**Module Decomposition**

**5.1 Hardware Hiding Modules (M1)**

**Secrets:** the primary secret hidden by this module is the hardware/software interfaces. Also the data structures and algorithms used to implement the virtual hardware

**Service:** controls the procedures that are used to handle the hardware/software interface changes with the same general capabilities. This is achieved by using virtual hardware, a digital hardware/software connection it can rely on.

**Implemented By:** OS

**5.2 Main Module (M2)**

**Secrets:** Hides everything from the game mechanics to the sensitive design decisions.

**Service:** Is used to execute the game, called by the user or the front-end implementation when a GUI is developed.

**Implemented By:**  -

**5.3 userInput Module (M3)**

**Secrets:** Hides all the software decision modules.

**Service:** serves as the only gateway in the behaviour-hiding module to use the software decision modules, reinforcing encapsulation and information hiding.

**Implemented By:**  Main Module

**5.4 Board Module (M4)**

**Secrets:** Hides the Tile object from the rest of the game logic.

**Service:** creates a six index by six index two-dimensional array, and populates the fields with the Tile object.

**Implemented By:**  Main Module

**5.5 Tile Module (M5)**

**Secrets:** generated randomly

**Service:** creates a tile object with a random selection made from a set array of colors.

**Implemented By:**  Board Module

**5.6 colorMatch Module (M6)**

**Secrets:** hides part of the data – Ignores the index location

**Service:** Given a set of Tile objects, the module is responsible for checking the colors of the tile and confirm if they are all same.

**Implemented By:**  userInput Module

**5.7 adjacent Module (M7)**

**Secrets:** hides part of the data – Ignores the index value

**Service:** Given a set of Tile objects, the module is responsible for checking if the tiles are all adjacent to each other – horizontal and/or vertical (not diagonal).

**Implemented By:**  userInput Module

**5.8 removeTile Module (M8)**

**Secrets:** secures the rest of the board tiles to avoid any loss of data

**Service:** Isolates the selected tiles from the remaining pieces to keep track and update the states of the game board. Finally marks the removed tile with a flag.

**Implemented By:**  userInput Module

**5.9 checkColumn Module (M9)**

**Secrets:** secures the order of the rest of the columns on the board to avoid any loss of data

**Service:** Isolates the selected columns and counts the number of flagged tiles and removes them.

**Implemented By:**  removeTile Module

**5.10 moveDown Module (M10)**

**Secrets:** works only with the columns being updated, hiding the rest of the board for being changed

**Service:** with the given column number and the number of missing tiles, moves all the tiles down. Implements the law of gravity within the board.

**Implemented By:**  checkColumn Module

**5.11 addTile Module (M11)**

**Secrets:** accesses only the very top row of the matrix, hiding the rest of the board from being changed

**Service:** with the given column number and the number of missing tiles, addTile fills the missing fields with new Tile objects.

**Implemented By:**  moveDown Module