

Phun, processore fisico virtuale

(i tutorial di Alessandro de Simone)

Copyright Alessandro de Simone 2008 (www.alessandrodesimone.net) - È vietato trascrivere, copiare, stampare, tradurre, riprodurre o divulgare il presente documento, anche parzialmente, senza l'autorizzazione scritta dell'autore. I siti Internet, le case editrici e le pubblicazioni di settore che intendano utilizzare questo documento possono contattare l'autore (contatti@alessandrodesimone.net) per gli accordi del caso.

Copyright Alessandro de Simone 2008 (www.alessandrodesimone.net) - No transcribing, no copyng, no reproducing, no translating, no printing, no publishing this document - even if partially - without author's written authorization. Websites and publishing house who wish to employ this document must write the author (contatti@alessandrodesimone.net).

Premesse

Le simulazioni al computer consistono nella capacità di animare una pluralità di oggetti e di dotarli di una sorta di "vita propria" in modo indipendente l'uno dall'altro. Ricorrendo a un esempio banale, ma efficace, supponiamo di voler simulare il gioco del biliardo: si tratta di visualizzare e gestire diverse palle che, in collisione tra loro e con i bordi del tavolo (oltre che con la punta dell'asta del giocatore), percorrono tragitti sottoposti alle leggi della fisica. Nel caso specifico, una palla da biliardo è sottoposta alle seguenti forze:

- a) Forza di gravità
- b) Attrito del fondo del tavolo
- c) Forza impressa dall'asta del giocatore
- d) Forza impressa da altre palle con le quali entra in collisione.
- e) Forza di repulsione dei bordi del tavolo.

Nel caso della forza di gravità, questa dovrebbe avere influenza pressoché nulla in quanto il tavolo è perfettamente orizzontale. L'influenza della forza di gravità è però *indirettamente* presente, in quanto legata al parametro **peso** (della palla) e **attrito** (del tappeto verde): maggiore è il peso della palla, maggiore sarà l'influenza dell'attrito che tenderà comunque a rallentare la palla.

Un programma chiamato a gestire una situazione del genere dovrà costantemente considerare la traiettoria delle singole palle che risultano via via in movimento (tutte, all'inizio, sono ferme) e intercettare, istante per istante (con intervalli di tempo tanto minori quanto maggiore è la precisione richiesta) eventuali collisioni e applicare le leggi che la Fisica richiede in questi casi: modifica della traiettoria, alterazione della quantità di moto in seguito a un impatto, conseguente diminuzione (se la palla urtata è più piccola) o aumento della velocità (in caso contrario) e perfino aumento della temperatura in seguito a collisione (che in teoria comporta una variazione del volume delle palle coinvolte).

Un programma di tal genere assorbe una quantità di risorse direttamente proporzionale al numero di oggetti in movimento e al grado di precisione richiesto alla simulazione.

Di recente si stanno compiendo studi per trasferire il gravoso impegno di calcolo a specifici processori (circuiti integrati) che affiancherebbero - nei computer delle prossime generazioni - il processore centrale e quello grafico.

Phun

Il programma **Phun**, scaricabile e utilizzabile liberamente (ma non per scopi di lucro) dal sito <http://www.acc.umu.se/~emilk/> è un sofisticato programma capace di gestire numerosi oggetti e di sottoporli a un processore fisico virtuale. Il campo di azione è solo bidimensionale, ma gli esperimenti che si possono compiere sono incredibilmente spettacolari, tanto quanto la facilità con cui è possibile crearli.

In questo tutorial descriverò i primi passi da compiere per impadronirsi rapidamente delle sue notevoli potenzialità, in modo da procedere in seguito con propri esperimenti.

