Funzione per determinare il varco migliore tra i Debris Orange (2D)

Versione 0

Funzione, da richiamare nella ***init()***, deve:

- restituire il **numero corrispondente al migliore varco tra gli Orange in cui passare** (varchi numerati da 0 a 3 in bianco nell'immagine di esempio)
- utilizzare **variabili globali** per gli **altri output da produrre**:

  1) la matrice **debris[11][4]** valorizzata dalla API del Game e di cui la funz. farà il debug (come programma ..OD_2_SIM..),

  2) le coordinate dei punti medi dei varchi da calcolare e salvare in **pmedio[4]** (come nel programma ...OD_2_SIM...)

  3) **le coordinate dei punti del percorso migliore da fare** (coordinate di **B** e **B1** del varco **3** nell'esempio) vanno salvate nelle 2 variabili globali: **PuntoGIU[3]** (nel semipiano **Y-** , nell'esempio: **B1**) e **puntoSU[3]** (in **Y+** , es: **B**)

la funzione (utilizzando **variabili locali**) deve quindi:

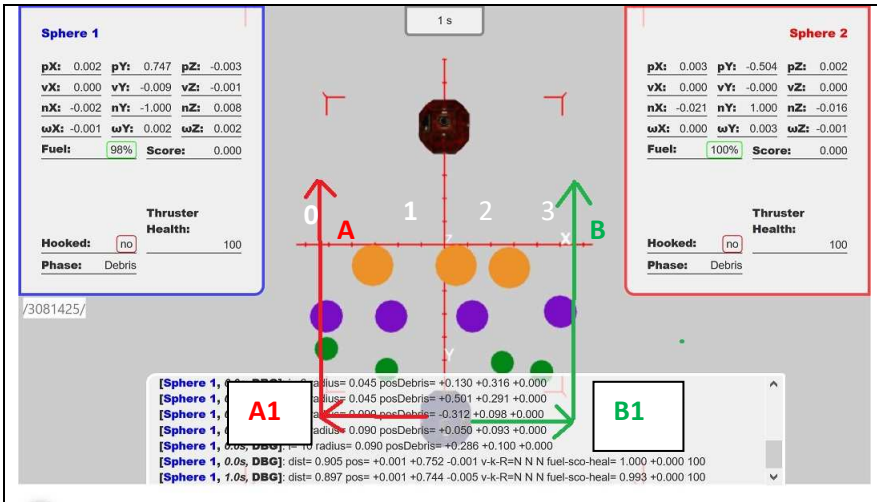
3) determinare il varco (o i 2 varchi) attraverso cui si potrebbe **passare** (non solo il più largo) **senza urtare gli Orange** o i bordi del campo di gioco `wall[6][3]`,

4) determinare, per ogni varco buono, il **numero di detriti verdi** e il **numero di detriti viola** che si urterebbero,

5) determinare il **numero corrispondente al percorso migliore da fare** (da 0 a 3) **e restituirlo alla *init***:

per il momento, assumiamo che il percorso migliore sia quello che **determina la riduzione minore della Thruster Health** passando per quel varco e **urtando il numero di verdi e viola** calcolato al punto 3)

Nella ipotesi (in via di verifica a cura di Matilda e Matteo) che possano determinarsi fino ad un massimo di 2 varchi tra gli Orange e che per attraversarli si proceda per segmenti (il primo sulla retta parallela all'asse X alla stessa altezza dello SPHERES BLUE, il secondo sulla retta parallela all'asse Y che passa per il punto medio), la funzione, dopo aver determinato le coordinate dei punti medi **A** e **B** e dei corrispondenti punti **A1** e **B1** (con *y = myState[1]*) deve **calcolare** per entrambi i percorsi il numero di detriti verdi e viola che si urterebbero, nell'esempio:



- 1 detrito verde e 1 viola per il percorso rosso (**-35%**),
- SOLO 1 detrito viola (**-25%**) per il percorso verde

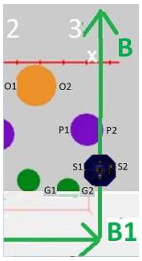
la funzione infine deve **restituire il numero** del varco migliore da attraversare (nell'esempio il varco **3**)

COME fare questo calcolo?

ci viene in aiuto la poesia del marinaio:

rosso al rosso , verde al verde

e la nave non si perde...



Indicando gli estremi dei diametri degli oggetti del percorso da **B1** a **B** con :

G1, **S1**, **P1** e **O1** **l'estemo a sinistra** rispettivamente del detrito **Green**, dello **SPHERES BLUE** (non in scala), del detrito **Purple** e di quello **Orange**, questi punti hanno **ascissa di valore minore** per l'oggetto,

G2, **S2**, **P2** e **O2** **l'estemo a destra** rispettivamente del detrito Green, dello SPHERES BLUE, del detrito Purple e di quello Orange, questi punti hanno **ascissa di valore maggiore** per l'oggetto,

lo **Spheres** **urterà** il detrito **Purple** perché per **S1** vale: **P1** <= **S1** <= **P2** invece **NON urterà** **Green** e **Orange** perché per **BLUE** valgono le seguenti condizioni: **S1** > **G2** **S1** > **O2**

Questo ragionamento va applicato ai detriti **Green** e **Purple** (tutti per semplicità) per calcolare il **numero di detriti verdi** e il **numero di detriti viola** che si urterebbero lungo un certo percorso/varco considerato tra gli **Orange**