

# DEU.

La rivista che ti insegna a programmare

# Il codice della privacy

## ASP

Compilare un'applicazione ASP.NET a linea di comando

## DB

Visibilità e profilazione con Microsoft SQL Server

## IA

Condividere la conoscenza tramite ontologie

## Linux

- Linux & Bluetooth
- Strumenti per la Programmazione Open Source

## Sicurezza

Buffer overflow e dintorni

**Provati per voi**

NINA Alloy Network Inventory  
Visual Web Developer 2005  
Express Edition

**Programmo**  
**Subito**

Utilizzare Dll C++ in Visual Basic

Qt Template Library  
e gestione dell'I/O con Qt



**Linux:** • Autopackage, il nuovo sistema 'standard' per le installazioni?  
• Le mille distribuzioni di Linux

**JAVA:** Maven: "intelligent projects"

**Web:** Green Orange Framework

di **Roberto Casarini**  
[rcasarini@infomedia.it](mailto:rcasarini@infomedia.it)

## Quale linguaggio di programmazione?

**I**l numero dei linguaggi di programmazione cresce così come gli informatici, allo stesso modo i computer si moltiplicano contemporaneamente agli utenti ed agli sviluppatori, l'unico aspetto che sembra essere in controtendenza è il tempo a disposizione, sempre piuttosto scarso soprattutto per fare le cose nel modo giusto.

Facendo una ricerca in rete, inserendo la parola "framework" nel motore di ricerca ci accorgiamo dell'alta quantità di informazioni disponibili su questo termine.

Banalmente, primo tra tutti è il .NET, e dopo alcune pagine, che forse "abusano" del termine, troviamo numerose soluzioni tra le quali alcune anche particolarmente innovative.

Se nella nostra ricerca aggiungiamo "language" e "programming" si presenta uno scenario più specifico che tuttavia sottintende quanto utopico possa essere per un informatico conoscere ed utilizzare tutti i potenziali "strumenti" del proprio mestiere.

Nel numero di questo mese, l'analisi svolta nell'articolo "Strumenti per la programmazione open source" è significativa e senza dubbio presenta un fetta di mercato considerevole anche se unicamente riferita alla comunità del software libero.

L'incremento concerne anche i linguaggi cosiddetti accessori, cioè i linguaggi di scripting di ultima generazione (Perl, Tcl, PHP, ecc.). Nati per semplificare la vita ai programmatori e per l'esecuzione di task di controllo, sono interessati da un costante trend positivo grazie soprattutto ai neofiti, spesso privi di qualsiasi nozione sulle basi della programmazione.

Quanti linguaggi dovrebbe invece conoscere un buon professionista?

A mio avviso è necessaria una giusta dose di elasticità, come in tutte le cose del resto. Un paio di settimane fa uscendo con alcuni colleghi mi sono imbattuto in un programmatore con molti anni di esperienza alle spalle, il quale sosteneva la supremazia assoluta del C, snobbando Java e C# perché a suo parere questi linguaggi rappresentano strumenti esclusivi dei neofiti. La mia conclusione in merito alla questione è stata piuttosto semplice: ogni problema ha la sua soluzione e nessuna soluzione potrà dare risposta a tutti i problemi. Infatti, ritengo che un elemento che può contribuire alla preparazione di un buon programmatore è la conoscenza di un buon numero di linguaggi. Tuttavia, realisticamente parlando, il programmatore si trova per lo più ad utilizzare ciò di cui necessita, tralasciando una conoscenza generica in favore di una specialistica.

In determinate situazioni, rimanere aggiornati può rappresentare una difficoltà: ad esempio se si lavora in un'azienda non particolarmente favorevole alle innovazioni ed ai cambiamenti, e che quindi tende, per così dire, a sostenere una politica interna "conservatrice" rispetto ai nuovi strumenti offerti dal mercato. Ma in questi casi non c'è da preoccuparsi, una panacea a tutti i mali non esiste, ma un aiuto sempre a portata di mano sì: non state forse leggendo DEV?!

