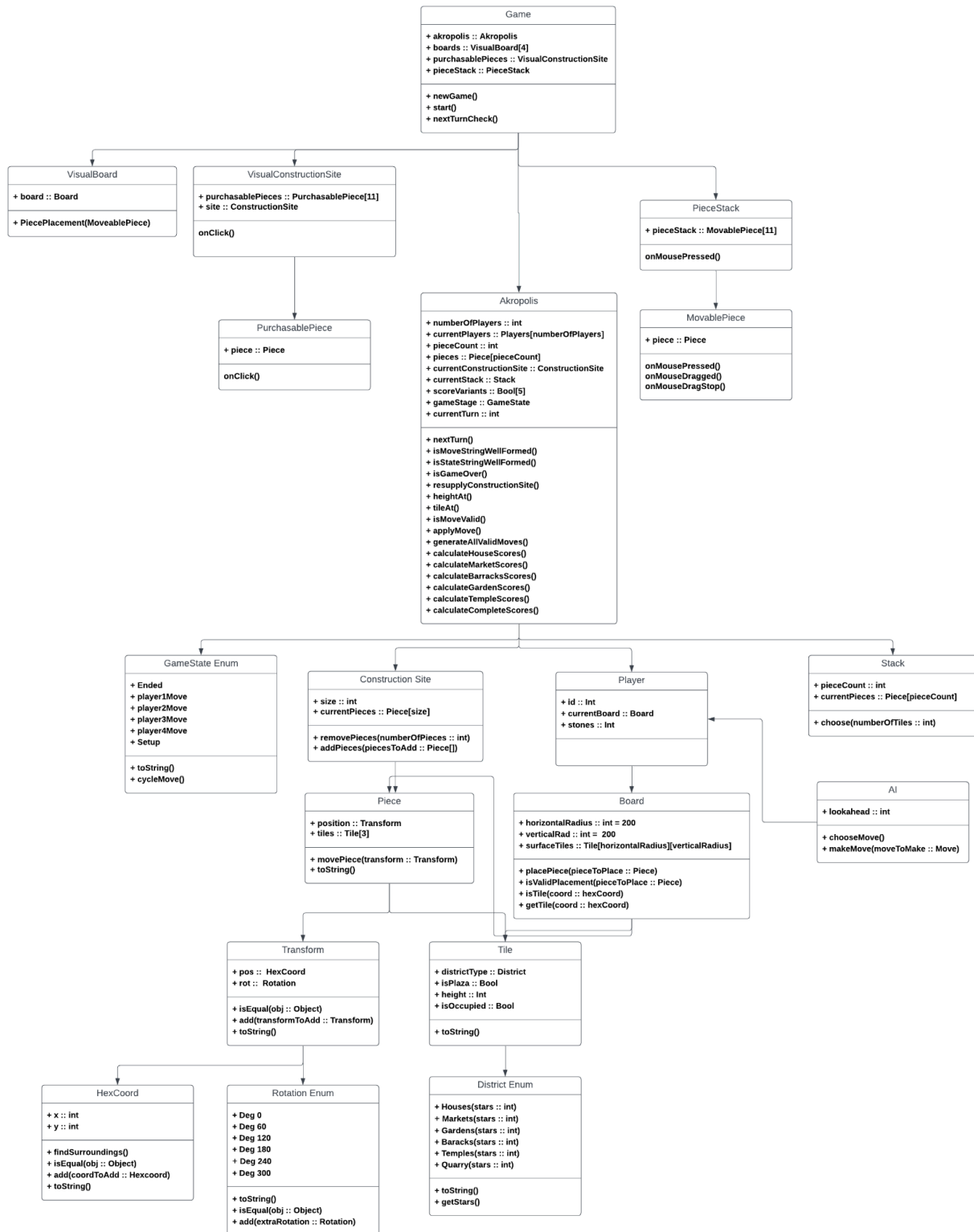
