

DEV.

La rivista che ti insegna a programmare



Main window e azioni astratte con Qt

Completamento automatico di una combobox

MSSQLServer

Appunti

Sicurezza

Commenti

Programmazione

DB

Programmare per PostgreSQL con libpq

Zope

Introduzione a Zope

Sicurezza

Buffer overflow e dintorni

Game DEV

Realizziamo un videogioco con Game Maker 6

JAVA

Sviluppo di applicazioni per la Televisione Digitale Terrestre in Java (Xlet)

Linux

Il modello Open Source

Porting

Da Java a .NET mediante J# e JLC



GRUPPO EDITORIALE
INFOMEDIA

THE SOFTWARE SIDE OF YOUR MIND

ISSN: 1124-5468

Sped. abb. post. 45% articolo 2 comma 20/B legge 662/96 - Filiale di Bologna

ISSN 1124-5468



5 0 1 3 0



9 771124 546002

Provato per voi

CA eTrust EZArmor 2005

di **Roberto Casarini**
rcasarini@infomedia.it

Open Source: un modello da imitare?

Anche quest'anno gli argomenti proposti alla Java Conference 2005 sono di notevole interesse. Partecipando a questo evento si prende atto delle reali potenzialità di un linguaggio ormai maturo e consolidato. Anche se la curiosità più grande è rivolta alle anticipazioni su OpenSolaris, di particolare interesse è la *parallela* che riguarda lo sviluppo di applicazioni interattive per la TV digitale (argomento che affrontiamo anche su questo numero di DEV). Immane una *parallela* dedicata all'Open Source, oggetto di studio persino per gli economisti: l'interesse e la necessità di acculturamento verso questo argomento è tale da aver indotto alcuni atenei italiani a promuovere Master universitari specifici. Molte società hanno fatto propria la filosofia dell'Open Source: soltanto per non essere inferiori ad altre? sono solo strategie di marketing?

C'è un gran fermento intorno all'Open Source, e non si riescono a determinare i limiti di sviluppo di questo approccio dal punto di vista informatico. In alcuni casi, l'Open Source rappresenta l'unica alternativa per poter essere competitivi: ma perché tutto ciò si sta evolvendo soltanto adesso?

In passato, chi produceva software custodiva il proprio codice sorgente con molta gelosia (forse per il timore di rivelare i propri segreti e/o imprecisioni?), oggi la tendenza è opposta: sarà una direzione obbligatoria per tutti?

Visto l'interesse per questo affascinante argomento, anche DEV non poteva astenersi dal parlarne, cercando di rispondere a queste e a tante altre domande: con questo numero, infatti, partirà una serie di articoli per affrontare l'argomento nel modo più esaustivo possibile, cercando di spiegarne la filosofia e fornirvi gli strumenti utili per poter partecipare a progetti Open Source (ovviamente come programmatori e non solo come filosofi).

L'entusiasmo verso questa filosofia è legato alla facilità con cui un programmatore esperto può partecipare attivamente allo sviluppo di un software, contribuendone alla realizzazione; così come si lascia una firma su di un libro aperto e accessibile a tutti.

La collaborazione tra più persone dà la possibilità di migliorare il codice e di correggere il maggior numero di possibili errori, per il semplice fatto che un programmatore può sbagliare, un milione no.

Da lettore non di professione non ho mai partecipato attivamente alle proposte lanciate dalle redazioni, e all'Università difficilmente alzavo la mano per esprimere un mio giudizio: "Professore, ci sarebbe una soluzione migliore di quella proposta!". Non perché non avessi mai avuto nulla da dire, ma semplicemente per pigrizia giovanile o per timore reverenziale nei confronti del "prof".

A ruoli invertiti, ho capito di aver perso un diritto importante: partecipare, esprimere e condividere un mio pensiero, che è poi il successo dell'Open Source.

Il successo del Web (ma anche dei vari "reality" in TV) è senza dubbio dovuto alla interattività degli utenti e alla facilità con la quale è possibile partecipare, decidere e criticare, mantenendo il completo anonimato. Concludo, quindi, rinnovando ai lettori l'invito a scrivermi (ringraziando tutti coloro che lo hanno già fatto!), dopotutto anche per DEV l'Open Source è un modello da imitare.

