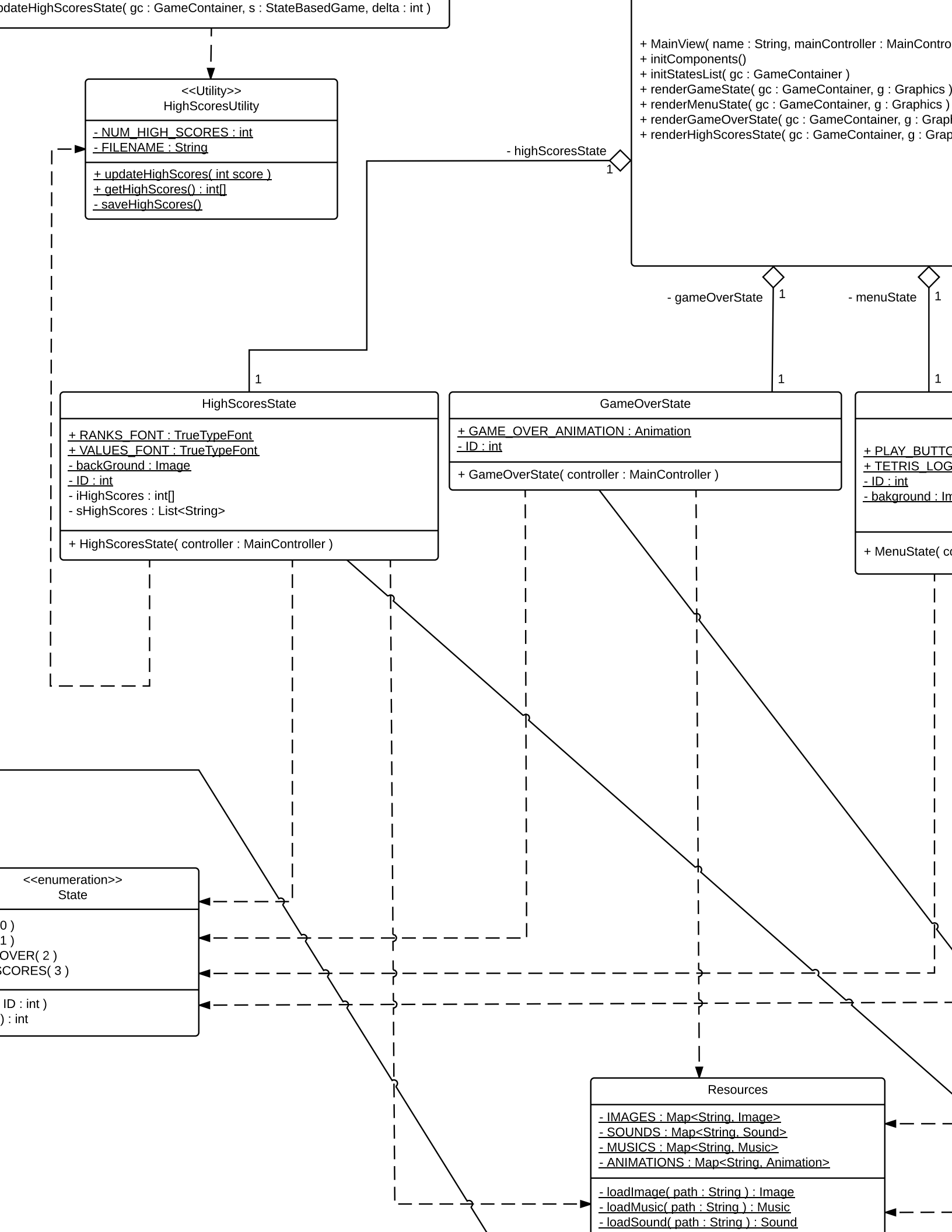
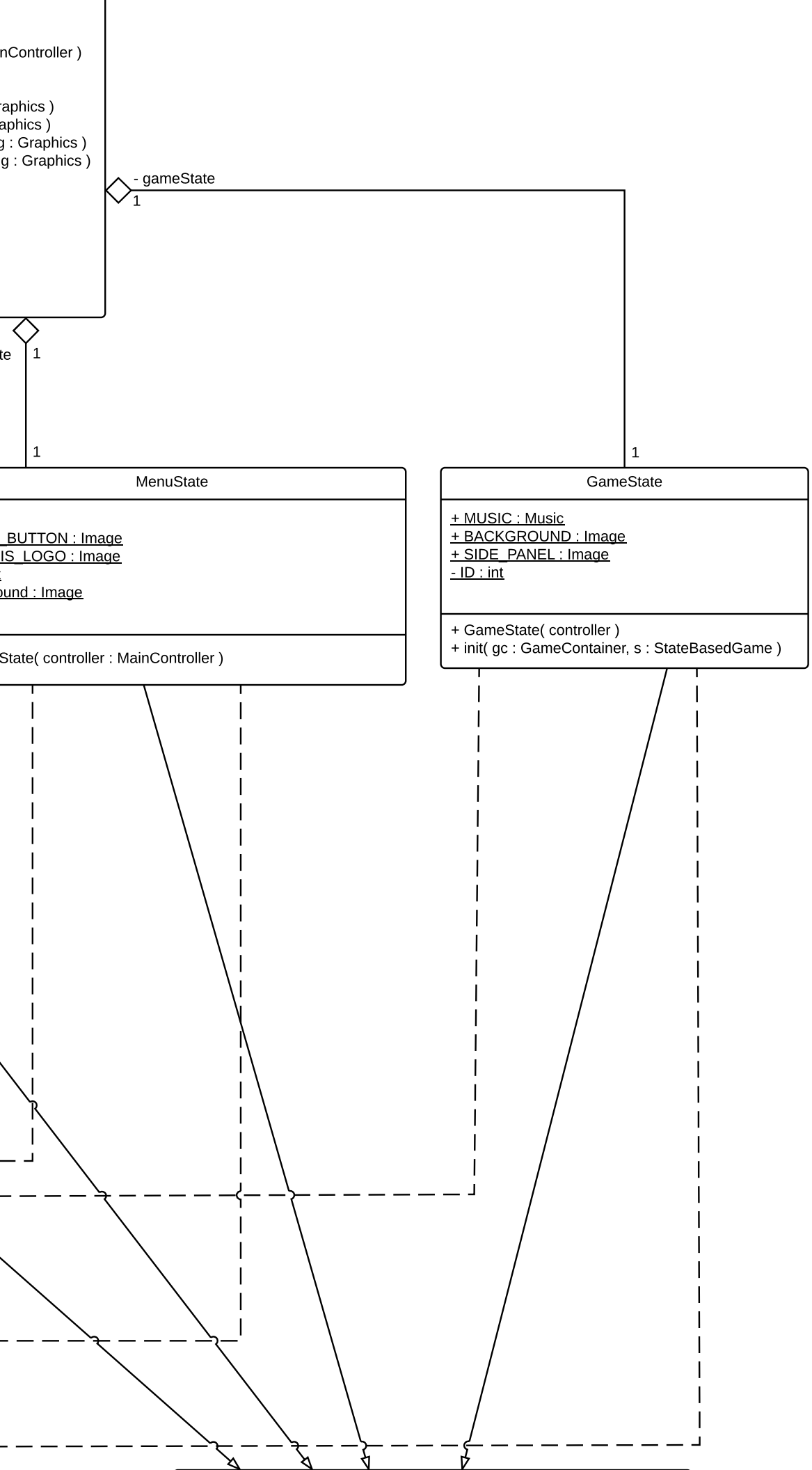