Battute a parte, vorrei effettuare un mini sondaggio estivo per capire quale linguaggio, ambiente di sviluppo o piattaforma state utilizzando attualmente per lavoro e soprattutto quale vorreste eventualmente impiegare o conoscere. Attendo le vostre mail! Augurandovi una felice estate, vi lascio alla lettura di questo numero, stavolta forse un po' più vacanziera e rilassata del solito.

*Roberto Casarini*



### *Lo speciale di questo numero:* **Il codice della privacy**

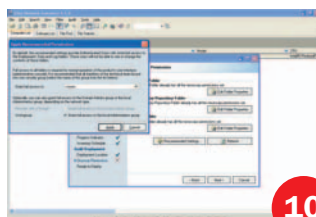
- 19 **Linee fondamentali del Codice Privacy** di Massimo Farina
- 23 **Misure minime di sicurezza: aspetti pratici**  
di Massimo Farina e Stefano Sanna
- 28 **Privacy e Videosorveglianza** di Massimo Farina
- 30 **Quali altri modi per intendere la Privacy?**  
di Massimo Farina

### Programmazione

- 32 **ASP .NET** - Compilare un'applicazione ASP.NET a linea di comando di Gabriele Bartolini
- 34 **Cellulari** - Il protocollo AT+ dei telefoni cellulari. Classe di gestione delle rubriche (PARTE 17)  
di Vito Vessia
- 38 **C#** - .NET e remoting di Gian Maria Ricci
- 41 **C#** - Utilizzare le API di shell in C# di Gian Maria Ricci
- 44 **Database** - Visibilità e profilazione con Microsoft SQL Server  
di Lorenzo Braidì
- 48 **Networking** - Grid Computing di Angelo Raffaele Bruno
- 50 **Mainframe** - L'evoluzione tecnologica nel mondo bancario e finanziario: dal mainframe al browser web di Stefano Corti
- 55 **Intelligenza Artificiale** - Condividere la conoscenza tramite ontologie di Paolo Besana
- 58 **Sicurezza** - Buffer overflow e dintorni (PARTE 3)  
di Gianluca Insolubile
- 62 **Web** - Green Orange Framework  
di Massimiliano Arcidiacono, Giuseppe Buscemi, Massimo Mancino
- 65 **Java** - Maven: "intelligent projects" (PARTE 2) di Francesco Tinti

### Rubriche

- 6 **Extended Mode** - EVE Online  
a cura di Stefano Arcidiacono
- 8 **Spazio Libri** - a cura di Marco Iannaccone
- 13 **CrackMe** - RegMe: registrare un software già registrato: soluzioni a cura di Matteo Baccan
- 92 **Game DEV** - Realizziamo un videogioco con Game Maker 6 (PARTE 2) di Mauro Pesci
- 98 **Horizon** - La tigre di carta di Luigi Morelli



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 84

- 70 **Livello 2** Qt Template Library e gestione dell'I/O con Qt  
di Fabrizio Angius
- 75 **Livello 2** Utilizzare Dll C++ in Visual Basic  
di Gian Maria Ricci



Numero 63

- 77 **Livello 1** Le mille distribuzioni di Linux (PARTE 1) di Corrado Ignotti
- 80 **Livello 1** Strumenti per la programmazione Open Source  
(PARTE 2) di Alessio Saltarin
- 84 **Livello 2** Autopackage, il nuovo sistema 'standard'  
per le installazioni? di Marco Iannaccone
- 87 **Livello 2** Linux & Bluetooth (ULTIMA PARTE)  
di Stefano Sanna

## In prova

### Alloy Network Inventory 4.2

di Paolo De Nictolis

10

### Microsoft Visual Web Developer 2005 Express Edition Beta 2

di Paolo De Nictolis

14

### Tempest

di Manuel Leone

17

## Ai lettori

**Italian Free** a cura di Gianluca Masina  
Njabe

7

**Low Level** a cura di Matteo Baccan  
Browser: gioia e delizia

54

**E-ducational** a cura di Luigi Morelli  
Wikipedia

61

**Free Reading** a cura di Lorenzo Vandoni  
Model Driven Architecture

79

**J2ME World** a cura di Stefano Sanna  
MIDP 3.0

83

**Doktor Morb** a cura di Luigi Morelli  
Il pentacolo

90



Numero 131  
Luglio/Agosto 2005  
Anno XIII

### Direttore Responsabile

Marialetizia Mari ([mmari@infomedia.it](mailto:mmari@infomedia.it))

