

Nut Planet: Simulatore di Gravità

Enrico Mannocci

September 30, 2019

Contents

1	A cosa serve Nut Planet?	2
2	Modalità	2
3	Finestre del Simulatore	2
3.1	Finestra Universo	2
3.2	Finestra Movimento	2
3.3	Finestra Inserimento Nuovo Oggetto Massivo	2
3.4	Finestra Impostazioni	3
3.5	Finestra Informazioni Pianeta	4
3.6	Finestra Informazioni Universo	4
4	Esplorando il codice	4
4.1	Calcoli	5
5	Effetti Speciali	6
5.1	Esplosioni	6
6	ScreenShot	7

1 A cosa serve Nut Planet?

Il programma permette di visualizzare graficamente come evolve un sistema di masse puntiformi soggette alla forza attrattiva gravitazionale.

2 Modalità

Il menù principale permette la scelta fra "Modalità Sistema Solare", "Modalità Creativa", "Credits" e "Esci". Selezionando la prima, compare una schermata che mostra un modello semplificato dei pianeti del nostro sistema solare mentre ruotano attorno al Sole immobile al centro, è concessa l'aggiunta di oggetti massivi ulteriori. La seconda opzione lascia completamente carta bianca all'utente, creando un universo vuoto nel quale è possibile aggiungere oggetti massivi a piacimento. La terza fa apparire un disegno sul terminale e la quarta chiude il programma.

3 Finestre del Simulatore

A seguito dell'avvio del simulatore mediante la selezione della modalità vengono aperte 6 finestre. Di seguito presentate.

3.1 Finestra Universo

Rappresenta l'universo attualmente attivo e tutti gli oggetti che esso contiene con l'aggiunta del sole come punto fisso nel caso in cui si abbia scelto la modalità Sistema Solare.

3.2 Finestra Movimento

Finestra utilizzata per spostare la camera che inquadra la porzione di universo attivo visualizzata sulla Finestra Universo. I comandi sono 7: Sinistra, Destra, Sopra, Sotto, +, -, e Torna A Casa. I primi quattro spostano la camera, del 10% della lunghezza inquadrata nella direzione che identificano, i seguenti due modificano sempre del 10% lo zoom e l'ultimo riporta il centro della camera alle coordinate origine (0,0).

Movimento			
+	Su	-	Back Home
Sinistra	Giù	Destra	

Figure 1: Finestra Movimento

3.3 Finestra Inserimento Nuovo Oggetto Massivo

Questa finestra serve appunto per inserire un nuovo oggetto massivo nell'universo attualmente attivo e consultabile sulla Finestra Universo, qualunque modalità si abbia scelto, è necessario premere il tasto **Spawn Planet** nella finestra dopo aver terminato la compilazione di tutti i dati richiesti. Questi ultimi sono:

- Nome (Stringa, tutti i caratteri ASCII sono consentiti)[Default = '-']
- Massa (Numero Reale **positivo**, bisogna inserire una sola volta il carattere '.' e deve essere **diversa da 0**)[Default = 1]
- Raggio (Numero Reale **positivo**, bisogna inserire una sola volta il carattere '.')[Default = 1]
- Posizione X (Numero Reale, stesse regole del punto precedente, valore della componente x della posizione)[Default = Componente X del punto in cui è posizionato il centro della camera osservabile nella Finestra Universo]
- Posizione Y (Numero Reale, stesse regole del punto precedente, valore della componente y della posizione)[Default = Componente Y del punto in cui è posizionato il centro della camera osservabile nella Finestra Universo]
- Velocità X (Numero Reale, stesse regole del punto precedente, valore della componente x della velocità)[Default = 0]
- Velocità Y (Numero Reale, stesse regole del punto precedente, valore della componente y della velocità)[Default = 0]
- Colore [Default = Trasparente (non visibile)]

Add Planet		
Name:		
Mass:		kg
Radius:		m
Position		
X:		m
Y:		m
Speed		
X:		m/s
Y:		m/s
Color:		
Spawn Planet		

