PROGETTO V4T

CORSO DI E-LEARNING

Capitolo 1 - Introduzione alla programmazione dei videogiochi

Tecnologie disponibili - Parte 1

La creazione di giochi coinvolge diverse aree di lavoro e di conoscenza, dalla creazione artistica alla programmazione informatica di basso livello. Per ogni area, ci sono strumenti di aiuto o di sviluppo e tecnologie specializzate. La maggior parte di questi strumenti può essere utilizzata in modo indipendente, dando ai team di sviluppo del gioco la possibilità di lavorare separatamente per una grande parte del progetto. Nelle fasi finali dello sviluppo, tutto il lavoro dei diversi team e creatori viene integrato nel prodotto finale. Per ottenere questo risultato, il team di sviluppo del gioco utilizza un software che prende tutti gli elementi del gioco e genera il prodotto finale. Questo software è in grado di gestire immagini, suoni, video, unitamente ad un insieme di regole o a un programma e generare così il gioco vero e proprio.

In sostanza, la creazione di un gioco è programmazione, come nella creazione di qualsiasi altro software. Tuttavia, con l'evoluzione del software e la crescita della specializzazione, sorgono differenze tra lo sviluppo di giochi o di software in generale. E gli strumenti utilizzati per assistere gli sviluppatori di software si evolvono e si specializzano di conseguenza. Spesso le aziende di software offrono nuovi strumenti per realizzare qualsiasi programma, sviluppo web, software integrato, ecc. che promettono un balzo di semplificazione e di efficienza in quel campo. Questo vale anche per gli strumenti di sviluppo dei giochi.

Tuttavia, a differenza di altri settori del software, le tecnologie di sviluppo dei giochi vanno da strumenti molto personalizzati a strumenti molto aperti. Ci sono alcuni strumenti che possono essere utilizzati per creare giochi complessi di qualsiasi genere, mentre altri strumenti hanno lo scopo di creare un tipo di gioco molto specifico. Ci sono anche alcuni giochi che vengono forniti con gli strumenti necessari per espandere quel gioco con nuovi personaggi, o livelli, offrendo un prodotto misto gioco-strumento di sviluppo.

Inoltre lo sviluppo del gioco condivide alcune tecnologie chiave con altre aree scientifiche e delle tecnologie dell'informazione. Ad esempio, l'imaging 3D è una parte importante della rivoluzione dei

Co-funded by the Erasmus+ Programme of the European Union
The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

giochi, e le fondamenta sottostanti sono anche la base della progettazione e della produzione assistita da computer (CAD/CAM), della chirurgia assistita da computer, dei visualizzatori di mappe globali (google Earth) e così via.

1.2 - Tecnologie disponibili - Parte 2

C'è un elenco di campi dell'informatica che sono direttamente collegati allo sviluppo di giochi:

- Ingegneria del software. Come ogni software per computer, i giochi sono costruiti da sviluppatori che devono essere abili e addestrati nell'utilizzo delle più recenti tecniche di ingegneria del software.
- Linguaggi di programmazione. Che sono essenziali per costruire qualsiasi software complesso.
- Immagini 3D e 2D. I giochi sono il software più intensivo quando si tratta di visualizzare immagini, animazioni, movimenti, ecc. Questo ha portato i produttori di hardware a sviluppare e costruire potenti schede grafiche in grado di visualizzare scene complesse in un'animazione continua e fluida. Come risultato sono emersi alcuni standard per consentire ai costruttori di software di utilizzare questo potente hardware. Il più universale è OpenGL. Inoltre, il software di modellazione che aiuta le persone a progettare oggetti in un mondo tridimensionale è maturato e reso più semplice da usare.
- Apprendimento automatico e intelligenza artificiale. Giocare contro il computer richiede che lo sviluppatore programmi il gioco con una certa logica su come sfidare il giocatore. Di solito, il gioco ha anche "livelli di difficoltà" per abbinare il computer all'abilità del giocatore. Tuttavia c'è uno sforzo continuo per far sì che i giochi si adattino automaticamente ai giocatori e imparino a sconfiggerlo nel tempo.

