DESCRIZIONE SCRIPT

Linee guida:

**Variabili:** Tutte le variabili iniziano con una lettera minuscola, non ci sono spazi all`interno dei nomi ma le parole vengono separate dall`iniziale maiuscola (tranne la prima), il nome deve essere il più descrittivo possibile (tanto è presente l`auto compilazione). Nessuna variabile è mai dichiarata pubblica per evitare conflitti di nomi nel caso si dovesse utilizzare la classe all`interno di altre classi (Per la modifica da Inspector viene usato il comando [SerializedField])

Le variabili associate alle proprietà dell`oggetto devono essere nominate con my + tipo di dato (**è importante per la portabilità del codice**) ad esempio la Transform associata deve chiamarsi myTransform, myRigidBody2D ...

**Funzioni:** Tutte le funzioni iniziano con una lettera maiuscola, non ci sono spazi all`interno dei nomi ma le parole vengono separate dall`iniziale maiuscola, il nome deve essere il più descrittivo possibile. Cerco di creare più funzioni possibili per operazioni ripetute per aumentare la leggibilità del codice ed evitare di ripetere porzioni di codice.

**Script:** Gli script iniziano con lettera maiuscola, non ci sono spazi all`interno dei nomi ma le parole vengono separate dall`iniziale maiuscola, nome al solito più descrittivo possibile. In genere lo divido in tre parti: una regione **Variabili** che contiene tutte le variabili usate, una regione **Funzioni per Unity** che contiene le funzioni chiamate da unity ad esempio Start, Awake o Update e **Funzioni interne** che contiene tutte le altre funzioni usate dalla classe.

(ELIMINATO)~~GameSetup.cs: Lo script è pensato per impostare i confini della mappa di gioco in base alla grandezza della main camera usando una funzione della classe Camera “ScreenToWorldPoint” che trasforma le posizioni da screen space a world space.  
Vengono usati 4 oggetti di gioco composti da un boxCollider2D che rappresentano le 4 mura che vengono rimodellati e spostati ad ogni update di frame.~~

PlayerMovement.cs: È composto da 5 variabili settabili da inspector e una contenente il RigidBody2D del player (viene effettuato un controllo all`inizio dello script per verificarne la presenza e nel caso crearlo), le prime 4 son del tipo KeyCode e contengono i 4 tasti di movimento mentre la quinta il valore della velocità limitato in un range da 0.00 a 10.00.  
Il funzionamento dello script è semplice, vengono considerati i 9 casi di movimento possibile e con una struttura selettiva, in base ai tasti di movimento premuti, modifica la velocità dell`oggetto nella giusta direzione

SenseOfDepth.cs: Lo scopo dello script è dare un senso di profondità riducendo le dimensioni dei soggetti mobili in conformità con lo sfondo.  
Viene usata la funzione di una retta (in modo da avere una variazione ben proporzionata) calcolata a sistema con gli estremi (il valore massimo di y ovvero il punto più alto con il suo valore di scala deciso arbitrariamente e minimi).  
I valori usati (Modificabili da Inspector) al momento sono: per y=4(Y Max) valore di scala di 0.04(Min Scale) e per y=-4 (Y Min) valore di scala di 0.07 (Max Scale) quindi con y dato dal Transform del Player.

BulliedMovement.cs: Lo script si occupa di gestire il movimento del bullizzato. Da GDD il bullizzato deve scappare dal bullo più vicino; ho impostato un vettore di Transform dinamico in modo da poter modificare da Inspector il numero di bulli ma gli assegnamenti non sono automatici bisogna farli a mano una tantum, attraverso il calcolo della distanza tra due punti sul piano confronto le varie posizioni e trovo quello più vicino e aumento la velocità (per un valore settabile da inspector) in direzione opposta al bullo. Sempre da GDD in base all orientamento del Player il bullizzato deve o seguire o trattare il player come un bullo; In base a una variabile flag (per ora è settabile da inspector sarà da aggiornare una volta scriptato il GameMaster) dell`orientamento del player nella funzione update lo script si comporta in due modi: nel caso il player sia buono e sia ad una distanza minima (settabile da inspector) aggiunge velocità in direzione del player mentre se il player è flaggato cattivo tratta il player come un normale bullo e aumenta di 0,50f(settabile da Inspector) la velocità del bullizzato. Per dare la possibilità ai bulli e al player di andar dietro al bullo anche se si trova in un angolo ho impostato dei parametri (xMax, xMin, yMax, yMin) che rappresentano il limite entro il quale l`intelligenza artificiale si muove da sola (settabile da Inspector).