GameBoard
+ WIDTH : int
+ HEIGHT : int
- width : int
- height : int
+ GameBoard()
+ GameBoard(width : int, height : int)
+ validate(newPositions : Point[]) : boolean
+ detectLines() : List
+ clearLines(lineYCorrds : ArrayList<Integer>)
+ clearLine(y : int)
+ dropLines(lineYCoords : ArrayList<Integer>)
+ dropLine(y : int)
- scoreBoard
- theBoard
- points : int
+ POINTS_PER_L
+ HARD_DROP_P
+ SOFT_DROP_P
+ ScoreBoard()
+ addPoints(points
Block
- color : String
- location : Point
+ Block()
+ Block(color : String, location : Point)
+ copy() : Block
TetriminoPreviewComponent
- HOLD_OFFSET : Point
- NEXT_OFFSET : Point
- RELATIVE_LOCATION : Point
+ operation1(params):returnType
- operation2(params)
- operation3()
TetriminoCon
- relativeTetriminoLocation : Point
+ TetriminoComponent()
+ render(gc : GameContainer, g : G
- holdTetrimino
- nextTetrimino
- tetrimino
fourBlocks
<<enumeration>>
TShape
I_BLOCK
J_BLOCK
L_BLOCK
O_BLOCK
S_BLOCK
T_BLOCK
Z_BLOCK
- blockLocations : Point[]
- color : String
+ TShape(minoLocations : Point[], color : String)
Tetrimino
+ NUM_BLOCKS : int
+ Tetrimino(shape : TShape)
+ initBlockArray(shape1 : TShape, blockArray1 : Block[])
+ rotate(factor : int)
- shape
- bag
- activeTetrimino
- heldTetrimino
- myBoard
MainModel
- initialTetriminoLocation : Point
- activeTetriminoLocation : Point
- dropSpeed : int
- softDropActivated : boolean
- timer : int
- gameOver : boolean
+ MainModel()
+ initializeBag()
+ pickTShpape() : TShape
+ lockActiveTetrimino()
+ holdActiveTetrimino()
- theModel
- nextTetrimino