Inoltre, non possiamo dimenticare molte tecnologie, conoscenze e sviluppi che sono alla base del lavoro di non programmazione nello sviluppo di un gioco.

- Cinema, fumetti e grafica. I giochi tendono a ritrarre i personaggi, i paesaggi e l'ambiente nel modo più sorprendente, cercando di far vivere al giocatore un'esperienza potente. Ci sono tecniche ben note ed emergenti nella realizzazione di film che vengono riutilizzate per fare giochi. Queste tecniche sono di per sé una sorta di tecnologia che gli sviluppatori di giochi devono acquisire. I giochi riflettono le tecniche di ripresa del cinema.
- Psicologia. Da un certo punto di vista, giocare è come eludere la realtà. I game designer hanno successo quando il loro gioco assorbe e fa evadere il giocatore dal mondo in cui vive. Alcuni giochi vanno oltre e prendono di mira i sentimenti dei giocatori, come la tristezza, la rabbia, la paura, ecc. I giochi prendono lezioni di psicologia, per raggiungere meglio questi obiettivi.
- Scienze sociali. I giochi multigiocatore, specialmente i giochi multigiocatore di massa, sfruttano la socializzazione dei giocatori, approfittando della motivazione extra che un giocatore

Co-funded by the Erasmus+ Programme of the European Union
The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

sente quando sa che sta giocando contro un'altra persona. Il gioco diventa così uno strumento per far incontrare una persona con gli altri (mentre si gioca). Il modo in cui i giocatori interagiscono è oggetto di studio, le persone tendono a farsi amicizie, nemici, pianificare partite, ecc. Esiste una branca di scienza sociale legata ai giochi multiplayer online di massa. Il risultato è una conoscenza preziosa che viene applicata ai successivi giochi multiplayer.

1.2 Ambiente di sviluppo e linguaggi di programmazione - Parte 1

Chiamiamo Ambiente di Sviluppo o "Ambiente di Sviluppo Integrato" ("Integrated Development Environment", IDE d'ora in poi) un software speciale che aiuta gli sviluppatori a costruire software. L'IDE di base include un editor, un compilatore e un debugger, anche se l'IDE di oggi è dotato di diversi altri strumenti di aiuto. Man mano che le tipologie di software si differenziano, lo stesso vale per gli IDE. Esiste quindi un'ampia gamma di IDE che si rivolgono allo sviluppo di software in diversi campi. Gli IDE per sviluppare giochi sono uno dei più specializzati, con molte peculiarità. Un elemento centrale in qualsiasi sviluppo di software è il linguaggio utilizzato. Alcuni IDE sono progettati intorno ad un certo linguaggio, mentre altri possono essere configurati per utilizzare diversi compilatori o linguaggi.

Gli IDE per lo sviluppo di giochi hanno diverse caratteristiche speciali, ma due sono di grande importanza. La prima è l'integrazione di più strumenti per produrre diverse parti di un gioco (grafica, programmazione, suoni, networking, ecc.), e la seconda importante caratteristica è la specializzazione nello sviluppo del gioco, dove la maggior parte degli elementi di un gioco sono pre-costruiti (livelli, mappe, personaggi, animazioni, ecc.).

Un IDE per creare giochi deve permettere allo sviluppatore di programmare diverse parti del gioco che hanno scopi diversi. Per soddisfare le diverse esigenze, l'ambiente può consentire allo sviluppatore di utilizzare diversi linguaggi. Ad esempio, uno sviluppatore di sparatutto in prima persona online deve fare un uso massiccio della programmazione 3D, ma deve anche occuparsi della comunicazione con un server remoto attraverso la rete. Potremmo aver bisogno di linguaggi totalmente diverse per queste due esigenze. Inoltre, il gioco può avere delle regole su ciò che è permesso o vietato al giocatore, o sui checkpoint che l'utente deve visitare. Questo requisito di solito richiede un linguaggio di alto livello col quale le regole possano essere scritte facilmente. Si consideri anche che il gioco può avere dei bot, cioè giocatori artificiali del computer che giocano autonomamente. Come programmeremmo l'intelligenza e il comportamento di questi bot?