### Coordinamento editoriale

Roberto Casarini ([rcasarini@infomedia.it](mailto:rcasarini@infomedia.it))

### Collaboratori

Fabrizio Angius, Stefano Arcidiacono,  
Massimiliano Arcidiacono, Matteo Baccan,  
Paolo Besana, Lorenzo Braidì,  
Angelo Raffaele Bruno, Giuseppe Buscemi,  
Stefano Corti, Paolo De Nictolis,  
Massimo Farina, Marco Iannaccone,  
Corrado Ignotti, Gianluca Insolubile,  
Manuel Leone, Massimo Mancino,  
Gianluca Masina, Luigi Morelli,  
Mauro Pesci, Gian Maria Ricci,  
Alessio Saltarin, Stefano Sanna,  
Francesco Tinti, Lorenzo Vandoni,  
Vito Vessia

### Grafica

Manola Greco ([mgreco@infomedia.it](mailto:mgreco@infomedia.it))

### Impaginazione 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](http://www.infomedia.it) - [red\\_dev@infomedia.it](mailto:red_dev@infomedia.it)

### Direzione

Natale Fino ([nfino@infomedia.it](mailto:nfino@infomedia.it))

### Divisione Libri

Lisa Vanni ([book@infomedia.it](mailto:book@infomedia.it))

### Marketing & Advertising

Tel. 0587 73 64 60  
[marketing@infomedia.it](mailto:marketing@infomedia.it)

### Supporto Sistemistico

[webmaster@infomedia.it](mailto:webmaster@infomedia.it)

### Amministrazione

Sara Mattei ([amministrazione@infomedia.it](mailto:amministrazione@infomedia.it))

### Segreteria

Enrica Nassi ([info@infomedia.it](mailto:info@infomedia.it))

### Stampa

Betagraf Spa - Fano di Argelato (BO)

### Distribuzione

Parrini & C. (Roma)

### Ufficio Abbonamenti

Tel. 0587 73 64 60  
Fax 0587 73 22 32  
[abbonamenti@infomedia.it](mailto:abbonamenti@infomedia.it)  
[online.infomedia.it](http://online.infomedia.it)

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

# Realizziamo un videogioco con **Game Maker 6**

## PARTE 2

*Come promesso, con questo appuntamento, inizieremo la realizzazione vera e propria del videogioco proposto nel precedente articolo. Avremo, così, la possibilità di conoscere in maniera più approfondita questo fantastico tool arrivato proprio in questi giorni alla versione 6.1*

**D**opo la trattazione puramente discorsiva riguardo al nostro progetto siamo finalmente pronti per affrontare argomenti più interessanti. Avendo determinato, infatti, in via definitiva quali entità prenderanno parte al videogioco possiamo iniziare a preparare l'ambiente ed inserire tutte le risorse che saranno necessarie nel corso dello sviluppo. Ovviamente mi sto riferendo agli elementi grafici e sonori ossia *sprite*, *sfondi* e *suoni*. Avremo la possibilità di conoscere nel dettaglio gli editor che Game Maker ci mette a disposizione per gestire gli *sprite*, le animazioni, le immagini, i parametri di gioco e molto altro.

