PROGETTO PROGRAMMAZIONE NON-GRAVITAR 2018/2019

Non-Gravitar è un gioco da terminale. Il codice è stato scritto da Luca Genova e Matteo Lorenzoni, senza l’utilizzo di librerie grafiche, utilizziamo i caratteri:

* **Y** ->la nostra navicella ed avrà cinque vite, ed il carburante è inizialmente settato a …
* **O** -> per i pianeti
* **X** -> per i pianeti distrutti (ogni qualvolta vengano distrutti tutti i bunker all’interno di un pianeta, il pianeta è distrutto e al posto del carattere **O** viene messa una **X**)
* **c** (minuscolo) -> il carburante con meno capacità di ricarica (ricarica di …)
* **C** (MAIUSCOLO) -> il carburante con massima capacità di ricarica (ricarica di …)
* **b** (minuscolo) -> i bunker che sparano in due direzioni ed avranno tre vite
* **B** (MAIUSCOLO) -> i bunker che sparano in tre direzioni ed avranno tre vite
* **.** (il punto) -> per gli spari
* **/ “ \** -> per la superfice dei pianeti

I comandi di gioco sono anche scritti nella "tabella" destra della schermata di gioco:

* **FRECCIA SU** -> per spostarsi in su con la navicella
* **FRECCIA GIU** -> per spostarsi in giù con la navicella
* **FRECCIA DESTRA** -> per spostarsi a destra con la navicella
* **FRECCIA SINISTRA** -> per spostarsi aa sinistra con la navicella

Dato che abbiamo realizzato il tutto basandoci su di una matrice lo spostamento, su, giù, destra e sinistra equivale ad andare avanti di una posizione e cancellare la posizione precedente.

* **BARRA SPAZIATRICE** -> spara i proiettili verso il basso
* **CARATTERE Q** -> tenendo premuto questo carattere (all’interno di un pianeta) attiveremo il raggio traente, basta rilasciare il tasto per disattivarlo, che “aggancerà” i caratteri **c** e **C** dei due diversi tipi di carburante

Distrutti tutti i pianeti è possibile accedere al nuovo Sistema Solare andando verso il bordo destro.

Per uscire da un pianeta basta salire con la navicella verso l’alto.

Il progetto è composto da otto classi .cpp ed anche gli equivalenti sette file .h (non otto perché uno è il Main):

1. **Mapp** -> utilizziamo una classe a parte per avere un’unica matrice sia per il sistema solare, sia per i pianeti.
2. **Spaceship** -> la classe della navicella, contiene anche un oggetto di tipo bullets
3. **SolarSystem** -> la classe che contiene la lista di pianeti (per ogni sistema solare diverso)
4. **PlanetSurface** -> la classe del pianeta, che genera una superficie random per ogni pianeta, gestisce i proiettili della navicella e dei bunker, contiene i due diversi tipi carburanti
5. **BulletsList** -> la classe dei proiettili, che contiene i metodi add e delete bullets, verrà usata sia dalla navicella che dai bunker
6. **Bunker1** -> prima classe bunker che ha al suo interno due oggetti di tipo bullets, dei proiettili che sparano verso sinistra e dei proiettili che sparano verso destra
7. **Bunker2** -> prima classe bunker che ha al suo interno tre oggetti di tipo bullets, dei proiettili che sparano verso sinistra, dei proiettili che sparano in direzione centrale e dei proiettili che sparano verso destra
8. **Main** -> classe di esecuzione dove viene gestito il ciclo di gioco in base anche alla vita ecc.. //aggiungi qualcosa al main

I vari metodi delle classi verranno descritti meglio nella relazione.

NOME E COGNOME: Luca Genova

MATRICOLA: 0000882970

EMAIL: [luca.genova2@studio.unibo.it](mailto:luca.genova2@studio.unibo.it)

NOME E COGNOME: Matteo Lorenzoni

MATRICOLA: ….

EMAIL: [matteo.lorenzoni@studio.unibo.it](mailto:matteo.lorenzoni@studio.unibo.it)