Molti ambienti includono il proprio linguaggio cercando di soddisfare ogni esigenza, pur mantenendolo semplice nel tentativo di rendere facile l'uso dell'ambiente, mentre altri consentono che il linguaggio generale (tipicamente C++) sia il linguaggio principale dell'ambiente. Questi ultimi però non limitano l'uso di altri linguaggi, perché ci sono strumenti e modi per incorporare le routine e il codice

fatti in un programma non C++ in un programma C++. Ma anche i linguaggi generici come C++, rust, python o lua, che sono usati nello sviluppo di giochi, sono potenziati con una serie di classi e funzioni di libreria che sono strettamente integrate nel motore di gioco.

1.3 Ambiente di sviluppo e linguaggi di programmazione - Parte 2

Un'alternativa interessante all'utilizzo degli IDE è l'utilizzo di "editor" di giochi. Alcuni giochi sono distribuiti con editor integrati. Sono gli stessi strumenti che il team di sviluppo ha utilizzato per creare i livelli, le mappe, le fasi delle fasi del gioco, e che sono stati essi stessi creati dagli sviluppatori del gioco.

Il giocatore può usare liberamente l'editor per costruire il proprio livello, compresi il proprio impianto visuale, le immagini e i suoni. Alcuni editor consentono di apportare semplicemente alcune modifiche alle mappe predefinite, mentre la maggior parte di essi permette di personalizzarle completamente o addirittura sviluppare nuove mappe e nuovi livelli.

I generi di gioco che di solito includono gli Editor sono quelli di strategia in tempo reale e gli sparatutto in prima persona. Da un certo punto di vista, un gioco editabile o modificabile consiste nei seguenti elementi:

- Un insieme di risorse o beni come immagini, suoni, animazioni, ecc. che compongono l'aspetto fondamentale degli oggetti del gioco
- Un insieme di mappe, livelli o tappe che corrispondono ad una delle tante sfide che il giocatore deve completare
- Un software che prende gli elementi di cui sopra e crea il gioco vero e proprio eseguendo le mappe e le risorse. Questo elemento è chiamato "Motore di gioco".

Per creare un gioco, potrebbe essere necessario scegliere un solo gioco con un editor integrato e poi progettare i livelli e le risorse per esso. Molti, se non tutti gli editor dispongono di un linguaggio di programmazione di scripting che consente un comportamento preciso e dettagliato degli elementi di gioco (nemici, proiettili, "power-up", ecc.)

Questo approccio è stato particolarmente utilizzato Per sviluppare alcuni giochi davvero complessi, i programmatori prendono due fasi comprensive di risultati diversi. Inizialmente costruiscono un motore che può far funzionare il gioco, ma i contenuti vengono creati separatamente. Il motore di gioco è abbinato ad un editor specializzato utilizzato per generare livelli, grafica, personaggi, ecc. solo per questo motore. Poi, in una seconda fase completano tutti i contenuti di gioco che costruiscono la storia del gioco in livelli.

Quando il gioco è finito, due prodotti sono disponibili per il pubblico: il gioco vero e proprio con il contenuto completo insieme al motore incorporato e l'editor per essere utilizzato dagli appassionati

Co-funded by the Erasmus+ Programme of the European Union
The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

che amano fare nuovi contenuti. Possono essere venduti separatamente portando alcuni sviluppatori a utilizzare il motore e il suo editor per costruire un gioco diverso, non solo una modifica o un'estensione del gioco originale. A volte, il motore di gioco viene annunciato e venduto a terzi o al pubblico in modo indipendente anche prima che qualsiasi gioco sia stato completato con quel motore. Ma comunque questi non possono essere considerati strumenti di sviluppo di giochi "generici", perché si rivolgono ad un tipo preciso di gioco fin dall'inizio.
Questo è qualcosa di diverso dal semplice "gioco con un editor integrato".
Esiste una buona e crescente serie di motori di gioco