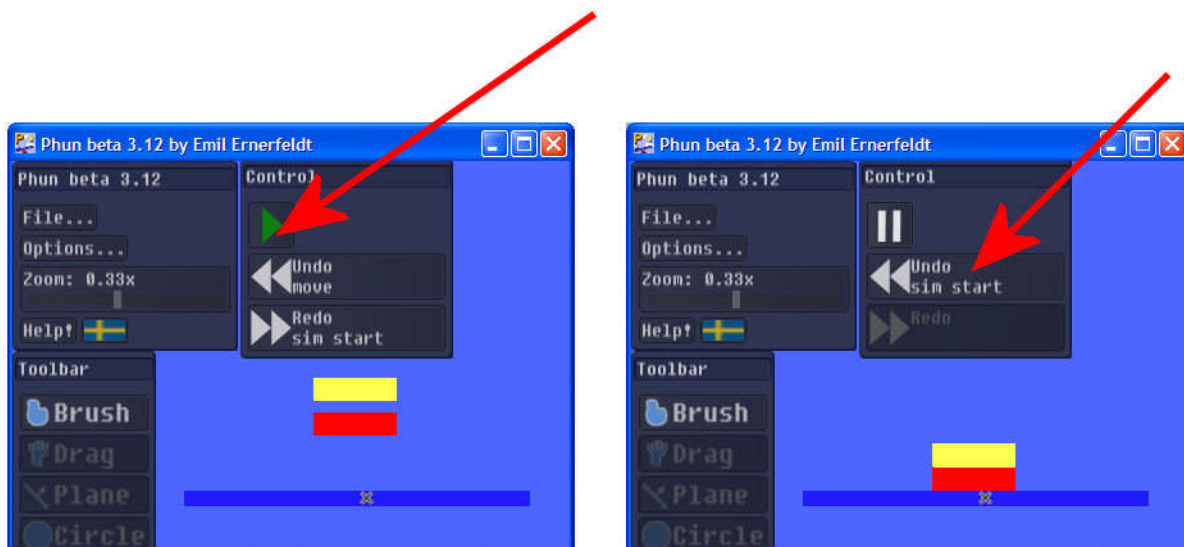
Si consiglia di scaricare, sempre dal mio sito (www.alessandrodesimone.net/phun) i diversi "scenari" descritti nel presente tutorial. Tutti gli scenari sono in realtà semplici file di testo dotati di suffisso **phn** e sono inclusi nel file **2008_03_01_Tutorial_Phun.zip**.

Primo scenario: A_Due_rettangoli.phn

Il primo scenario è utile per rendersi conto dell'ambiente in cui si opera. Lo scenario di *Phun*, infatti, si riferisce sempre a una rappresentazione della realtà immaginata **verticale**, nel senso che simula la visione comune di un osservatore che, in piedi, guarda dinanzi a sé. Le direzioni possibili sono quindi esclusivamente *alto*, *basso*, *destra* e *sinistra*. Manca quindi la *profondità*, ad eccezione dei piani sui quali possono essere posizionati i vari oggetti. Inoltre, come vedremo in uno dei prossimi Tutorial (in preparazione), sarà possibile simulare una visione ancora bidimensionale, ma osservabile **in pianta**, cioè come se l'osservatore guardasse verso il basso.

Nello scenario *A_Due_rettangoli.phn* vi sono tre rettangoli. Quando si carica uno scenario (menu *File/Load scene*) la simulazione viene sempre impostata come "bloccata", pronta però per essere messa in movimento. Nel menu *Control* dovrebbe infatti essere visualizzato il tasto verde *Play* (a sinistra, nell'immagine qui in basso).

Per posizionare al centro i tre rettangoli (qualora non visibili o non correttamente posizionati) cliccate con tasto destro del mouse in un punto qualunque del cielo e trascinate il cursore. Per *ingrandire* / *ridurre*, invece, agite sulla rotellina del mouse.



Assicuratevi che sia applicato un segno di spunta nella casella *Gravity* (finestra *Options* vedi figura) e premete quindi il tasto *Play*. Verrà subito attivata la simulazione, vale a dire che la forza di gravità verrà applicata agli oggetti presenti nello scenario.



Dal momento che il rettangolo blu è fisso, i due rettangoli giallo e rosso precipiteranno verso il basso, fino a toccarlo, rimbalzando impercettibilmente più volte. Per ripetere la simulazione basterà premere il pulsante *Undo sim start*. Approfittatene per scoprire l'influenza degli altri segni di spunta nelle caselle della finestra *Options*: il cursore *Speed* permette di gestire la velocità con cui procede la simulazione; la casella *Draw cloud* si limita semplicemente a visualizzare delle nuvole che scorrono sullo schermo; il cursore *Zoom* svolge le stesse funzioni della rotellina del mouse.