Roberto Casarini



Lo speciale di questo numero: MS SQL Server: appunti, concetti, programmazione, sicurezza

- 12 **SQL Server 2005 va in scena** di Andrea Benedetti
- 17 **Intervista a Michele Zoppi** di Natale Fino
- 20 **Microsoft SQL Server: un sistema sicuro?** di Lorenzo Braidi
- 24 **Stored Procedures in SQL Server** di Alan Lenisa
- 27 **SQL Server 2005 - Scrivere un aggregatore custom in VB** di Gian Maria Ricci

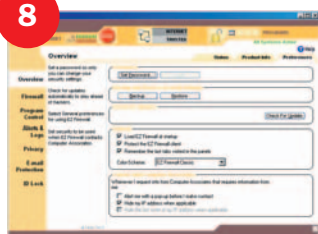
Programmazione

- 32 **Cellulari** - Il protocollo AT+ dei telefoni cellulari. Gestione della sicurezza e dei blocchi (PARTE 16) di Vito Vessia
- 37 **Database** - Programmare per PostgreSQL con libpq (PARTE 2) di Gabriele Bartolini
- 42 **Database** - ADO.NET, istruzioni per l'uso (PARTE 5) di Lorenzo Vandoni
- 44 **Java** - Maven: "intelligent projects" (PARTE 1) di Francesco Tinti
- 47 **Java** - Sviluppo di applicazioni per la Televisione Digitale Terrestre in Java (Xlet) di Fabio Ramacciani
- 55 **Porting** - Da Java a .NET mediante J# e JLCA di Stefano Franco
- 62 **Sicurezza** - Buffer overflow e dintorni (PARTE 2) di Gianluca Insolubile
- 66 **Visual Objects** - VO e XML (PARTE 1) di Fabio Cresta
- 69 **Zope** - Introduzione a Zope di Franco Gambato

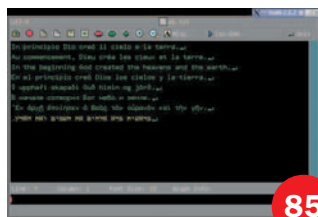
Rubriche

- 6 **Extended Mode** - Airborne Troops a cura di Stefano Arcidiacono
- 11 **CrackMe** - Timer Password: quando il tempismo è tutto: le soluzioni a cura di Matteo Baccan
- 59 **Normative** - La tutela dei testi in formato digitale (PARTE 2) di Massimo Farina
- 94 **Game DEV** - Realizziamo un videogioco con Game Maker 6 (PARTE 1) di Mauro Pesci
- 97 **SourceForge News** a cura di Gabriele Ganzi
- 98 **Horizon** - Nuovi confini della trasmissione dati di Luigi Morelli

8



85



94

Sei agli inizi? Non riesci a risolvere i primi problemi? Niente paura! Iscriviti alle mailing-list infomedia!

programmo_subito@infomedia.it
e linux-it@infomedia.it

I listati e i programmi di utilità che corredano gli articoli sono prelevabili gratuitamente dal sito <http://www.infomedia.it/DEV> oppure da <ftp://ftp.infomedia.it/pub/DEV/Listati/>





Numero 83

- 73 **Livello 2** La gestione del tempo: fissare appuntamenti via Internet di Alessandro Scoscia
- 76 **Livello 2** Main window e azioni astratte con Qt di Fabrizio Angius
- 80 **Livello 2** Completamento automatico di una combobox in Visual Basic .NET di Emanuele Mattei



Numero 62

- 82 **Livello 1** Il modello Open Source (PARTE 1) di Alessio Saltarin
- 85 **Livello 2** Editor Unicode e composizione di documenti Unicode con LaTeX di Gianluca Pignalberi
- 90 **Livello 2** Linux & Bluetooth (ULTIMA PARTE) di Stefano Sanna

In prova

CA eTrust EZArmor 2005

di Marco Barontini

8

Ai lettori

Low Level a cura di Matteo Baccan
Una porta aperta in ASPNuke

23

J2ME World a cura di Stefano Sanna
Dove sono i PDA?

26

Doktor Morb a cura di Luigi Morelli
Sintomi

36

E-ducational a cura di Luigi Morelli
Acronimi, che passione!