### L'ambiente Game Maker

A prima vista l'ambiente sembra privo delle innumerevoli funzionalità necessarie alla realizzazione di un videogioco complesso e funzionale, ma come si suol dire, l'apparenza è ingannevole. L'immagine in **Figura 1** mostra l'aspetto della finestra principale. Iniziamo a vedere, quindi, la barra dei menu dove troviamo il classico chiamato *File*. Nel menu oltre alle funzioni per l'apertura, il salvataggio e la compilazione dei progetti, è presente la voce *Advanced Mode*. Questa funzione particolare permette all'utente di scegliere la modalità in cui lavorare con Game Maker: la modalità *Advanced*, appunto, e la *Simple*. La differenza tra le due è costituita dall'insieme di funzionalità che si possono utilizzare nel progetto. Nella *Simple*, infatti, non si possono usare i *font*, definire percorsi predefiniti per gli oggetti o realizzare gli script in GML. Una lista completa delle funzionalità presenti nelle due modalità di utilizzo viene presentata nel **Riquadro 1**. Procedendo oltre troviamo il menu *Edit* che permette di effettuare operazioni sulle entità di inserimento, duplicazione, cancellazione, rinomina e visualizzazione delle proprietà. Il menu *Add* semplicemente permette di creare una nuova entità in una delle categorie indicate nello stesso **Riquadro 1**. Troviamo, poi, il menu *Scripts* presente solo in modalità avanzata, il quale permette di effettuare

alcune operazioni sugli script che vedremo in seguito. Rimangono il menu *Run*, nel quale sono contenute le funzioni per lanciare il gioco in modalità normale o in debug, il menu *Window* che serve per organizzare le finestre sullo schermo e, infine, il menu *Help* dove è possibile trovare informazioni riguardanti Game Maker e una guida di riferimento completa.

Nella barra orizzontale sottostante si trova un insieme delle funzioni di più comune utilizzo presenti nei menu.

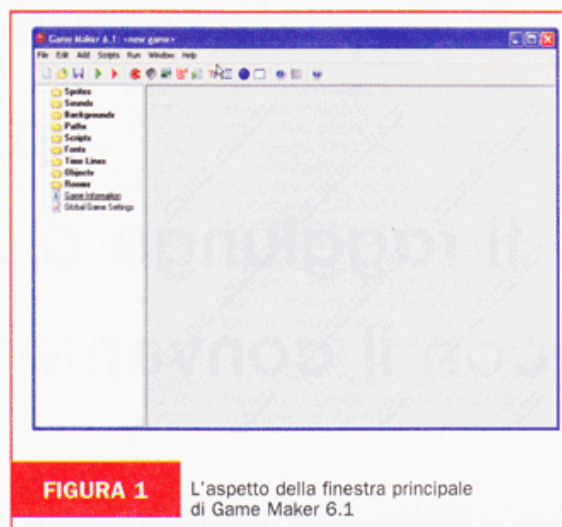
Il menu ad albero verticale posto sul lato sinistro della finestra principale rappresenta il fulcro dell'attività di sviluppo in Game Maker. Questo, infatti, raggruppa, tra le altre, anche la maggior parte delle funzionalità offerte dai menu appena visti.

### Preparazione del progetto

Per poter iniziare a lavorare è necessario creare un nuovo file di progetto salvando lo stato di un progetto vuoto tramite la voce *Save* o *Save as* nel menu *File*.

E' bene specificare che un file di progetto Game Maker presenta un'estensione ".gmX" dove la X sta per la versione di Game Maker nel quale il progetto è stato creato. Così per i progetti della versione 6 verranno creati file di tipo ".gm6". Quando si aggiunge una risorsa immagine, suono, script o altro questa viene inserita permanentemente all'interno del file di progetto. In questa maniera a meno che non si utilizzino DLL esterne ogni lavoro realizzato non dipenderà da alcun file.

Ogni file EXE prodotto dalla funzione *Create Executable* presente nello stesso menu *File*, infatti, è in grado di funzionare da solo anche se distribuito ed eseguito su altri computer dove non è stato installato l'ambiente.