1.4 Ambiente di sviluppo e linguaggi di programmazione - Parte 3

Infine, e seguendo la tendenza a stabilire negozi di software online, come Steam di Valve, Google Play, Sony PlayStation Store, alcuni IDE facilitano il compito di pubblicare e distribuire il software. L'IDE è in grado di utilizzare l'account dello sviluppatore in uno qualsiasi di questi negozi digitali per pubblicare un gioco senza problemi.
Come conclusione di questa unità, dobbiamo considerare la convenienza di utilizzare qualche generico strumento di sviluppo di giochi, che è focalizzato sulla semplicità d'uso e pensato per i principianti senza alcuna conoscenza precedente di programmazione. E, dalla vasta lista di software disponibili, GameMaker di YoYo Games è uno strumento molto adatto a questo progetto. Si tratta di una soluzione collaudata e ampiamente utilizzata, con una grande comunità di utenti e diversi prodotti di successo.

Co-funded by the Erasmus+ Programme of the European Union
The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Capitolo 2 - Game Maker

2.1 Caratteristiche del framework Game Maker. Tipo di giochi.

"GameMaker Studio" ("GameMaker" in questo tutorial) è un ambiente integrato completo per sviluppare giochi. Inizialmente è stato concepito per permettere ai novizi di creare giochi semplici senza avere molte conoscenze di programmazione. Per soddisfare questa esigenza, ha offerto diverse tecnologie per consentire di comporre visivamente l'equivalente di un programma per computer. Tuttavia, se necessario, gli sviluppatori possono scrivere codice e usare un linguaggio di programmazione simile al C ma con alcune idee prese da javascript. Il risultato è un linguaggio abbastanza facile da essere usato dai novizi.

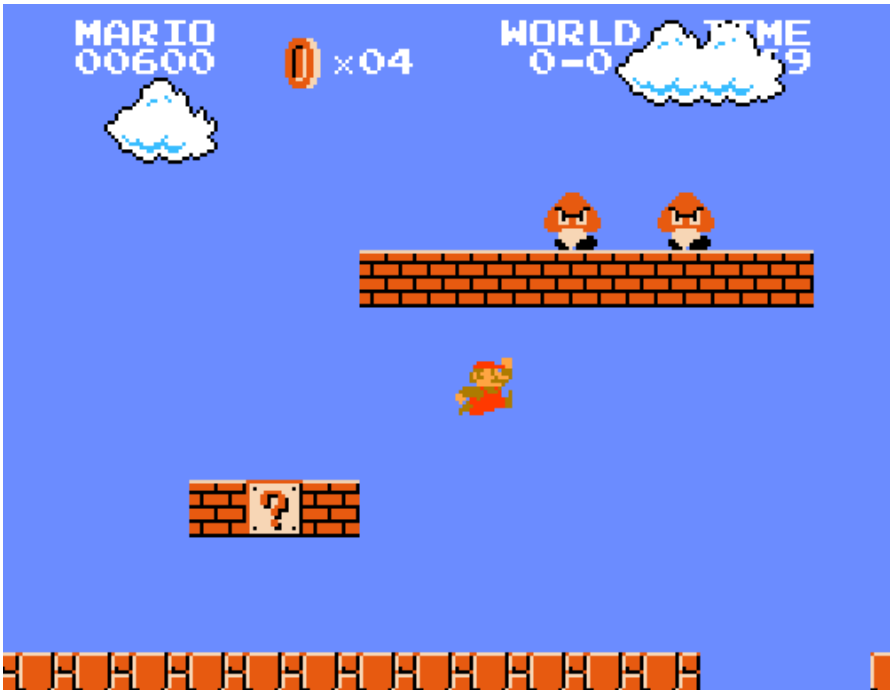
Le caratteristiche principali sono:

- Integrato. Tutte o la maggior parte delle fasi di sviluppo del gioco possono essere fatte con strumenti integrati. Per la maggior parte, non ha bisogno di aiuto da parte di software esterni.
- Ha un proprio linguaggio (Game Maker Language o GML) che non è sconosciuto ai programmatori abituati a C, Javascript, ecc. ed è strettamente integrato con il motore di gioco e l'ambiente di sviluppo.
- Per i giochi semplici non è necessario alcun codice. Game Maker permette allo sviluppatore di utilizzare la programmazione "DnD" Drag and Drop Programming come alternativa all'uso di GML
- I giochi possono essere creati per la maggior parte dei sistemi operativi e delle piattaforme di gioco. Ed è inclusa anche l'integrazione con le piattaforme di distribuzione digitale.
- Particolarmente adatto a diversi tipi di giochi, ma può essere adattato a molti altri generi con un po' di lavoro extra.