54

Free Reading a cura di Lorenzo Vandoni
Aspect-Oriented Programming

58

Italian Free a cura di Gianluca Masina
Spider Chess

68



Numero 130
Giugno 2005
Anno XIII

Direttore Responsabile

Marialetizia Mari (mmari@infomedia.it)

Coordinamento editoriale

Roberto Casarini (rcasarini@infomedia.it)

Collaboratori

Fabrizio Angius, Stefano Arcidiacono,
Matteo Baccan, Marco Barontini,
Gabriele Bartolini, Andrea Benedetti,
Lorenzo Braidi, Fabio Cresta,
Paolo De Nictolis, Massimo Farina,
Stefano Franco, Franco Gambato,
Gianluca Insolubile, Alan Lenisa,
Emanuele Mattei, Luigi Morelli,
Mauro Pesci, Gianluca Pignalberi,
Lino Polo, Fabio Ramacciani,
Gian Maria Ricci, Alessio Saltarin,
Stefano Sanna, Alessandro Scoscia,
Francesco Tinti, Lorenzo Vandoni,
Vito Vessia

Grafica

Manola Greco (mgreco@infomedia.it)

Immaginazione e Copertina

ADM Studio - Brughiero (MI)



Gruppo Editoriale Infomedia S.r.l.
Via Valdera P. 116 - 56038 Ponsacco (PI)
Tel. 0587 73 64 60 (r.a.)
Fax 0587 73 22 32
www.infomedia.it - red_dev@infomedia.it

Direzione

Natale Fino (nfino@infomedia.it)

Divisione Libri

Lisa Vanni (book@infomedia.it)

Marketing & Advertising

Tel. 0587 73 64 60
marketing@infomedia.it

Supporto Sistemistico

webmaster@infomedia.it

Amministrazione

Sara Mattei (amministrazione@infomedia.it)

Segreteria

Enrica Nassi (info@infomedia.it)

Stampa

Betagraf Spa - Funo di Argelato (BO)

Distribuzione

Parrini & C. (Roma)

Ufficio Abbonamenti

Tel. 0587 73 64 60
Fax 0587 73 22 32
abbonamenti@infomedia.it
online.infomedia.it

Si prega di inviare i comunicati stampa e gli inviti stampa per la redazione all'indirizzo: comunicatistampa@infomedia.it

La riproduzione con qualsiasi mezzo di illustrazioni e di articoli pubblicati sulla rivista, nonché la loro traduzione, è riservata e non può avvenire senza espressa autorizzazione dell'editore. I manoscritti e le illustrazioni inviati alla redazione non saranno restituiti anche se non pubblicati e l'editore non si assume alcuna responsabilità per il caso che si tratti di esemplari unici.

"DEV. Developing Software & Solutions"
è una rivista del Gruppo Editoriale Infomedia S.r.l.
Direzione e Amministrazione:
Via Valdera P. 116 - Ponsacco
Registrazione presso il Tribunale di Pisa
n. 21 del 25.11.1999

Contenuto pubblicitario inferiore al 45%

di **Roberto Casarini**
rcasarini@infomedia.it

Open Source: un modello da imitare?

Anche quest'anno gli argomenti proposti alla Java Conference 2005 sono di notevole interesse. Partecipando a questo evento si prende atto delle reali potenzialità di un linguaggio ormai maturo e consolidato. Anche se la curiosità più grande è rivolta alle anticipazioni su OpenSolaris, di particolare interesse è la *parallela* che riguarda lo sviluppo di applicazioni interattive per la TV digitale (argomento che affrontiamo anche su questo numero di DEV). Immane una *parallela* dedicata all'Open Source, oggetto di studio persino per gli economisti: l'interesse e la necessità di acculturamento verso questo argomento è tale da aver indotto alcuni atenei italiani a promuovere Master universitari specifici. Molte società hanno fatto propria la filosofia dell'Open Source: soltanto per non essere inferiori ad altre? sono solo strategie di marketing?

C'è un gran fermento intorno all'Open Source, e non si riescono a determinare i limiti di sviluppo di questo approccio dal punto di vista informatico. In alcuni casi, l'Open Source rappresenta l'unica alternativa per poter essere competitivi: ma perché tutto ciò si sta evolvendo soltanto adesso?

In passato, chi produceva software custodiva il proprio codice sorgente con molta gelosia (forse per il timore di rivelare i propri segreti e/o imprecisioni?), oggi la tendenza è opposta: sarà una direzione obbligatoria per tutti?