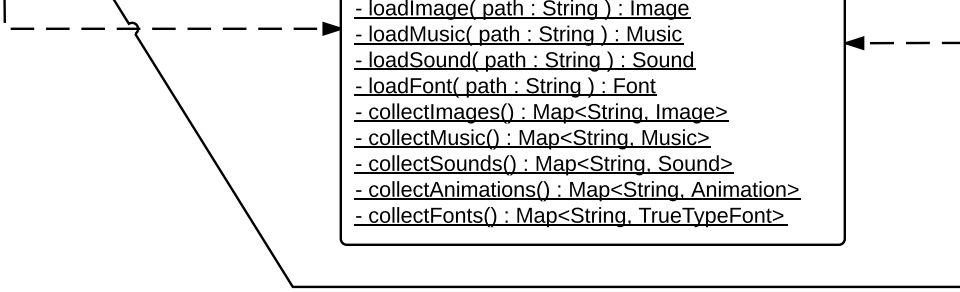

ScoreBoard
nt
_PER_LINE : int
ROP_POINTS_PER_ROW : int
ROP_POINTS_PER_ROW : int
ard()
ts(pointsToAdd : int)
ScoreBoardComponent
+ ScoreBoardComponent()
+ render(gc : GameContainer, g : Graphics)
- scoreBoard
1
1
- scoreBoardComponent
1
GameBoardComponent
- gameBoard
1
+ GameBoardComponent()
+ render(gc : GameContainer, g : Graphics)
- gameBoardComponent
1
- TetriminoPreviewComponent
minoComponent
Point
er, g : Graphics)
1
- tetriminoComponent
<<enumeration>>
Direction
LEFT
RIGHT
DOWN
- d : point
+ Direction(d : int)
<<enumeration>>
KeyMap
PAUSE
HOLD_TETRIMINO
HARD_DROP
SOFT_DROP
MOVE_RIGHT
MOVE_LEFT
ROTATE_TETRIMINO
START_NEW_GAME
- key : int
+ KeyMap(key : int)
TetrisMain
+ _APPLET : boolean
+ main(String[] args)
1
1
1
1
MainController
ainController()
pdateViewFromModel() : void
ender(gc : GameContainer, s : StateBasedGame, g : Graphics)
pdate(gc : GameContainer, s : StateBasedGame, delta : int)
odateGameState(gc : GameContainer, s : StateBasedGame, delta : int)
odateMenuState(gc : GameContainer, s : StateBasedGame, delta : int)
odateGameOverState(gc : GameContainer, s : StateBasedGame, delta : int)
odateActiveTetrimino(gc : GameContainer, s : StateBasedGame, delta : int)
odateHighScoresState(gc : GameContainer, s : StateBasedGame, delta : int)
- theView
1
1
MainView
- controller
1
1
+ MainView(name : String, mainController : MainContro