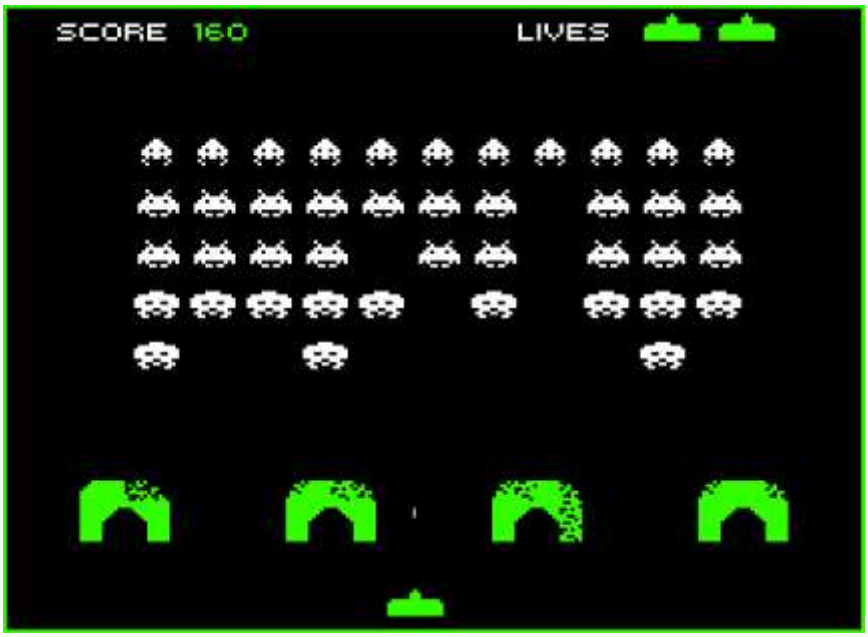
Game Maker eccelle in semplicità e l'utilità nella creazione di giochi di piattaforma e shoot'em up giochi. Ma si possono creare anche altri tipi di giochi.

- Nei giochi di piattaforma, un personaggio si muove in un mondo bidimensionale che raccoglie oggetti, evitando gli ostacoli e distruggendo i nemici. Un esempio ben noto è Mario Bros

Co-funded by the Erasmus+ Programme of the European Union
The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.



- Shoot Them up giochi. In questo gioco il giocatore avanza continuamente su una mappa, sia in volo che a piedi, e inoltre può raggiungere diversi punti dello schermo. I nemici appaiono nella direzione "avanti" del giocatore, che deve evitare o uccidere quei nemici. Un esempio eterno è “Space Invaders".



Co-funded by the Erasmus+ Programme of the European Union
The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