Visto l'interesse per questo affascinante argomento, anche DEV non poteva astenersi dal parlarne, cercando di rispondere a queste e a tante altre domande: con questo numero, infatti, partirà una serie di articoli per affrontare l'argomento nel modo più esaustivo possibile, cercando di spiegarne la filosofia e fornirvi gli strumenti utili per poter partecipare a progetti Open Source (ovviamente come programmatori e non solo come filosofi).

L'entusiasmo verso questa filosofia è legato alla facilità con cui un programmatore esperto può partecipare attivamente allo sviluppo di un software, contribuendone alla realizzazione; così come si lascia una firma su di un libro aperto e accessibile a tutti.

La collaborazione tra più persone dà la possibilità di migliorare il codice e di correggere il maggior numero di possibili errori, per il semplice fatto che un programmatore può sbagliare, un milione no.

Da lettore non di professione non ho mai partecipato attivamente alle proposte lanciate dalle redazioni, e all'Università difficilmente alzavo la mano per esprimere un mio giudizio: "Professore, ci sarebbe una soluzione migliore di quella proposta!". Non perché non avessi mai avuto nulla da dire, ma semplicemente per pigrizia giovanile o per timore reverenziale nei confronti del "prof".

A ruoli invertiti, ho capito di aver perso un diritto importante: partecipare, esprimere e condividere un mio pensiero, che è poi il successo dell'Open Source.

Il successo del Web (ma anche dei vari "reality" in TV) è senza dubbio dovuto alla interattività degli utenti e alla facilità con la quale è possibile partecipare, decidere e criticare, mantenendo il completo anonimato. Concludo, quindi, rinnovando ai lettori l'invito a scrivermi (ringraziando tutti coloro che lo hanno già fatto!), dopotutto anche per DEV l'Open Source è un modello da imitare.

Roberto Casarini

Realizziamo un videogioco con **Game Maker 6**

PARTE 1

Lo scopo della serie di articoli alla quale diamo inizio con questo numero è quello di produrre un videogioco completo e distribuibile mettendo alla prova le capacità di questo ambiente. Il presente articolo è stato dedicato alla realizzazione di una mini analisi per definirne tutte le caratteristiche

Nel numero precedente è stata presentata una breve visione d'insieme di Game Maker 6 cercando di esporre nel modo più conciso possibile gli argomenti necessari alla realizzazione di un videogioco. Si è partiti dal concetto di *stanza*, paragonabile ad un livello di gioco, definendola come l'area all'interno della quale prendono vita gli *oggetti*. Questi ultimi rappresentano il pilastro principale dello sviluppo in Game Maker in quanto entità animate. Essi sono in grado di rispondere opportunamente agli eventi di gioco effettuando una serie prestabilita di azioni.

E' stato possibile capire in che modo l'interfaccia Drag&Drop è in grado di semplificare ed accelerare tutte le attività di sviluppo. Essa offre, ad esempio, la possibilità di programmare la logica degli *oggetti* con semplici clic e trascinamenti di icone. L'argomento finale era costituito da una breve introduzione al linguaggio GML il quale presenta una struttura molto simile a quella del C.

Nella serie di articoli che seguiranno approfondiremo lo studio di questo ambiente di sviluppo in ogni suo aspetto realizzando un videogioco completo e facilmente distribuibile. Otterremo un unico file eseguibile (.EXE) in grado di funzionare senza l'ausilio di altri file o DLL esterne da installare prima del suo utilizzo.

Introduzione ad una breve analisi

Questo articolo, come annunciato già dall'introduzione, anche se poco tecnico e molto discorsivo ci aiuterà a capire cosa realizzeremo, quali sono i personaggi del gioco e quali sono gli eventi che possono accadere. Dato che un videogioco nasce sempre e solo da un'idea è sviluppandola che si arriva a definire l'obiettivo si vuole raggiungere. Purtroppo, però, l'esiguo spazio a disposizione ci obbliga ad accontentarci di un gioco molto semplice nella sua struttura. Scegliamo così di riprendere lo schema di un classico che ha fatto storia nelle sale giochi degli anni '70-'80, il famoso 'Asteroids'.

Prima di provare delusione per via della soluzione scelta tentiamo di sfoderare la nostra immaginazione considerando le possibilità offerte dall'ambiente. Pensiamo a *sprite* e *sfondi* realizzati con programmi di grafica 3D, ad animazioni fluide, a *suoni* proiettati in un ambiente tridimensionale. Bene! In **Figura 1** potete osservare una possibile interpretazione di questa idea.