**FIGURA 1** L'aspetto della finestra principale di Game Maker 6.1



## Lavorare con gli sprite

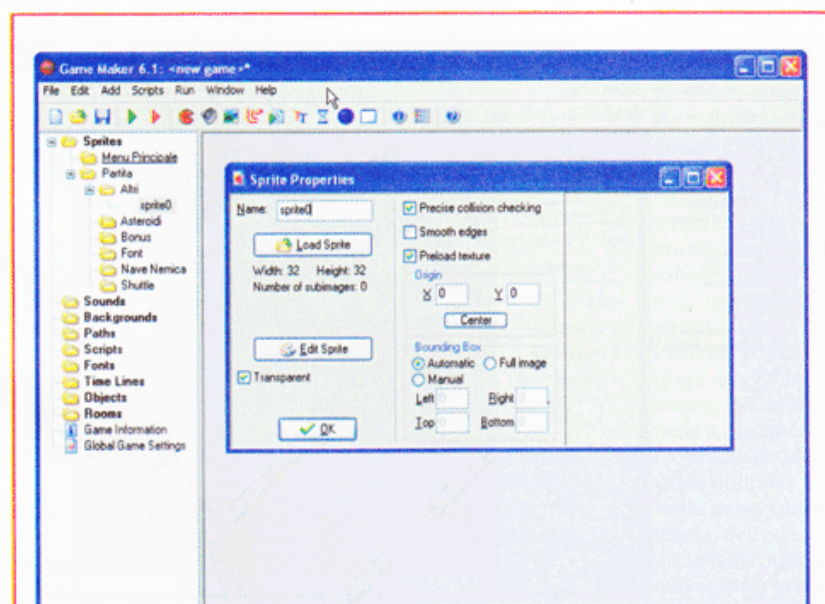
Abbiamo già parlato, a grandi linee, degli *sprite* e del ruolo che rivestono in Game Maker, ma per quelli che non hanno letto i precedenti articoli possiamo brevemente dire che con *sprite* si intende un'immagine o serie di immagini che vengono visualizzate sullo schermo in luogo dei personaggi del gioco o le altre entità al quale prendono parte.

Per aggiungere uno *sprite* al nostro progetto è sufficiente fare clic con il tasto destro del mouse, nel menu ad albero a sinistra, alla voce *Sprites*. Comparirà il menu pop-up con attive le voci *Add Sprite*, *Add Group* e *Sort by Name*. L'ultima serve per ordinare gli elementi in ordine crescente secondo il loro nome, la seconda per creare una sottocartella nella categoria, mentre la prima per creare un nuovo *sprite*. Per prima cosa creiamo due sottocartelle chiamate *Partita* e *Menu Principale*, questo per mantenere una separazione tra le voci appartenenti ad ogni modulo del gioco secondo le convenzioni decise in fase di analisi. All'interno di *Partita* si dovranno creare le seguenti sottocartelle *Altri*, *Asteroidi*, *Bonus*, *Font*, *Nave Nemica*, *Shuttle*. Questa operazione sarà effettuata per tutte le categorie con le quali lavoreremo e, per questioni di brevità, mi riserverò di ometterne i passi durante la trattazione.

Cliccando sulla cartella *Altri* e selezionando, questa volta, la voce *Add Sprite* verrà creata una nuova voce chiamata *spriteXX* come si vede in **Figura 2**, dove XX sta per il numero progressivo degli *sprite* creati nel progetto. Contemporaneamente si aprirà una finestra chiamata *Sprite Properties*.

Questa finestra è suddivisa in tre sezioni, una contenente i pulsanti per creare, modificare gli *sprite*, una contenente alcuni parametri che vedremo tra poco e l'ultima per visualizzarne l'anteprima.

Nella casella *Name* inseriamo il nome *sprController* e premendo il pulsante *Load Sprite* apparirà la classica finestra di dialogo tramite la quale è possibile caricare un'immagine. Scegliamo il file contenuto nel pacchetto del progetto "[DirectoryProgetto]\AstroCrisis\Grafica\Sprite\Altri\PC.ico".



**FIGURA 2**

Quello che accade quando viene creato un nuovo *sprite* è la comparsa della finestra delle proprietà degli *sprite*.

### RIQUADRO 1

Le principali differenze di funzionalità tra le due modalità *Advanced* e *Simple*

#### Modalità *Advanced*

Sprites  
Sounds  
Backgrounds  
Paths  
Scripts  
Fonts  
Time Lines  
Objects  
Rooms  
Game Information  
Global Game Settings

#### Modalità *Simple*

Sprites  
Sounds  
Backgrounds  
Objects  
Rooms

Nella finestra verranno apportate le modifiche opportune ai parametri.

Le voci *Width* ed *Height* riportano le dimensioni in larghezza ed altezza dello *sprite*, mentre *Number of subimages* indica il numero di fotogrammi di cui lo stesso è composto. Premendo il tasto *OK* le modifiche verranno memorizzate.