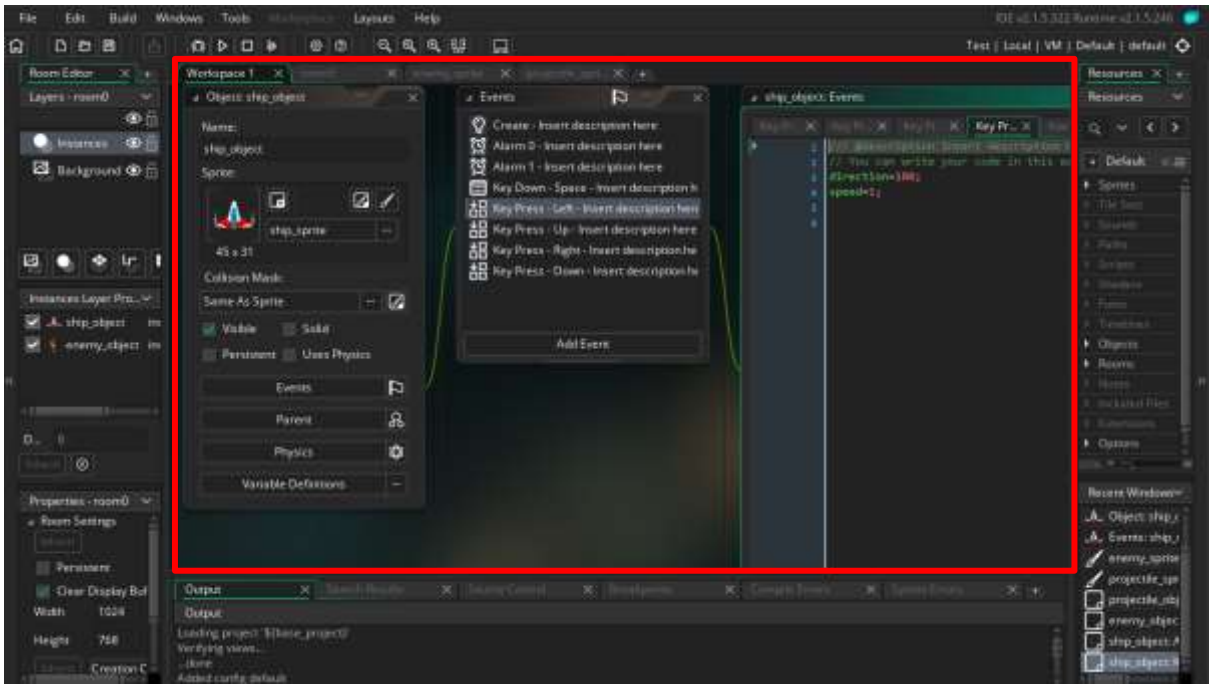
2.2 Ambiente di sviluppo

Game Maker è un ambiente di sviluppo completo. Include tutti gli strumenti necessari per sviluppare e testare i giochi senza richiedere strumenti esterni, dispositivi o configurazioni speciali. L'interfaccia utente è costruita attorno ad un telaio centrale dove possono essere allocati più spazi di lavoro. Gli spazi di lavoro sono pannelli o tavole in cui gli elementi e le risorse di gioco possono essere posizionati e organizzati durante lo sviluppo per consentire un'organizzazione più personalizzata. Gli spazi di lavoro stessi visualizzeranno finestre interne con contenuti diversi, da oggetti generici con proprietà, a liste di eventi e al codice GML. Ogni finestra di elemento che fa parte di un altro o che è collegata in qualche modo ha una linea tracciata da un elemento all'altro.

Inoltre, intorno all'area di lavoro centrale, ci sono pannelli secondari. La maggior parte di essi sono collassabili, se necessario, per consentire l'espansione dell'area centrale. Questi pannelli mostrano:

- Una lista ad albero di ogni risorsa del gioco. Questo albero è perpetuo e cambia solo quando vengono create o cancellate risorse (o attività).
- Una lista di controlli per modificare le proprietà di qualsiasi risorsa selezionata. Questo pannello cambia ogni volta che lo sviluppatore si concentra su un elemento diverso.
- Una finestra di output per vedere il risultato della compilazione e dell'esecuzione del gioco.

Un'altra finestra può apparire sia solitaria, sopra il resto o come scheda all'interno dell'area centrale quando viene lanciato uno strumento speciale. Per esempio, l'editor di immagini viene usato raramente, ma quando una risorsa immagine viene modificata, viene portata in primo piano come scheda nell'area centrale.



Game Maker offre due modi per creare un gioco:

Co-funded by the Erasmus+ Programme of the European Union
The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

- Trascina e rilascia (“Drag and Drop”, o "DnD"). Questa modalità è destinata ai principianti che non amano o non vogliono codificare ma vogliono costruire giochi standard. Game Maker permette di definire la logica del gioco utilizzando uno "strumento di scripting visivo". In questa modalità, lo sviluppatore trascina e rilascia gli elementi logici che sono collegati e raggruppati per specificare il comportamento di qualsiasi elemento.
- Progetti di Game Maker Language. Questa modalità è quella predefinita e standard, e la logica del gioco e il comportamento di qualsiasi elemento è specificato dal codice di programmazione in linguaggio GML.