Figure 2: Finestra Aggiunta Oggetto Massivo

3.4 Finestra Impostazioni

Questa finestra permette la modifica di alcune impostazioni di base riguardanti la rappresentazione degli oggetti massivi attraverso 3 sliders. Il primo modifica la velocità dello scorrere del tempo, il secondo varia il raggio con cui viene rappresentato il sole e l'ultimo cambia il raggio con cui vengono mostrati gli oggetti massivi. A proposito di questo ultimo punto e di quello che lo precede si ricorda che le dimensioni con cui si rappresentano gli oggetti massivi e il sole non sono quelle reali, queste sono utilizzate solo per quanto riguarda le collisioni. Questa scelta è stata fatta al fine di mostrare i pianeti che altrimenti sarebbero stati invisibili. Uno svantaggio di questa scelta riguarda i pianeti più vicini al sole che quando la telecamera inquadra sezioni molto vaste di universo risultano interni al sole se pur essi non lo siano realmente.

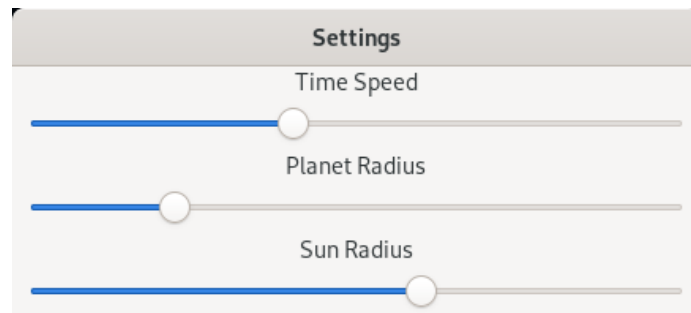


Figure 3: Finestra Aggiunta Oggetto Massivo

3.5 Finestra Informazioni Pianeta

Questa Finestra permette di visualizzare gli attributi principali dei pianeti precedentemente aggiunti nel simulatore, sia che essi appartengano al sistema solare o che siano stati aggiunti dall'utente. Le informazioni visualizzabili sono il nome, la massa ed il raggio. Con i pulsanti **Previus** e **Next** è possibile cambiare il pianeta di cui si stanno visualizzando le informazioni. Quando avviene un cambio di pianeta la camera si centra sulla posizione attuale di esso. Il pulsante **Delete** serve invece per rimuovere l'oggetto massivo attualmente selezionato.

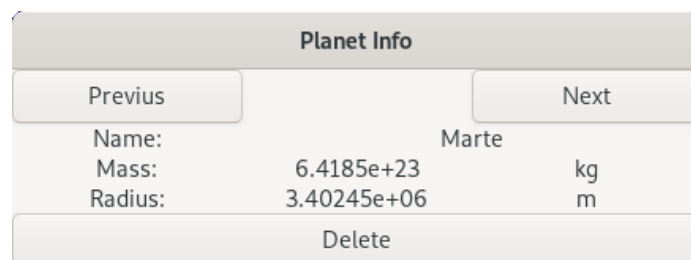


Figure 4: Finestra Informazioni Pianeta

3.6 Finestra Informazioni Universo

Questa finestra mostra la posizione attualmente puntata dalla camera, che è anche evidenziata da un puntatore azzurro disegnato nell finestra universo: al centro di essa.

4 Esplorando il codice

Il programma è composto da due file:

- Il primo adibito ai calcoli ed all'ascolto dei vari input
- Il secondo adibito alla parte grafica del Menu Generale

Sono stati creati due tipi (struct) di cui vale la pena parlare:

- Vettore
- Pianeta

Il primo adibito al calcolo e alla creazione di vettori nello spazio bidimensionale, il secondo invece fa da stampo per ogni oggetto massivo creato. Il programma contiene un "vector" di Pianeti di cui fanno parte tutti gli oggetti massivi aggiunti ed un ciclo che permette tutti i calcoli quante più volte riesce a farli il processore. Una volta fatti i calcoli il tutto viene continuamente disegnato sulla schermata permettendo una animazione dei pianeti.

4.1 Calcoli

I calcoli effettuati dal programma non sono altro che la risoluzione numerica del problema di un corpo su cui agisce una forza. Una volta che questa è stata trasformata in accelerazione il programma non fa altro che modificare la velocità e poi con questa la posizione dell'oggetto. L'intervallo di tempo utilizzato nei conti per calcolare la variazione di velocità e spazio cambia a seconda della posizione dello slider **Time Speed**.