Analizzeremo ora le caratteristiche più importanti per creare uno *sprite* animato. Come prima è necessario aggiungere un nuovo *sprite* al progetto nella cartella *Shuttle*, ma questa volta si dovrà cliccare sul pulsante *Edit Sprite*. Verrà aperta un'ulteriore finestra, lo *Sprite Editor*. Questo editor permette di raggruppare diversi fotogrammi per realizzare un'animazione. E' possibile anche creare animazioni partendo da una singola immagine tramite le funzioni contenute nel menu *Animation* come le sequenze di rotazione, di traslazione, di ridimensionamento, di morph ed altre.

Nei menu *Images* e *Transform* sono presenti varie funzioni per la modifica di un singolo fotogramma in particolare per l'applicazione di effetti sui colori e di trasformazioni geometriche.

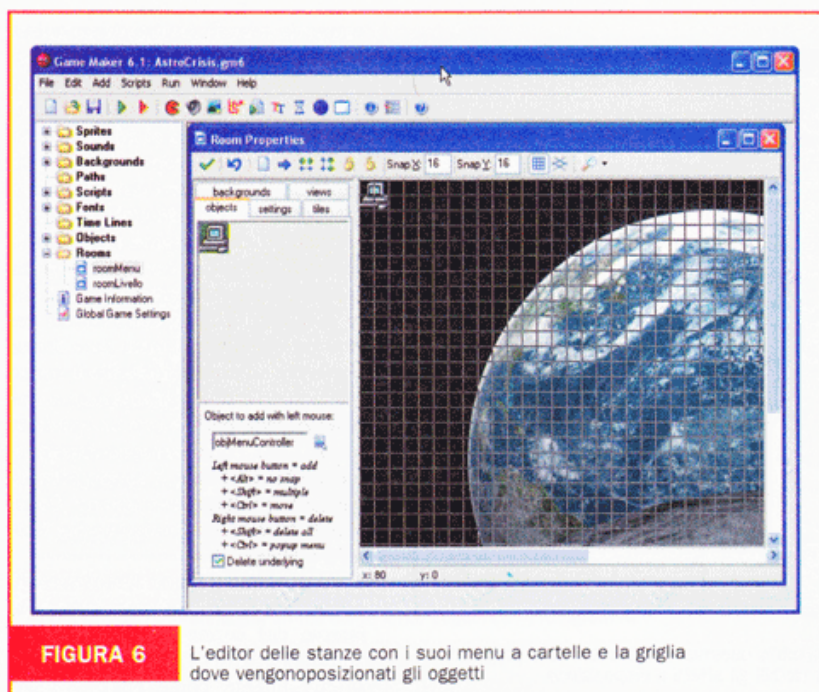
Il menu *Edit* contiene le funzioni classiche taglia/copia/incolla più alcune per modificare la posizione dei fotogrammi all'interno della sequenza.

Dal menu *File* possiamo creare, aggiungere e salvare singoli fotogrammi o intere sequenze da file esterni.

Nel nostro caso risulta interessante osservare come vengono caricate le sequenze di fotogrammi sotto forma di file *Strip* il quale non è altro che un'immagine contenente i fotogrammi dell'animazione affiancati tra loro. A questo scopo selezionare dal menu *File* la voce *Create from Strip* e caricare il file "[DirectoryProgetto]\AstroCrisis\Grafica\Sprite\Shuttle\Shuttle\_93x93\_178.Strip". Per evitare problemi di decodifica del file, assieme al nome, sono state aggiunte le informazioni relative alla dimensione del singolo fotogramma, 93x93, e il numero di questi, cioè 178.

La finestra di dialogo che si apre dopo la selezione del file chiamata *Loading a strip image* permette di decodificare la sequenza scelta tramite l'impostazione dei parametri corretti. Il significato delle caselle risulta abbastanza intuitivo.