In questo tutorial useremo l'opzione GML, poiché sebbene la complessità degli esempi sarà poco profonda e non faremo una programmazione complessa, una volta completata, avremo più possibilità aperte in futuro.

2.3 Sprite, stanze, oggetti e collisioni.

I giochi visualizzano gli elementi in uno schermo in modo mutevole e dinamico. Ciò che il giocatore vede sono semplicemente "cose" che si muovono all'interno di un'area o di un "mondo", e spesso le cose si schiantano, urtano o si scontrano con altre cose.

Game Maker, e anche sviluppatori e giocatori, danno nomi specifici per i diversi elementi visivi.
L'area o "mondo" in cui avviene il gioco, in GameMaker si chiama stanza.

Le immagini animate sono chiamate "sprite". Le immagini statiche o non animate sono solo decorative e possono essere chiamate sfondo.

Agli elementi di un gioco che hanno un certo comportamento o che influenzano in qualche modo il gioco viene assegnato il nome di "oggetto".

E infine, la "collisione" è l'evento più comune che avviene tra due oggetti quando si incontrano fisicamente.

Questi elementi sono alla base delle caratteristiche principali di GameMaker.

Sprite, stanze, oggetti e collisioni - Una guida approfondita sugli elementi principali di un gioco

2.4 Eventi e azioni

Un gioco, quando è avviato o si gioca, è un sistema dinamico. È qualcosa che cambia, spesso rapidamente dal punto di vista del giocatore. Questi continui cambiamenti nel gioco sono in realtà il risultato di piccoli cambiamenti molto frequenti, percepiti come una svolta più grande e complessa degli eventi. La chiave per determinare come il gioco si evolve è capire i cambiamenti di base e come stabilirli a partire dalle loro componenti chiamate "eventi" e "azioni".

Per comprendere meglio gli eventi e le azioni, dobbiamo prima considerare che un gioco per computer è in realtà un programma in esecuzione continua. A volte questo programma è noto come "motore di gioco". Questo programma controlla frequentemente la tastiera, il mouse e altri controller; può leggere l'ora, conosce la posizione e la direzione di qualsiasi oggetto del gioco, ecc. Pertanto, il motore di gioco è il primo a sapere che è successo qualcosa: o è stato premuto un tasto, o è trascorso un ritardo, o due oggetti si sono toccati sullo schermo, ecc. GameMaker richiama eventi a queste cose che si verificano e possono influenzare il gioco. Molti eventi nascono come risultato dell'interazione del giocatore con il computer. Alcuni altri eventi si verificano come risultato dell'evoluzione del gioco.

Il motore di gioco controlla il gioco e può cambiare lo stato di qualsiasi cosa in esso contenuta, in qualsiasi momento desiderato. Può riposizionare o spostare un oggetto, aggiungere qualche danno, creare un nuovo oggetto, ecc. Queste modifiche avvengono come previsto dalla programmazione effettuata dallo sviluppatore, poiché il motore di gioco è solo l'esecutore di un programma. GameMaker richiama "azioni" a fronte di un cambiamento che viene specificato dallo sviluppatore. La chiave qui è che ogni azione avviene come risultato di un evento, anche le azioni che sono impostate per accadere in modo casuale sono collegate ad eventi con una componente casuale che agisce su di esse. L'obiettivo dello sviluppatore è quello di stabilire quali azioni avvengono in conseguenza di quali eventi. Analizzeremo prima gli eventi in profondità e poi vedremo come le azioni sono legate ad essi

C'è una lunga lista di eventi prestabiliti da Game Maker, ma lo sviluppatore del gioco è libero di intraprendere qualsiasi azione in risposta a qualsiasi evento scelto (si prega di fare riferimento alla risorsa Online).

Eventi e azioni - Una guida approfondita sugli eventi principali di GameMaker

2.5 Azioni condizionali

A volte è necessario eseguire azioni su condizioni complesse per le quali non vi è alcun evento associato.