Il passo successivo è organizzare il lavoro realizzando una schematizzazione della struttura generale del progetto definendone le varie componenti. Questa è una fase che interviene nella sua riuscita in maniera determinante ed è la ragione per cui i team di sviluppo dedicano tanto tempo a pianificare ed organizzare il lavoro.

Il gioco

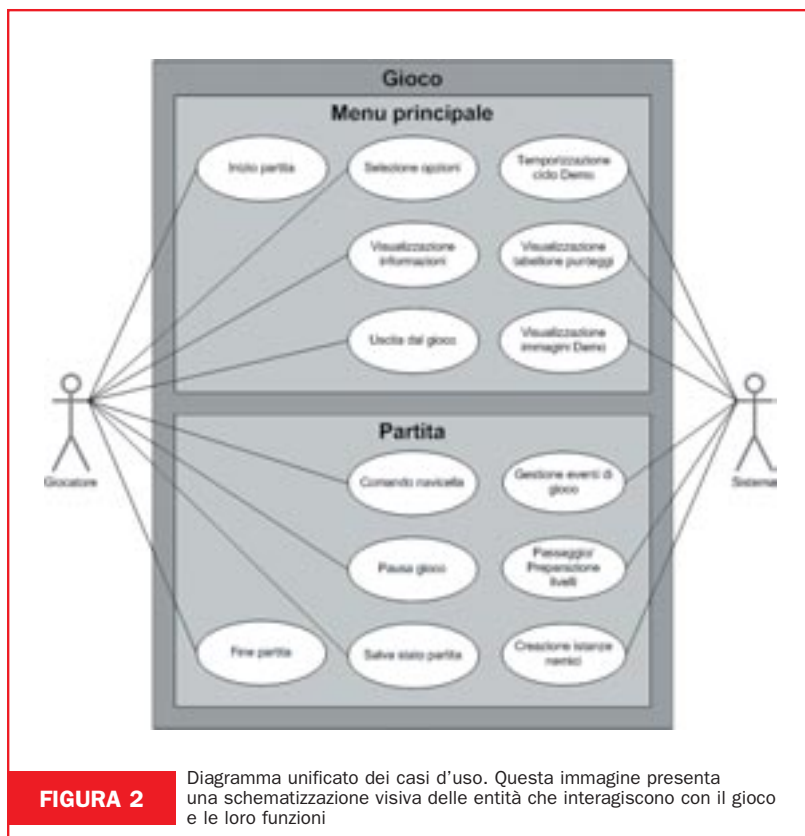
Iniziamo, quindi, col focalizzare quali sono gli aspetti essenziali del nostro progetto. E' bene ricordare che 'Asteroids' è un videogioco in grafica vettoriale ambientato nello spazio dove un'esule astronave è alle prese con decine di asteroidi vaganti. L'astronave ha a disposizione un arma da fuoco che colpendo questi ultimi ne provoca la distruzione. Quando esplodono, gli asteroidi, si suddividono in parti più piccole fino a che, raggiunta la dimensione minima, non scompaiono del tutto. Ogni tanto appare dal nulla un'altra astronave che spara all'impazzata nel tentativo di abbattere la prima. Il giocatore ha, ovviamente, la possibilità di spostare l'astronave tramite un reattore posto nella parte posteriore di questa. In più, nelle situazioni drammatiche, si può fare ricorso al salto iperspaziale che trasla istantaneamente la navicella in un punto dello spazio più sicuro.

Dato che copiare di pari passo un gioco già esistente non è il massimo a cui aspiriamo, apporteremo alcune modifiche



FIGURA 1

Un'immagine ripresa dal prototipo del gioco che realizzeremo



all'idea originale cercando di renderlo più avvincente. Si potrebbe, infatti, fare in modo che la nostra astronave raccolga dei bonus che potenzino l'armamento originale, equipaggiarla di testate nucleari con un'enorme capacità distruttiva o, anche, dotare gli asteroidi di una certa resistenza in modo che non vengano distrutti al primo colpo. Un tocco di professionalità è rappresentato dalla presenza di un menu principale dal quale il giocatore può effettuare le sue scelte ed un tabellone che mantiene i punteggi migliori e i nomi dei relativi giocatori.