**FIGURA 6** L'editor delle stanze con i suoi menu a cartelle e la griglia dove vengono posizionati gli oggetti

espressa in fotogrammi al secondo. La voce *Persistent* sta ad indicare se le istanze degli oggetti, di questa stanza, devono continuare ad esistere passando ad altre stanze. Per tutte e due le stanze impostiamo "AstroCrisis" come caption, 800 pixel di larghezza, 600 pixel di altezza, una frequenza di aggiornamento di 60 fotogrammi al secondo e la voce *Persistent* disabilitata.

Il pulsante *Creation Code* non è importante per i nostri scopi in quanto permette di specificare un codice in GML da eseguire ogni volta che la stanza viene visualizzata. Creeremo un oggetto al quale, questo compito, verrà delegato.

Ignoriamo la tabella *Tiles* che serve a gestire gli sfondi come composizione di tante piccole immagini e passiamo direttamente alla *Backgrounds*. Qui possiamo decidere se l'aspetto dello sfondo della stanza deve essere riempito con un colore oppure tramite una delle immagini presenti nella categoria *Backgrounds*.

Le caselle denominate *X* e *Y* indicano la posizione dell'immagine all'interno della stanza, mentre *Hor. Speed* e *Vert. Speed* permettono di animare lo sfondo facendolo muovere un numero dei pixel indicato ad ogni fotogramma. Per la stanza *roomLivello* dobbiamo impostare l'immagine di sfondo *bgTerra*, per l'altra *bgMenu*.

In entrambi i casi è necessario che sia selezionata la voce *Visible when room starts*.

L'ultima e non meno importante tabella è quella chiamata *Views*. Impostando opportunamente i parametri qui presenti è possibile fare in modo che un oggetto venga seguito in ogni suo movimento e centrato sullo schermo. Al seguire dell'oggetto verranno spostati anche tutti gli altri creando di spostamento all'interno di un'area di dimensioni maggiori dello schermo.

Ogni stanza ha a disposizione otto viste, ognuna con i propri parametri ed oggetti da agganciare. Solo per la stanza *roomLivello* selezionare la voce *Enable the use of views*, la *view0* nell'elenco delle viste e la voce *Visible when room starts* la quale indica che la vista deve essere attivata appe-

na si entra nella stanza. Le caselle *View in room* e *Port on screen* devono riportare i valori  $X = 0$ ,  $Y = 0$ ,  $W = 800$ ,  $H = 600$ . Queste impostazioni servono a definire le posizioni e le dimensioni della vista rispetto alla *view port* e allo schermo.

In *Object following* viene impostato l'oggetto da seguire, mentre le coppie *HBor*, *VBor* e *Hsp*, *Vsp* rappresentano rispettivamente l'area entro la quale tale oggetto può muoversi senza che la vista si sposti e la velocità di spostamento della vista rispetto all'oggetto. Queste ultime impostazioni verranno effettuate nel prossimo articolo.

### Le impostazioni globali

Nel menu ad albero sulla finestra principale troviamo la voce *Global Game Settings*. Questa ci permette di accedere ad un'area dove è possibile impostare i parametri generali che influenzano tutto il progetto. Si possono personalizzare molti aspetti del gioco come la modalità a finestra o a schermo pieno, la risoluzione video, la barra di caricamento, le costanti, i

file inclusi, la gestione degli errori e le informazioni sulla versione del prodotto.

### La pagina delle informazioni

Come altra facilitazione Game Maker mette a disposizione un semplice meccanismo per aggiungere le informazioni sul gioco tramite un editor testuale accessibile dalla voce *Game Information*. Questa finestra potrà essere visualizzata in ogni momento durante l'esecuzione del gioco.

### Conclusioni

A questo punto abbiamo preparato il nostro progetto con tutte le risorse necessarie alla realizzazione del videogioco proposto. Nel prossimo articolo, quindi, creeremo gli oggetti e realizzeremo i loro comportamenti tramite gli script in GML.

Mi raccomando a tutti i lettori interessati all'argomento di scriverci e non esitare a farci domande per ogni aiuto necessario e a manifestare il vostro interesse a questo affascinante mondo, lo sviluppo dei video giochi.

### Bibliografia

- [1] Mark Overmars, "Designing Games with Game Maker", [www.gamemaker.nl](http://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).