**In JSFGameManager.cs**

void Awake () { // board needs to be initialized before other scripts can access it

JSFUtils.gm = this; // make a easy reference to the GameManager ( this script ! )

JSFUtils.wc = GetComponent<JSFWinningConditions>(); // make a easy reference to the WinningConditions script~!

JSFUtils.vm = GetComponent<JSFVisualManager>(); // make a easy reference to the GUIManager script~!

JSFRelay.onPreGameStart();

initializeGame();

preGameSetup();

// init the board objects

foreach(JSFBoard \_board in board){

\_board.init(); // to show the GUIs for the objects

}

canMove = false; // initially cannot be moved...

gameState = JSFGameState.GamePending; // game is waiting to be started...

}

void Start(){

// init the board objects

foreach(JSFBoard \_board in board){

\_board.init(); // to show the GUIs for the objects

}

}

**In JSFUtils.cs**

public static void creatSwipeLine(GameObject obj, Vector3 a, Vector3 b, float zValue){

obj.transform.localScale = new Vector3 (obj.transform.localScale.x, 1, 1 ); // resets the scale first

// auto scaling feature

Bounds bounds = findObjectBounds(obj);

float val = (Vector3.Distance(a,b)) / // get the bigger size to keep ratio

Mathf.Clamp( Mathf.Max(bounds.size.x,bounds.size.y),0.0000001f,float.MaxValue);

obj.transform.localScale = new Vector3 (obj.transform.localScale.x, val, 1 ); // the final scale value

obj.transform.position = Vector3.Lerp(a,b,0.5f) + new Vector3(0,0,zValue); // reposition to the middle

obj.transform.LookAt(b,Vector3.forward); // auto adjust the rotation to look at the target

// adjust the box collider if present...

BoxCollider bc = obj.GetComponent<BoxCollider>();

if ( bc != null){

float maxSize = Mathf.Max( new float[] {bounds.size.x,bounds.size.y,bounds.size.z} );

bc.size = new Vector3(maxSize, maxSize, bounds.size.z + 0.01f);

bc.center = Vector3.zero;

}

}