Da questo primo scenario abbiamo imparato tre cose:

- Tutti gli oggetti (se non diversamente sollecitati) precipitano verso il basso, ad eccezione di quelli dotati di un "blocco" **X** (rettangolo blu in basso). Il blocco può essere rimosso trascinandolo fuori dell'area dell'oggetto. Analogamente, è possibile applicare un blocco a qualunque oggetto: selezionate l'icona **Fixate**, presente nella *Toolbar*, e cliccate sull'oggetto che volete bloccare.
- Dopo essersi assicurati che è selezionata l'icona *Brush* della *Toolbar*, cliccate su uno dei rettangoli e trascinateli a *sinistra / destra / alto / basso*. Vi accorgete che potete posizionare un qualunque oggetto in una qualunque posizione. Nel caso in cui uno degli oggetti sia posizionato in modo anomalo (rettangolo giallo, nella figura qui in basso) quando verrà premuto il tasto *Play* lo vedrete rimbalzare sul rettangolo rosso, simulando perfettamente la realtà.



- Se la traiettoria di un oggetto non incontra un ostacolo, l'oggetto precipita indefinitamente.

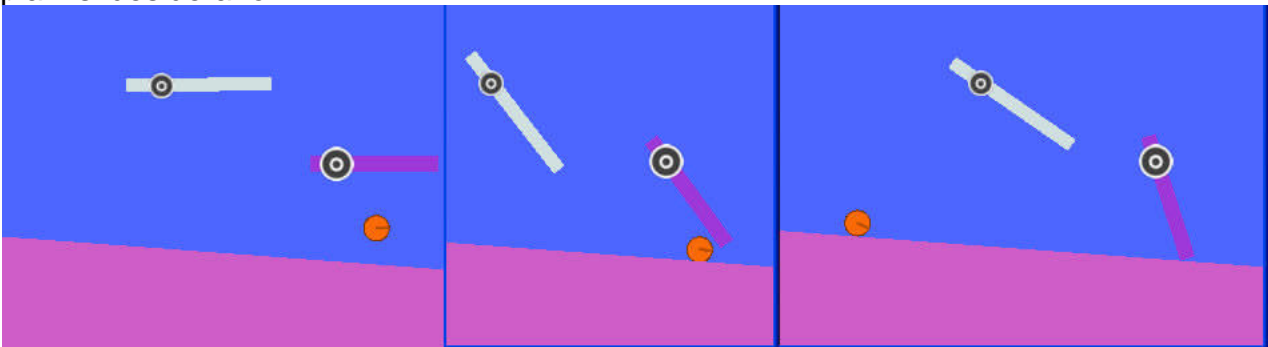
Prima di proseguire con gli altri scenari, effettuate le seguenti prove dopo aver ripristinato, volta per volta, la situazione iniziale (*Redo sim start*):

- Applicate / rimuovete il blocco "**X**" (*Fixate*) da uno o più rettangoli.
- Spostate i rettangoli a destra / sinistra / alto / basso prima di premere *Play*.

Secondo scenario: *B_Barre_mobili.phn*

In questo caso è stato introdotto un piano di fondo, di colore rosa. A proposito, dopo aver acquisito esperienza con lo scenario *B_Barre_mobili.phn* potrete sperimentare la creazione di piani: per tale scopo, dopo aver selezionato l'icona **Plane** della *Toolbar*, cliccate con tasto sinistro del mouse in un punto libero qualunque del cielo, quindi

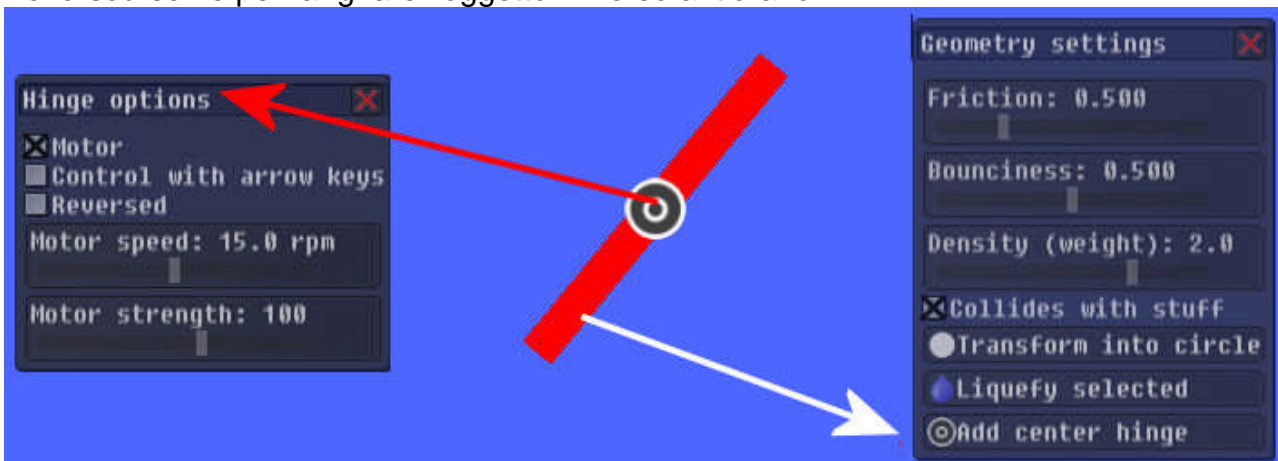
trascinate il puntatore in una delle direzioni possibili fino a che il piano non risulterà posizionato (e/o inclinato) come desiderato. Per spostare un piano creato in precedenza occorre cliccarlo con tasto sinistro e trascinarlo. E' possibile creare e posizionare quanti piani si desiderano.



