

---

---

# Non-Gravitar

Progetto d'esame  
Programmazione

---

---

# Gravitar

***Gravitar*** è un gioco arcade di grafica vettoriale a colori pubblicato da Atari, Inc. nel 1982.

Live demo:

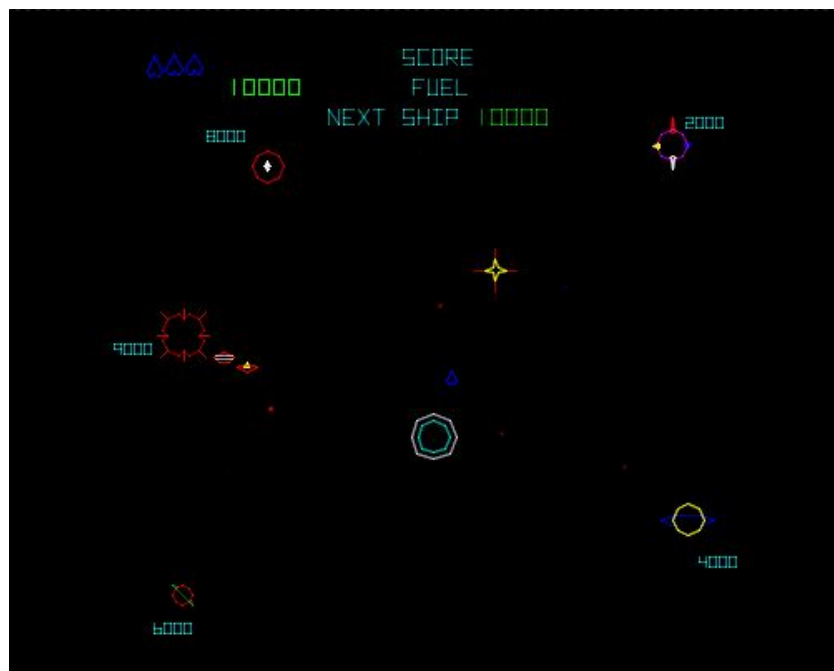
<https://my.ign.com/atari/gravitar>



# Atari 2600



# Gameplay - Storyline



Il giocatore controlla una piccola astronave blu. Il gioco inizia in un immaginario **sistema solare** con diversi **planeti** da **esplorare**.

Se il giocatore sposta la sua nave su un pianeta, **cambia la visuale di gioco** per mostrare la superficie del pianeta (*visuale pianeta*; vedi prossime slide).

# Gameplay - Controller

In Gravitar i pianeti hanno la gravità, in Non-Gravitar no.  
Il giocatore ha 6 pulsanti:

- 4 per **spostarsi**: su, giù, dx, sx
- 1 per **sparare** nella direzione di movimento
- 1 per attivare il **raggio traente**

Ogni pianeta ha una superficie diversa e sulla superficie si trovano:

- Bunker (i nemici)
- Carburante; il raggio traente serve prendere carburante

# Gameplay - Combattimenti

Nella *visuale pianeta*, il giocatore deve **distruggere i bunker** rossi che sparano costantemente.

Una volta distrutti tutti i bunker, il **pianeta esploterà** e il giocatore guadagnerà un **bonus**. Una volta che tutti i pianeti saranno distrutti, il giocatore si sposterà su un **altro sistema solare**.



# Gameplay - Game Over



Il giocatore perde un **punto vita (HP)** se si schianta sul terreno o viene colpito da un nemico, e il gioco termina immediatamente se il carburante si esaurisce.

# Requisiti

1. Le *visuali pianeta* ed i sistemi solari devono essere **generati casualmente e a run-time**, se però il giocatore torna indietro deve poter vedere le schermate di gioco come le ha lasciate in precedenza.
2. Ci devono essere almeno **2 tipi di bunker** diversi, con diverse strategie di attacco: il primo spara a caso in 2 direzioni diverse, il secondo in 3 direzioni.
3. Ci devono essere almeno **2 tipi di carburante** diverso, con diverse capacità di ricarica del carburante.



# Linee Guida

1. **Minimizzate il numero di righe di codice:** riutilizzate il codice che avete già scritto, cercate soluzioni semplici.
2. **Commentate il codice.** Non serve un commento ogni linea di codice, ma cercate di spiegare le soluzioni meno semplici ed intuitive.
3. I nomi delle variabili devono essere **auto esplicativi**. Poco importa se sono lunghi 30 caratteri, sono una forma di documentazione del codice.

# Impostazione del progetto

1. Il progetto deve essere realizzato usando le classi
  - a. Le classi sono tipi di dato definiti dal programmatore
  - b. Definiscono sia i valori (campi) che il comportamento (metodi) del dato
  - c. Un dato il cui tipo è una classe si chiama oggetto
2. Il progetto è organizzato in più file
3. Ad ogni classe corrispondono due file: ad esempio Planet.**cpp** e Planet.**hpp**
4. .hpp è l'estensione dell'interfaccia

# Planet.hpp

```
class Planet{
protected:
    int circumference;
    list<ControlPoint> surface;
    list<Bunker> bunker_list;
    list<Fuel> fuel_list;
    Planet(int circumference, int bunkers_count); // bunkers_count
        ↪ -> destroy all bunkers, destroy the planet
    ...
public:
    void method()
    ...
}
```

# Planet.cpp

```
#include Planet.hpp  
  
void Planet::method(){  
    // do something  
}
```

In ogni file in cui si usa il tipo Planet occorre importare Planet.hpp!

# Regole e valutazione

- Il progetto è parte integrante dell'esame ed è obbligatorio
- Ha un voto massimo di **8 punti** che si sommano al voto dello scritto (max 24)
- Il lavoro viene svolto in gruppi, ma la valutazione è individuale
- Studenti dello stesso gruppo possono prendere voti diversi
- Il progetto **si consegna una volta sola** e il voto è valido per l'intero anno accademico (e per gli anni successivi se le regole non cambiano)
- Chi non ottiene almeno 2 punti alla discussione dovrà presentare un nuovo progetto

# Gruppi

- Il progetto si svolge in gruppi **di 3-4 persone, ma il voto non è di gruppo**
- **NON è possibile fare gruppi di 5 o più persone**
- Sono ammessi anche progetti individuali in casi eccezionali (ma il carico di lavoro non è ridotto)
- Auto-organizzatevi per la creazione dei gruppi
- Comunicherete i componenti del gruppo in fase di prenotazione e/o discussione del progetto

# Quando presentare il progetto?

- Il progetto potrà essere presentato entro il 01/11/2019
- Orientativamente un appello orale ogni 3 mesi, non troppo vicino agli appelli scritti
- Date e aule da confermare. Saranno pubblicate su AlmaEsami e annunciate sul forum del corso
- Se ci sono molte richieste possiamo fissare ulteriori date, anche fuori dalle sessioni ordinarie

# Consegna e discussione

- Il progetto si consegna due giorni prima della discussione
- TUTTI i membri del gruppo devono partecipare alla discussione (portare un laptop)
- La discussione riguarda solo il progetto
- Il gruppo consegna:
  - Codice (sorgenti e binari)
  - README
  - Breve relazione (3-4 pagine) che presenti le principali scelte nella definizione del gioco e nella sua implementazione
- **Consegna via mail a entrambi i tutor ([francesco.sovrano2@unibo.it](mailto:francesco.sovrano2@unibo.it); [adele.veschetti2@unibo.it](mailto:adele.veschetti2@unibo.it)): mandare un link ad un folder condiviso**  
(va bene Dropbox, Google Drive o qualunque altro servizio)