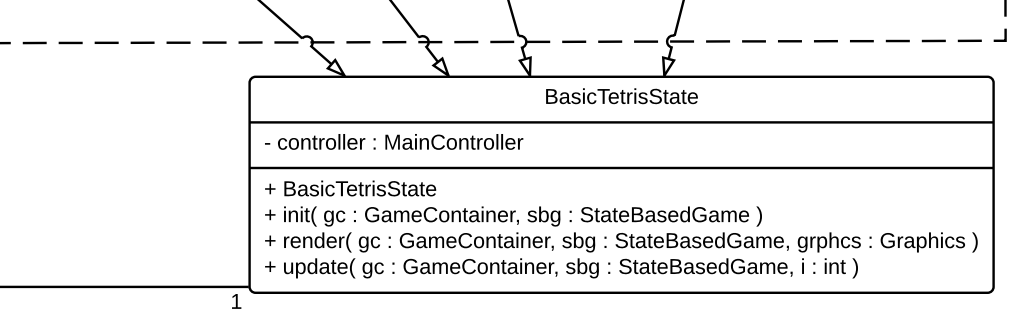

nt
<<enumeration>>
PixelDimension
BLOCK_WIDTH(32)
WINDOW_WIDTH(640)
WINDOW_HEIGHT(640)
+ PixelDimension(pixels : int)
1
nController)

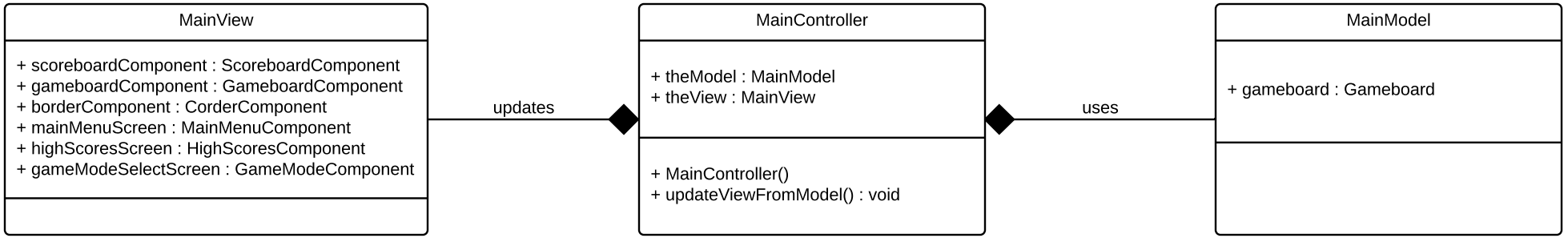

- nextTetrimino
1
+ pickTShpape() : TShape
+ lockActiveTetrimino()
+ holdActiveTetrimino()
+ calculateNewBlockPositions(shiftAmountX : int, shiftAmountY : int) : Point[]
+ checkGameOver() : boolean
+ moveActiveTetrimino(d : Direction) : boolean
+ instantDropTetrimino()
+ rotateActiveTetrimino(factor : int)
- advanceTetriminoQueue()
- updateHig
1
<<e
GAME(0)
MENU(1)
GAME_OVER(2
HIGH_SCORES
+ State(ID : int)
+ getID() : int