Di seguito Riporto alcuni pezzi di codice commentati.

```

330     for (int c=0; c<P.size(); c++){
331         //Calcolo della forza con il SOLE!
332         r = sqrt(pow(P[c].getPosizione().getX(),2) + pow(P[c].getPosizione().getY(),2));
333         P[c].cancForza();
334         if (sistema solare){
335             P[c].addForza((G*M*P[c].getMassa())/(r*r*r));
336         }

```

Figure 5: Prima parte sostanziale del metodo Motore

In questo pezzo di codice viene calcolata la distanza del c-esimo pianeta del vector rispetto al sole posizionato in (0,0). Di seguito vengono poi cancellate le forze riguardanti precedenti iterazioni e nel caso in cui la modalità fosse "Sistema Solare" allora viene aggiunta alla risultante delle forze (che in questo momento è nulla) la forza di gravità tra sole e oggetto valutata con la **Legge di Gravitazione Universale** (l'r cubo al denominatore è dovuto al calcolo vettoriale).

```

337         //Calcolo della forza con gli altri corpi!
338         for (int cc=0; cc<P.size(); cc++){
339             if (cc!=c){
340                 r = sqrt(pow(P[c].getPosizione().getX()-P[cc].getPosizione().getX(),2) + pow(P[c].getPosizione().getY()-P[cc].getPosizione().getY(),2));
341                 if (r<(P[c].getRaggio()+P[cc].getRaggio())){
342                     if (Cestino.size()==0){
343                         Cestino.push_back(c);
344                         Cestino.push_back(cc);
345                         E.push_back(Esplosione(P[c].getPosizione()));
346                     }
347                     }else{
348                         P[c].addForza((G*P[cc].getMassa()*P[c].getMassa())/(r*r*r), P[cc].getPosizione());
349                     }
350             }
351         }
352         P[c].calcDati();
353     }

```

Figure 6: Seconda parte sostanziale del metodo Motore

Qui invece vengono calcolate le forze di gravità tra pianeti, i passaggi sono gli stessi del punto precedente. In aggiunta si fanno alcuni controlli e si riempie un secondo vector chiamato "Cestino" nel caso in cui due pianeti siano troppo

vicini! Si noti infine la riga numero 352 che calcola accelerazione, variazione di velocità e posizione per ogni oggetto.

```

122 void cancForza(){
123     forza.setX(0);
124     forza.setY(0);
125 }
126 Vettore temp = Vettore(0,0);
127 void addForza(double f, Vettore v=Vettore(0,0)){
128     temp.setX(f*(posizione.getX()-v.getX()));
129     temp.setY(f*(posizione.getY()-v.getY()));
130     forza=forza+temp;
131 }
132 void calcDati(){
133     accelerazione.setX(-forza.getX()/massa);
134     accelerazione.setY(-forza.getY()/massa);
135     velocita.setX(velocita.getX() + T*accelerazione.getX());
136     velocita.setY(velocita.getY() + T*accelerazione.getY());
137     posizione.setX(posizione.getX() + T*velocita.getX());
138     posizione.setY(posizione.getY() + T*velocita.getY());
139 }

```

Figure 7: Metodi di calcolo della struct Pianeti

Nel precedente codice vengono mostrate le definizioni dei metodi usati in Motore. La funzione `cancForza` non fa nulla se non azzerare le forze calcolate in precedenti iterazioni, `addForza` aggiunge, mediante l'overload dell'operatore `+` binario tra vettori, la forza ricevuta come argomento alla risultante mentre `calcDati` calcola l'accelerazione, la nuova velocità e posizione del pianeta. La prima grazie al secondo principio della dinamica mentre la seconda e la terza rispettivamente con le relazioni lineari che legano accelerazione, tempo e velocità e velocità, spazio e tempo.

5 Effetti Speciali

5.1 Esplosioni

In caso di collisione tra due oggetti (viene definita tale una distanza tra i due oggetti minore o uguale della somma dei raggi di questi ultimi), il simulatore produce un'esplosione che oltre a far apparire una semplice animazione cancella dal vector che contiene tutti gli oggetti questi ultimi due. Ne consegue che a seguito di una collisione entrambi gli oggetti che collidono vengono eliminati irrimediabilmente.