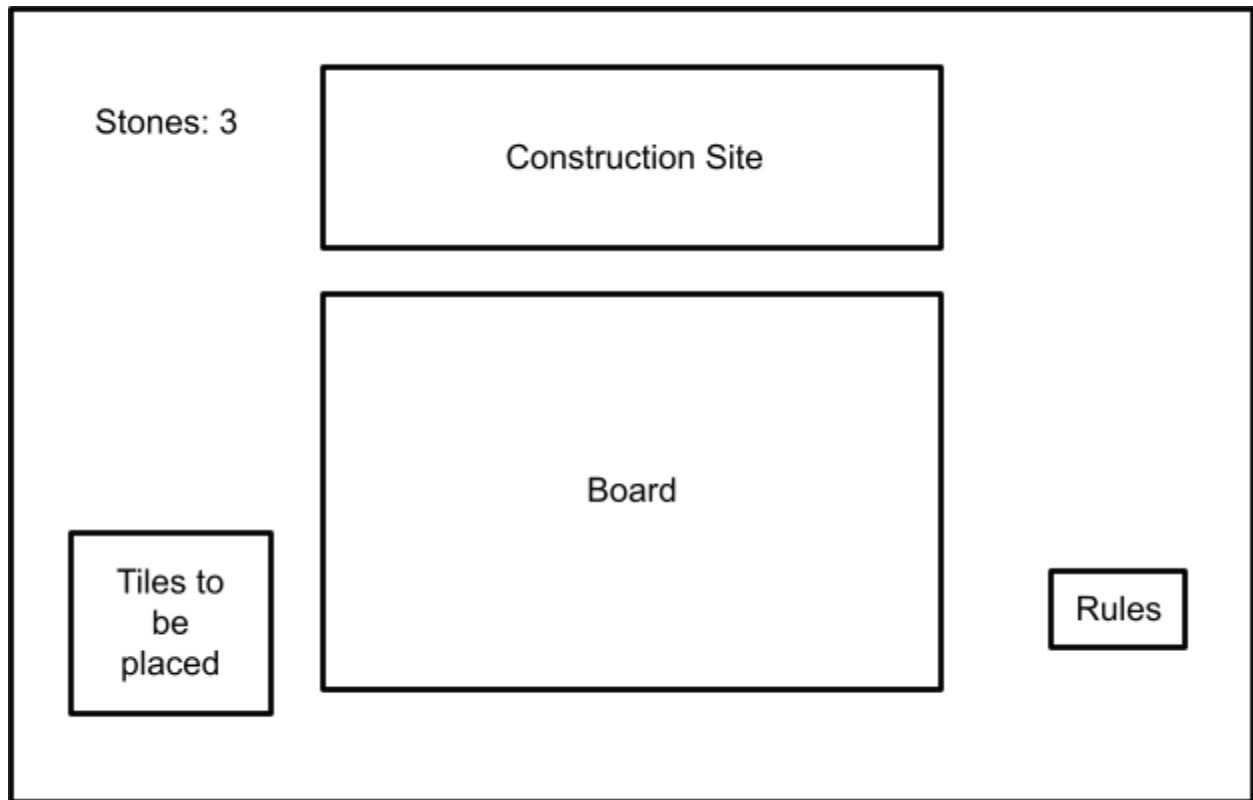