pdateHighScoresState(gc : GameContainer, s : StateBasedGame, delta : int)
+ MainView(name : String, mainController : MainContro
+ initComponents()
+ initStatesList(gc : GameContainer)
+ renderGameState(gc : GameContainer, g : Graphics)
+ renderMenuState(gc : GameContainer, g : Graphics)
+ renderGameOverState(gc : GameContainer, g : Grapl
+ renderHighScoresState(gc : GameContainer, g : Grap
<<Utility>>
HighScoresUtility
- NUM_HIGH_SCORES : int
- FILENAME : String
+ updateHighScores(int score)
+ getHighScores() : int[]
- saveHighScores()
- highScoresState
1
- gameOverState
1
- menuState
1
1
1
1
HighScoresState
+ RANKS_FONT : TrueTypeFont
+ VALUES_FONT : TrueTypeFont
- backGround : Image
- ID : int
- iHighScores : int[]
- sHighScores : List<String>
+ HighScoresState(controller : MainController)
GameOverState
+ GAME_OVER_ANIMATION : Animation
- ID : int
+ GameOverState(controller : MainController)
+ PLAY_BUTTO
+ TETRIS_LOG
- ID : int
- bakground : Im
+ MenuState(co
<<enumeration>>
State
0)
1)
OVER(2)
SCORES(3)
ID : int)
) : int
Resources
- IMAGES : Map<String, Image>
- SOUNDS : Map<String, Sound>
- MUSICS : Map<String, Music>
- ANIMATIONS : Map<String, Animation>
- loadImage(path : String) : Image
- loadMusic(path : String) : Music
- loadSound(path : String) : Sound


nController)
raphics)
aphics)
g : Graphics)
g : Graphics)
- gameState
1
te
1
1
1
MenuState
_BUTTON : Image
IS_LOGO : Image
ound : Image
State(controller : MainController)
GameState
+ MUSIC : Music
+ BACKGROUND : Image
+ SIDE_PANEL : Image
- ID : int
+ GameState(controller)
+ init(gc : GameContainer, s : StateBasedGame)


- loadImage(path : String) : Image
- loadMusic(path : String) : Music
- loadSound(path : String) : Sound
- loadFont(path : String) : Font
- collectImages() : Map<String, Image>
- collectMusic() : Map<String, Music>
- collectSounds() : Map<String, Sound>
- collectAnimations() : Map<String, Animation>
- collectFonts() : Map<String, TrueTypeFont>


BasicTetrisState
- controller : MainController
+ BasicTetrisState
+ init(gc : GameContainer, sbg : StateBasedGame)
+ render(gc : GameContainer, sbg : StateBasedGame, grphcs : Graphics)
+ update(gc : GameContainer, sbg : StateBasedGame, i : int)
1


MainView
+ scoreboardComponent : ScoreboardComponent
+ gameboardComponent : GameboardComponent
+ borderComponent : CorderComponent
+ mainMenuScreen : MainMenuComponent
+ highScoresScreen : HighScoresComponent
+ gameModeSelectScreen : GameModeComponent
updates
MainController
+ theModel : MainModel
+ theView : MainView
+ MainController()
+ updateViewFromModel() : void
uses
MainModel
+ gameboard : Gameboard

MainModel	
Responsibilities	Collaborators
• Initialize bag • Pick next Tetrimino • Lock active Tetrimino • Hold active Tetrimino • Move active Tetrimino • Instant-drop active Tetrimino • Rotate active Tetrimino	• GameBoard • Tetrimino • TShape • Direction

TetrisMain	
Responsibilities	Collaborators
• Launch Game	• MainModel • MainView • MainController

MainView	
Responsibilities	Collaborators
• Render game screen • Render next Tetrimino preview • Render held Tetrimino preview • Render high scores screen • Render menu screen	• MainController • GameBoardComponent • TetriminoPreviewComponent • TetriminoComponent • ScoreBoardComponent • BasicTetrisState • PixelDimension

MainController	
Responsibilities	Collaborators
• Interperet user input • Distribute control • Manage game clock • Update BasicTetrisStates • Pass render responsibilities back to model	• MainModel • MainView • BasicTetrisState • State • KeyMap • Direction

BasicTetrisState	
Responsibilities	Collaborators
• Pass rendering responsibility to MainController • Contain necessary graphics/sounds	• MainController

Tetrimino	
Responsibilities	Collaborators
• Contain 4 Blocks in a specific shape	• Block • TShape

TetriminoComponent	
Responsibilities	Collaborators
• Render a Tetrimino	• Tetrimino

TetriminoPreviewComponent	
Responsibilities	Collaborators
• Render a Tetrimino preview	• Tetrimino

ScoreBoardComponent	
Responsibilities	Collaborators
• Render a ScoreBoard	• ScoreBoard

TetriminoComponent	
Responsibilities	Collaborators
• Render a Tetrimino	• Tetrimino