Tornando allo scenario *B_Barre_mobili.phn*, premendo il pulsante *Play* noterete che il rettangolo di sinistra (grigio) ruota attorno al perno (creato con l'icona **Hinge** della Toolbar, analogamente a come – nell'esempio precedente – si è creata la **X** del blocco con *Fixate*). Il rettangolo, grazie alla presenza della forza di gravità, oscillerà attorno al perno. Le oscillazioni si smorzano dopo un tempo determinato dal peso dell'oggetto, dalla posizione del perno (ovviamente non si verificheranno oscillazioni se il perno è fissato perfettamente al centro), dalla presenza della forza di gravità e dall'attrito dell'aria (se impostata apponendo il segno di spunta nella casella *Air friction* della finestra *Options*). Notare che il rettangolo di destra – o meglio la distanza tra il perno e il fondo – è di lunghezza superiore alla sua distanza dal fondo. Ne consegue che si bloccherà dopo averlo urtato. Il cerchio (creato con lo strumento *Circle* della Toolbar) inizierà a rotolare in direzione della pendenza ma, entrato in collisione con il secondo rettangolo, verrà sospinto in salita in quanto dotato di una capacità di *rimbalzo*. Dopo un po' di tempo, esaurita la spinta, il cerchio rotolerà nuovamente verso destra, fino a bloccarsi definitivamente contro il secondo rettangolo. L'oscillazione del primo rettangolo proseguirà fino allo smorzamento completo, che sarà più rapido se è presente l'attrito dell'aria.