6 ScreenShot

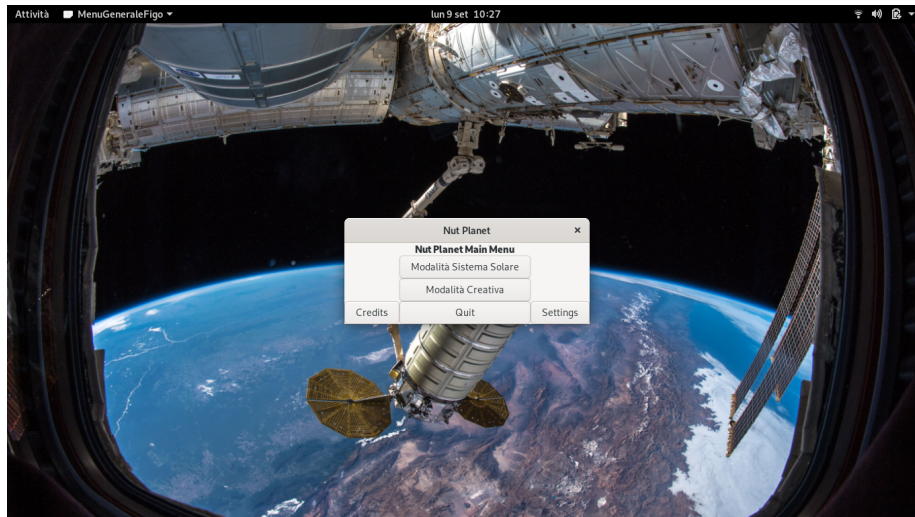


Figure 8: Menu Generale

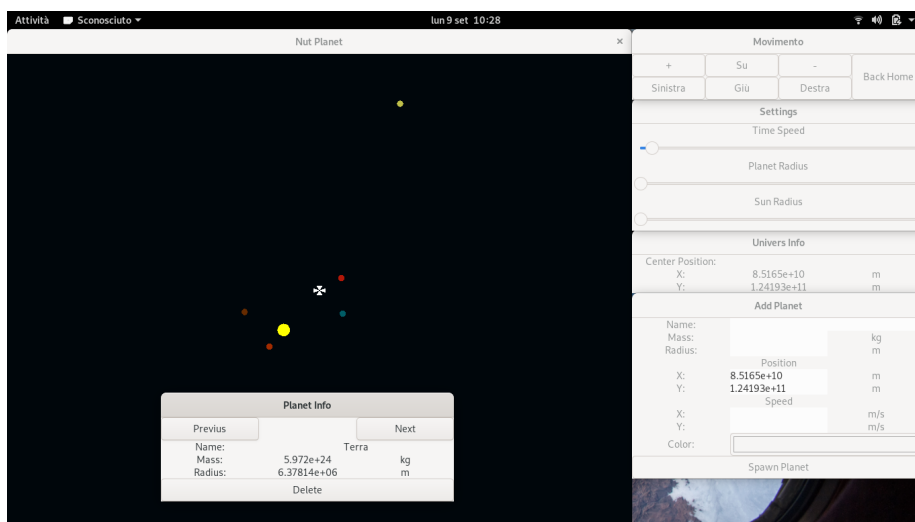


Figure 9: Esempio di visualizzazione

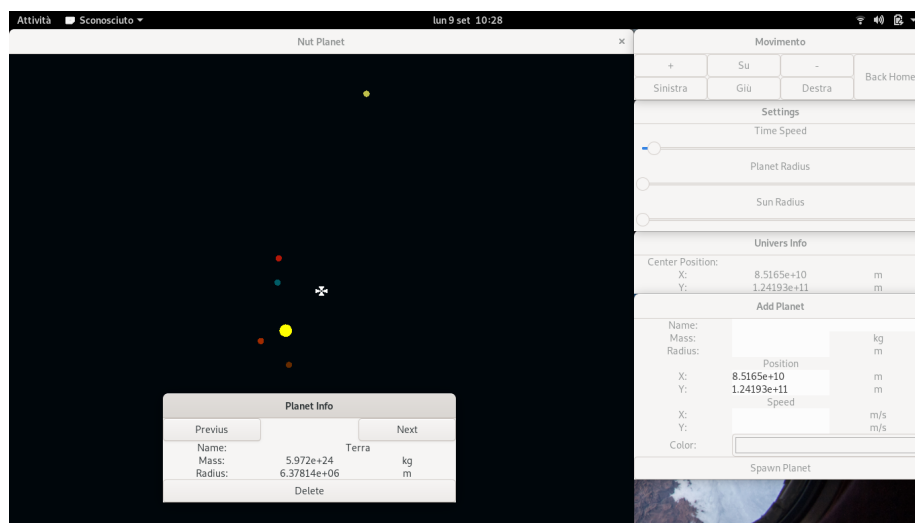


