# **Classes:**

## GameManager:

### Overview:

Start point of game, holds prefabs info, init method and open game field state (number of tiles on open play area and amount of each tile type etc)

### Awake() function:

Checks for rescaled tile size based on user’s phone resolution (canvas scales with screen size causing tile to have different dimensions, need to adjust for overlap calculations), binds singleton.

### Init() function:

Algo for generation of the game map and tiles is done here.   
  
Does a random for sets (each set = 3 tile of same color) of tiles to generate between editor defined minSet, maxSet then generates multiple layers starting from lowest layer and populating each layer with tiles (layers of size below 3x3 is fully filled, larger sizes have random fill amount between 50%-100%). As layers get added, tiles added to the level will check for overlap with tiles on lower levels and deactivate button interaction if partially covered or deactivate the entire game object if fully covered. Once all required tiles are generated, randomly choose the number of sets of each tile type/color and assign each tile a random color (shuffle function).

### AttemptInsertTile(tile) function:

Function for slotting interactable tiles to player hand.   
  
Will call the tile to uncover tiles it is currently covering and make interactable/reactivate the game object depending on whether the covered object is partially/fully uncovered. Following which, check for whether a set is formed in the player’s hand and destroy the set if yes after which checks for whether the player hits victory/game over condition.

### Shuffle() function:

Randomly assigns an available tile type to each tile in the open play field.   
  
Makes use of typesCount dictionary copy to ensure the amount of types shuffled is correct.  
  
  
Layer:

### Overview:

Class holding a NxN square grid space for tile placement. Spawns number of tiles as defined by game manager call and ensures position of tiles is aligned by ½ tile width and/or ½ tile height compared to layer right below.

Tile:

### Overview:

Holds information for each individual tile.   
  
Attributes includes tile type, tiles covered, quadrantsCoveredByTile (each tile is defined by 4 quadrants, will store all tiles covering each quadrant in a dictionary)

### CheckAndAddCoveredQuadrants(otherTile) function:

Compares own tile against otherTile to see which of its own quadrants is covered by that tile.

### RemoveSelf() function:

Called when tile is removed from the open play area, will go to all the tiles it is covering and uncover the quadrants covered, reactivating/making interactable the other tiles depending on whether the other tile is fully uncovered.

Utils:

### Overview:

Holds overlap calculation function and tile size, also holds tiletype to image mapping (currently just a type to color mapping instead).   
  
Overlapping () :

Checks whether 2 rectangles are overlapping using the min and max x/y coordinates of both rectangles.

Hand:

### Overview:

Holds info about tiles in player’s hand and algorithm for position to insert new tiles, also where undo/store skill calculation is done.

Hand tiles are stored in list to maintain ordering and count of each type is maintained in dictionary holding count of all 12 types.

### InsertTile(tile):

Checks whether count of tile type is more than 0, if not just insert to end of hand and increment count. Otherwise search for first tile of same type in hand and then set new tile sibling index to that tile’s position + number of tile type. If 3 of same tile is found, send list of 3 tiles back to game manager for destruction. Also stores information about where tile was drawn from for undo function.  
  
Store():

Checks if user has at least 1 tile and triggers store skill if yes, will take up to first 3 tiles available in hand and throw into storage area, stored tiles placement and overlaps done in Storage class.

### Undo():

Checks if user’s last move completes a set, if not proceed with undo and places last drawn tile to previous parent and position and covers any tiles it previously uncovered.

Storage:

### Overview:

Holds information about storage slots and tiles held in storage.

### Store(List<tile>):

Places given list of tiles into the 3 storage slots according to list order.

Skills:

### Overview:

Holds information about skills used and calls skills if the user is allowed to use them or display relevant error message if skill condition is not met.

### UseSkill(skillType):

Checks if user has used skillType and if not, attempts to use skill, triggers error message if condition not met, uses skill otherwise and disable skill button.

Overlay:

### Overview:

Holds overlay game object for when user pauses/loses or wins. Changes buttons and text shown depending on how overlay was triggered.

Enums:

### Overview:

Holds enum information: overlay types, tile types, quadrants (for tile calculation: topleft, top right, bottom right, bottom left), skill list