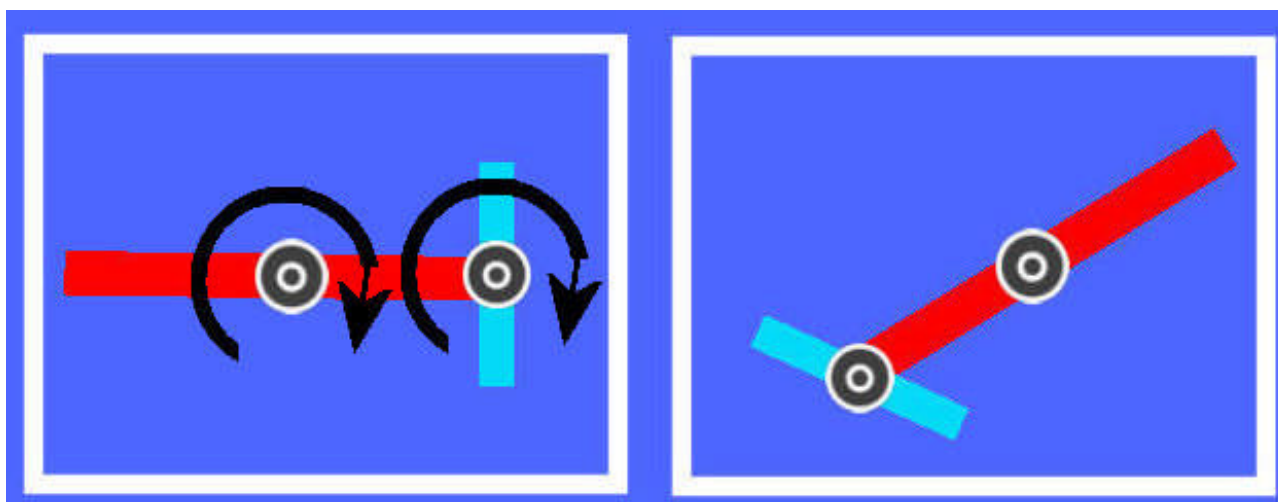
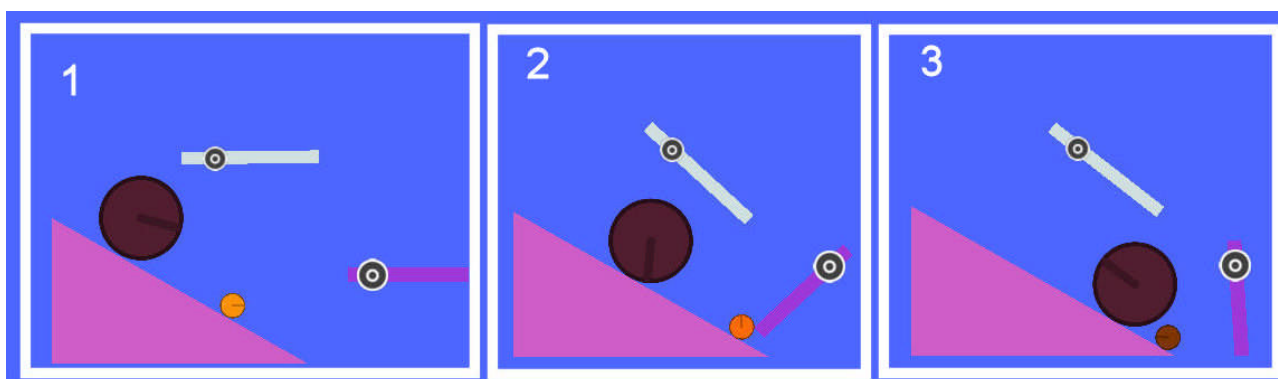
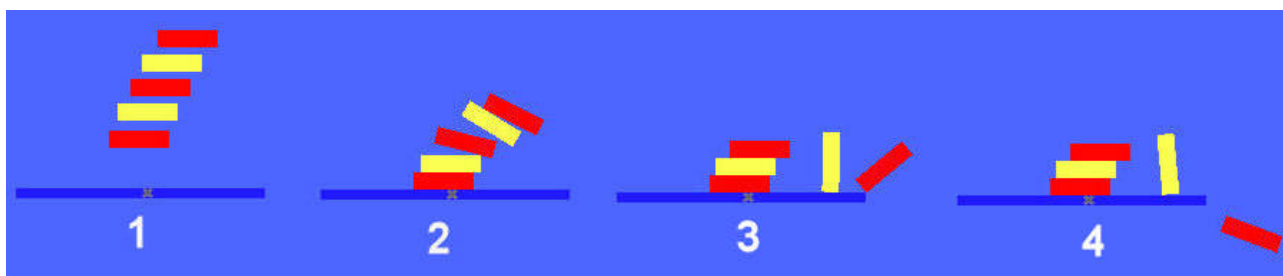
Terzo scenario: *C_Rotore.phn*

Un qualsiasi oggetto può essere fatto ruotare, in senso orario o antiorario, applicando il **motore** a un **cardine**. Lo scenario di *C_Rotore.phn* rappresenta un semplice rettangolo al quale (cliccando sopra con il tasto destro) è stato applicato un rotore esattamente al suo centro selezionando la voce *Add center hinge*. In seguito, cliccando ancora con il tasto destro sul cardine appena creato, è sufficiente apporre il segno di spunta sulla casella *Motor* e far scorrere il cursore *Motor speed* fino al numero di giri desiderato. La casella *Reversed* serve per far girare l'oggetto in verso antiorario.



Esercizi

Esercitatevi per vostro conto cercando di riprodurre le tre schermate che seguono. Per aiutarvi, vi dirò soltanto che ognuno dei tre scenari trae origine dagli scenari descritti.



Prossimamente...

Questo è solo il primo tutorial dedicato a **Phun**. Visitate frequentemente il mio sito (www.alessandrodesimone.net) per verificare la presenza di aggiornamenti o nuovi Tutorial.

Il presente Tutorial è stato modificato il giorno 1 marzo 2008
Per ulteriori aggiornamenti consultare il sito www.alessandrodesimone.net