E' importante definire ogni aspetto del gioco per avere un'idea chiara di cosa si vuole sviluppare e se il progetto risulta fattibile

Organizzazione del progetto

Abbiamo già individuato molte caratteristiche del nostro gioco le quali possono essere suddivise in due sezioni:

- Il menu principale
- La partita

La **Figura 2** espone una schematizzazione esaustiva delle funzionalità di ogni sezione identificando anche le entità che interagiscono con il gioco suddividendone responsabi-

lità e compiti. Queste sono essenzialmente due, il *giocatore* e il meccanismo che regola il susseguirsi degli eventi che chiameremo *sistema*.

Il *giocatore* ha la possibilità di selezionare una voce dal menu principale. Può iniziare una nuova partita, cambiare le impostazioni di gioco, visualizzare le informazioni ad esso relative o uscire. Il *sistema*, invece, ha il compito di regolare i tempi di visualizzazione delle varie schermate nel menu principale.

Durante la partita il *giocatore* può manovrare l'astronave, fare fuoco, saltare nell'iperspazio e catturare i bonus. Il *sistema* si occuperà di organizzare i livelli impostando i parametri necessari, creare le istanze dei nemici e gestire i vari eventi, quali:

- Inizio partita
- Gioco normale
- Fine livello
- Morte giocatore
- Fine partita

Tutte queste situazioni di gioco sono chiamate *stati* e possono essere ben schematizzate dal grafico in **Figura 3**. Come si può vedere quando ci troviamo nel menu principale il *sistema* ci permette solamente di selezionare una delle voci di menu oppure attendere e vedere le altre schermate che vengono ciclata a tempi regolari.

Nella fase di gioco si transita inizialmente nello *stato* di "Inizio partita". Dopo che questo evento si è verificato e tutte le operazioni di inizializzazione sono state portate a termine si passa allo *stato* chiamato "Gioco". In questa situazione vengono svolti tutti i compiti, di cui si è parlato sopra, necessari a mandare avanti la partita.

Durante una partita può succedere è che il giocatore venga distrutto allorché il *sistema* controllerà la disponibilità di altre vite e in caso contrario darà fine alla partita transitando nell'apposito *stato*. Se giocatore ha a disposizione altre astronavi la partita continua regolarmente.

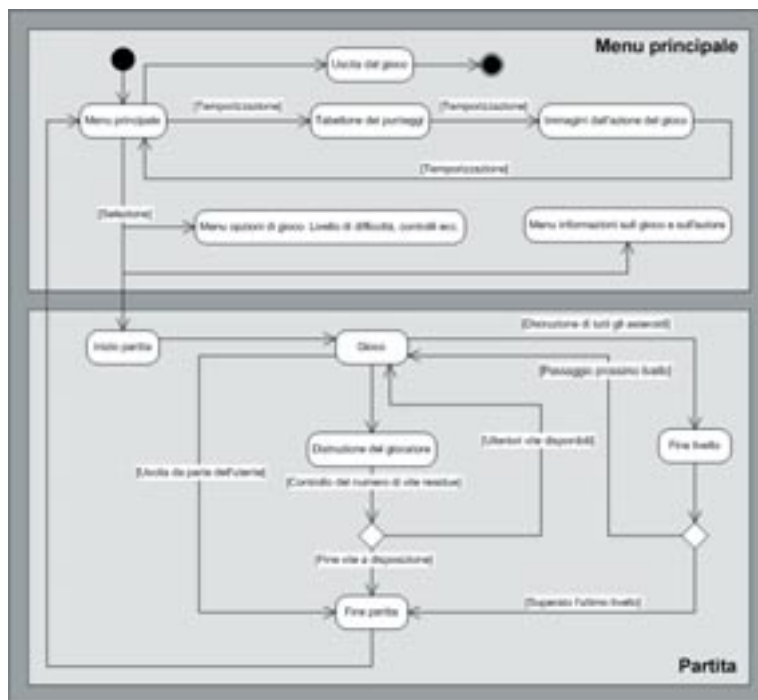
Ogni qualvolta, invece, che viene distrutto un asteroide il *sistema* controlla se ce ne sono altri ancora in giro per lo schermo. Qualora risulti che siano stati tutti distrutti provvederà ad impostare il livello successivo, sempre che non sia stato superato l'ultimo, nel qual caso darà comunque fine alla partita dopo aver visualizzato una breve sequenza di congratulazioni.