Figure 10: Esempio di visualizzazione

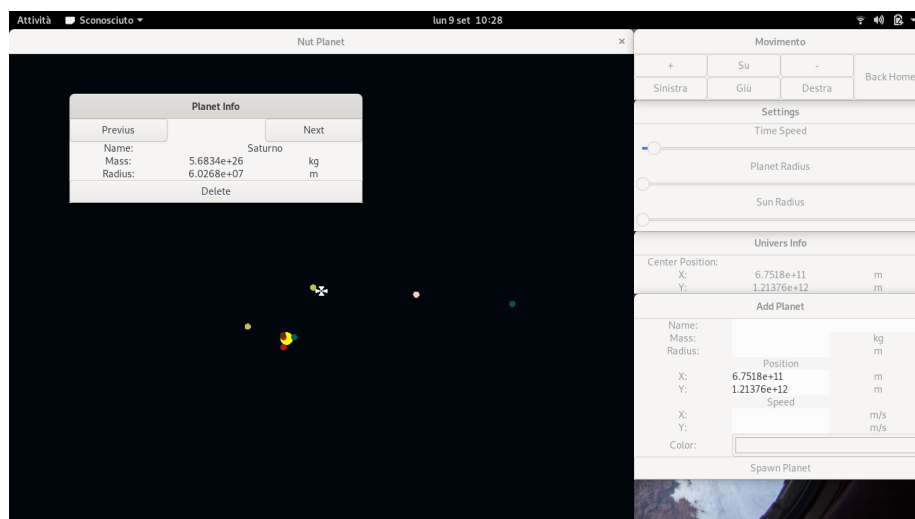


Figure 11: Esempio visualizzazione intero Sistema Solare

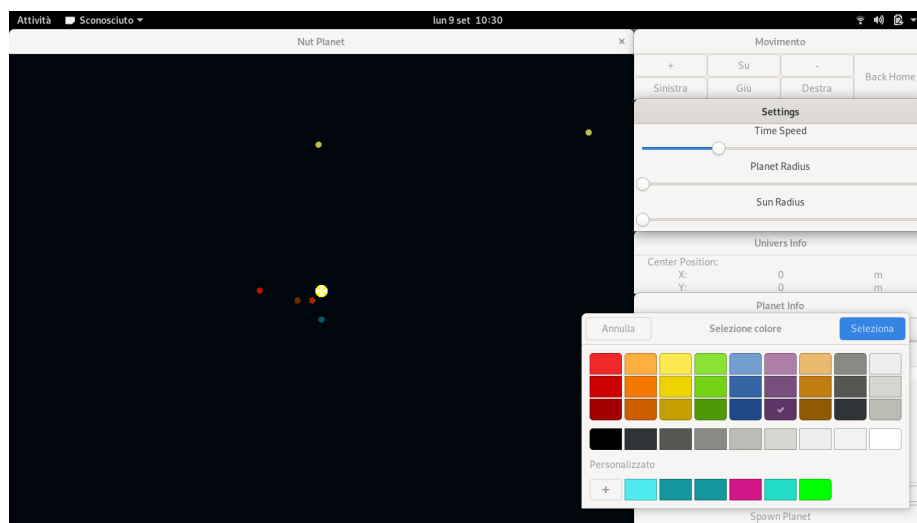


Figure 12: Cambio Colore