

Classe Giocatore (Model)

La classe contiene una serie di attributi utili per memorizzare le informazioni del giocatore:

Nome: Stringa

Id: Int

Colore: Int

Un giocatore è caratterizzato da un colore, per memorizzarlo si usano indici da 0 a 4 (5 colori), memorizzato nella variabile.

Danni: int[4]

Vettore di interi che rappresenta i danni subiti da ogni giocatore, per identificare chi è il giocatore che ha fatto il danno si usa l'indice dell'array (esempio il colore blu corrisponde all'intero '0', vedi nella variabile *colore*, quindi in posizione [0] dell'array ci sono i danni che sono stati dati dal giocatore con il colore blu). Non ci sono limiti sui danni, se non che la somma non può superare il valore 12. Alcuni danni hanno delle caratteristiche speciali perché corrispondono a punti extra, questi saranno considerati nella valutazione del Control, e sono:

- Danno 1 (1 punto) – Primo sangue
- Danno 11 (6 punti) – Uccisione
- Danno 12 (1 punto) – Infierire (eventualmente solo a chi ha fatto il danno dell'uccisione)

Marchi: int[4]

Vettore di interi con analogo significato al vettore *danni*. Ha il vincolo che si possono avere al massimo tre marchi da un dato giocatore nemico, e massimo tre marchi da ogni altro giocatore (significa nel complesso 12 marchi massimo, tre per ogni giocatore avversario).

Munizioni: int[3]

Vettore di interi che rappresenta per ogni indice le munizioni restanti di un dato colore. Anche in questo caso si rappresenta il colore con un intero che è indice dell'array (blu, rosso, giallo)→(0,1,2). Valore massimo di ogni cella è 3 (massimo tre munizioni per colore), minimo zero ovviamente.

Gun: Arma[3]

Vettore che tiene conto delle armi possedute dal giocatore, possono essere massimo tre.

Pow: Potenziamento[3]

Vettore che tiene conto dei potenziamenti posseduti dal giocatore, possono essere massimo tre.

Cel: int

Death: int

Variabile che indica il numero di volte che il giocatore è morto e serve per sapere quanti sono i massimi punti attribuibili in caso di nuova morte ai giocatori che hanno fatto danno.

Action: int

Primosangue: int

Variabile che tiene conto di chi è il giocatore che ha fatto il danno 1 (primo sangue).

Score: int

Per questi attributi sono necessari dei metodi interni alla classe per descrivere il modello (pensa come ad un'analisi statica). Oltre ai metodi utili e ai significati, dettaglio anche le possibili evoluzioni, e le modifiche che possono essere apportate ai parametri. Questi metodi andranno nel Controller, qui faccio un'analisi di insieme per pensare a tutto quello che può essere necessario.

Costruttore. Nel costruttore ho passato come parametri il nome scelto dal giocatore e la miniatura scelta. In base al colore della miniatura si inizializza la variabile intera *colore*. Importante in questo caso fare il controllo che non si passi il colore di una miniatura già scelta. Per ogni giocatore della partita ci possono essere massimo cinque colori distinti, e nessun colore aggiuntivo oltre a quelli considerati ovviamente, che corrispondono agli interi da 0 a 4. Il vettore che registra i danni e quello che registra i marchi vengono

inizializzati a tutti zeri, mentre il vettore delle munizioni a tutti 3 (valore massimo), perché ognuno parte con il massimo numero di munizioni). Punteggio, azione, morte e cel sono inizializzate a zero.

public void getname(). Ritorna il nome del giocatore.

public void getid(). Ritorna l'ID del giocatore.

public int getnumberdamage(Player enemy). Ritorna il numero dei danni inflitti al giocatore da uno specifico nemico.

public int gettotaldamage(). Ritorna il numero totale di danni subiti dal giocatore.

Controlli sulle variabili:

- Il colore è univoco, non si possono dare stringhe diverse dai colori predefiniti, e due giocatori non possono avere lo stesso colore.
- La stringa del nome è univoca, non ci possono essere giocatori con lo stesso nome. Stessa cosa per l'id.
- Il primo giocatore, quello che avrà la tessera di giocatore iniziale, andrà nella prima posizione dell'array che tiene traccia dei giocatori.
- Il vettore che registra i danni e quello che registra i marchi hanno 5 elementi ciascuno, tuttavia al giocatore considerato possono fare danno i quattro giocatori avversari quindi si dovrà controllare ogni volta che si aggiornano i valori del vettore che in corrispondenza dell'indice pari al valore di ritorno della getcolor() chiamata sul giocatore ci sia sempre il valore zero. Ovvero in parole povere che per nessun giocatore si registrino marchi o danni autoinflitti, nella cella corrispondente all'indice del proprio colore si abbia sempre valore nullo.
- Il vettore che registra i danni ha come somma massima dei propri elementi il valore 12.
- La somma degli elementi del vettore danni è significativa per stabilire le mosse lecite ed eventuali miglioramenti.