# UML Class Diagram



## GUI Sketch



## List of Classes

### Classes:

- **Akropolis**
  - **Fields:**
    - `numberOfPlayers :: int`
    - `currentPlayers :: Players[numberOfPlayers]`
    - `pieceCount :: int`
    - `pieces :: Piece[pieceCount]`
    - `currentConstructionSite :: ConstructionSite`
    - `currentStack :: stack`
    - `scoreVariants :: Bool[5]`
    - `response :: Input`
    - `gameStage :: gameState`
    - `currentTurn :: int`
    - `leader :: int`
  - **Methods:**
    - `constructor(String)`
      - `players(String)`
      - `pieces(String)`

- **stack(String)**
    - **costructionSite(String)** (Maybe make construction site a property of Stack?)
  - **nextTurn()**
  - **updateScore()**
  - **updateLeader()**
  - **getWinner()**
  - **isMoveStringWellFormed()**
  - **isStateStringWellFormed()**
  - **isGameOver()**
  - **resupplyConstructionSite()**
  - **heightAt()**
  - **tileAt()**
  - **isMoveValid()**
  - **applyMove()**
    - **previewMove()**
      - **purchase()**
      - **place()**
  - **generateAllValidMoves()**
  - **calculateHouseScores()**
  - **calculateMarketScores()**
  - **calculateBarracksScores()**
  - **calculateGardenScores()**
  - **calculateTempleScores()**
  - **calculateCompleteScores()**
  - **refillConstructionSite()**
  - **getters**
- **Stack:**
    - **Fields:**
      - **pieceCount :: int**
      - **currentPieces :: Piece[pieceCount]**
    - **Method:**
      - **constructor(String)**
      - **reorder/choose()**
      - **getters**
  - **ConstructionSite:**
    - **Fields:**
      - **size :: int**
      - **currentPieces :: Piece[Size]**
    - **Methods:**
      - **constructor(String)**
      - **deliverPiece(MetaPosition)**
      - **getters**

- **Piece:**
  - **Fields:**
    - `position :: Transform`
    - `districts :: Tile[3]`
  - **Methods**
    - `constructor(String)`
    - `transform(Transform)`
    - `move(metaPosition)`
    - `Getter for every field`
    - `toString()`
- **Tile:**
  - **Fields:**
    - `districtType :: district`
    - `isPlaza :: Bool`
    - `height :: int`
    - `isOccupied :: Bool`
  - **Methods:**
    - `toString()`
- **Board**
  - **Field:**
    - `horizontalRadius :: int 200`
    - `verticalRad :: int 200`
    - `surfaceTiles :: Tile[horizontalRadius][verticalRadius]`
  - **Methods:**
    - `updateBoard(Piece)`
    - `isValidPlacement(Piece)`
    - `isTile(hexCoord)`
    - `getTile(hexCoord)`
    - `getters`
- **Player**
  - **Fields:**
    - `Id :: Int`
    - `currentBoard :: Board`
    - `stones :: Int`
    - `firstPurchaseOfTurn :: bool`
  - **Methods:**
    - `constructor(String)`
      - `board()`
- **AI (last task) inheritance from Player**

- **Fields:**
  - difficulty/lookahead :: int
- **Methods:**
  - chooseMove()
  - makeMove()
- **Transform**
  - **Field:**
    - pos :: HexCoord
    - rot :: Rotation
  - **Method**
    - isEqual(obj)
    - add(Transform)
    - toString()
    - getters
- **HexCoord**
  - **Fields:**
    - Hor :: int (x)
    - Ver :: int (y)
  - **Method**
    - findSurroundings()
    - isEqual(obj)
    - add(Hexcoord)
    - toString()
- **Rotation (enum)**
  - **Field:**
    - Deg 0
    - Deg 60
    - Deg 120
    - Deg 180
    - Deg 240
    - Deg 300
  - **Method**
    - toString
    - isEqual(obj)
    - Constructor (int)
    - add(Rotation)
    - getAngle (to account for angle additions > 360)
    - getRad (because Math package assumes values are in radians)
- **district (enum):**

- **Fields:**
  - Houses
  - Markets
  - Gardens
  - Baracks
  - Temples
  - Quarry
- **Methods:**
  - constructor(string)
  - toString()
  - getStars()
- **gameState (enum):**
  - **Fields:**
    - Over
    - player1Move
    - player2Move
    - player3Move
    - player4Move
    - Beginning
  - **Methods:**
    - toString()
    - cycleMove()