Analizziamo, ora, nel dettaglio gli oggetti che prenderanno parte al gioco; ne prevediamo 4:

- Il giocatore
- Gli asteroidi
- La navicella nemica
- I bonus

Il giocatore

Il giocatore è rappresentato, come già detto, da un'astronave che, nello spazio, può ruotare su se stessa tramite propulsori laterali, essere sospinta in avanti tramite il reattore


FIGURA 3

Questo diagramma riassume gli eventi di gioco. Quando si transita da uno stato ad un altro il corso della partita viene modificato

posteriore, saltare nell'iperspazio, sparare agli asteroidi e far esplodere le poche testate nucleari a disposizione. In più aggiungiamo i livelli energetici della vitalità, dei motori e del salto.

Le entità che interagiscono con il gioco sono il "giocatore" e il "sistema"

La navicella è sensibile, ovviamente, alle collisioni con gli asteroidi e al fuoco della navicella nemica. Questi provocano la perdita di energia in proporzione alla loro dimensione o alla potenza del fuoco. Quando l'energia vitale raggiunge lo zero l'astronave esplode ed una vita viene detratta da quelle a disposizione. Anche la potenza dei motori e la disponibilità del salto iperspaziale sono condizionati dal relativo livello di energia.

Come facilitazione aggiungiamo che l'energia aumenti leggermente ad intervalli di tempo ben definiti.

Gli asteroidi

Ogni asteroide possiede un moto determinato da direzione, velocità di spostamento e velocità di rotazione su se stesso. Anche per questi prevediamo dei livelli energetici che permettono di definirne la resistenza al fuoco del giocatore. Quando un asteroide esaurisce la propria energia esplode liberando alcuni frammenti più piccoli del primo. Questo accade fino a che i frammenti raggiungono una dimensione minima, dopodiché scompaiono definitivamente.

Nei livelli avanzati gli asteroidi più piccoli hanno un effetto magnetico e vengono, quindi, attratti dalla nave del giocatore rendendo più difficile il gioco.

La nave nemica

La nave nemica fa parte di una civiltà con tecnologia superiore ed ha a disposizione un numero illimitato di salti nell'iperspazio, non possiede moto ma spara in diverse direzioni secondo la posizione del giocatore. Decidiamo che essendo difficile da colpire non possieda resistenza al fuoco del giocatore, cioè distrutta all'istante quando colpita.

I bonus

Per rendere più interessante la partita faremo in modo che il giocatore possa rigenerare i propri livelli energetici, aumentare la potenza di fuoco oppure aggiungere punteggio al quantitativo già posseduto in modo da guadagnare ulteriori vite. Le testate nucleari sono in numero fissato durante la partita e non possono essere aggiunte delle altre. I bonus di energia e punteggio provengono dall'interno di alcuni asteroidi, mentre gli armamenti potranno essere conquistati distruggendo una nave nemica.

Il mostro finale

Nell'ultimo livello il giocatore dovrà affrontare l'asteroide padre. Questo risulterà molto difficile da abbattere poiché sarà molto resistente ai colpi e ad ognuno libererà una grande quantità di frammenti pericolosi.

Anticipazioni sul prossimo articolo

Abbiamo raggiunto una visione pressoché completa del nostro progetto e dal prossimo articolo daremo inizio al lavoro vero e proprio di sviluppo utilizzando gli *sprite*, i *suoni*, i *glif*. Vedremo come preparare l'ambiente, come funziona lo *'Sprite Editor'* per organizzare i fotogrammi delle nostre animazioni e l'*'Image Editor'* per eventuali ritocchi.

Bibliografia

- [1] John P. Flynt, "Software Engineering for Game Developers", Thomson, Anno 2005, Codice ISBN 1592001556
- [2] Luca Vetti Tagliati, "UML e ingegneria del software: dalla teoria alla pratica", Hops, Anno 2003, ISBN 8883780698
- [3] Mark Overmars, "Designing Games with Game Maker", www.gamemaker.nl

Mauro Pesci

Si occupa di Analisi e Sviluppo del software in vari ambienti e piattaforme. Ha lavorato per diverse aziende nel settore, mentre ora è impegnato in un progetto presso la HP di Roma. Attualmente studia Informatica presso l'università di Tor Vergata (Roma).