Co-funded by the Erasmus+ Programme of the European Union
The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

In questi casi, lo sviluppatore deve scrivere una condizione speciale in base alla quale, l'azione sarà eseguita.
E il motore di gioco deve essere istruito a controllare le condizioni in ogni momento possibile.
Quando la condizione è soddisfatta, l'azione viene eseguita. Questo offre una grande flessibilità allo sviluppatore.
Tuttavia, le azioni condizionate sono alla fine azioni che devono essere collegate ad un evento (questo è il modo in cui funziona GameMaker).
Se la condizione deve essere controllata ogni volta, allora l'evento è naturalmente l'evento chiave.
E l'azione è semplicemente un pezzo di programma che controlla ulteriormente la condizione.

Azioni condizionate - Fare riferimento a questo documento per avere una visione più chiara di come funzionano le azioni condizionate

2.6 Animazione e collisione avanzata

L'animazione nei giochi è fondamentale per una buona esperienza di gioco e di immersione. Fin dall'inizio dei videogiochi, i programmatori e gli artisti hanno lavorato duramente per produrre il look più realistico e accattivante nei giochi. Ciò richiede di replicare i movimenti che molti oggetti hanno nella vita reale. Per esempio, guidare un'auto da sopra o da dietro non dovrebbe portare solo a guardare l'auto che si muove lungo una strada, ma anche a vedere le ruote che girano, il fumo che esce dallo scarico, l'auto stessa che urta o la strada che scuote le gomme. Ci sono diverse tecniche per generare effetti di animazione nei giochi, alcune delle quali sono ben integrate in Game Maker. Le immagini si occupano anche della rappresentazione fisica di oggetti che spesso si muovono e si scontrano con altri oggetti.

La tecnica più estesa per ottenere animazioni, deriva dall'utilizzo di un insieme di immagini simili che vengono visualizzate ciclicamente. Se la velocità con cui le immagini cambiano è quella giusta, il giocatore percepirà le immagini come un'animazione. Il difficile compito che rimane è quello di disegnare un insieme di immagini che producano la migliore animazione possibile

Consideriamo che lo “sprite” abbia diverse sotto-immagini. Prendete ad esempio la seguente "striscia" di immagini. Lo sprite dovrebbe mostrare una ragazza alla volta passando rapidamente alla successiva in fila, dando l'effetto che la ragazza cammini effettivamente muovendo le gambe (l'immagine dovrebbe cambiare insieme alla posizione).

https://opengameart.org/content/girl-walking-side

Co-funded by the Erasmus+ Programme of the European Union
The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Game Maker è molto potente quando si tratta di creare e utilizzare gli sprite. Può leggere un file come quello di cui sopra e "dividerlo" in diverse sotto-immagini abbastanza facilmente, creando così lo sprite quasi automaticamente. Può gestire immagini le cui sotto-immagini sono allineate in altri layout (ad esempio in una matrice)

L'utilizzo di sprite in Game maker è abbastanza facile, molto potente e dà buoni risultati. Ma disegnare o creare gli spites è un'altra storia. Per iniziare a creare giochi, ci sono molti set di sprite gratuiti in alcuni siti web. Uno di questi siti web è http://opengameart.org

Gli sprite impostano anche la forma dell'oggetto e, a seconda delle dimensioni dell'oggetto rispetto alla stanza, può essere inaccettabile gestire le collisioni tra oggetti che si limitano a relazionarsi sulle coordinate degli oggetti. Ad esempio, si consideri su un oggetto piccolo che si muove verso un oggetto grande. Nella maggior parte dei casi, il rilevamento delle collisioni deve evitare che gli oggetti si sovrappongano o penetrino l'uno nell'altro. Pertanto, il sistema di collisione può tener conto non solo della posizione degli oggetti, ma anche della forma e delle dimensioni.

Quando si definisce uno sprite, è anche utile e semplice impostare la sua "maschera di collisione", che si applica agli oggetti che lo visualizzano. La maschera di collisione è un'area di forma potenzialmente arbitraria. In genere, per ragioni di performance, la maschera di collisione è un rettangolo, che può essere ruotato se necessario per meglio adattarsi alla forma dello sprite Anche i cerchi e le ellissi possono essere impostati come maschera di collisione, ma sono più lenti da calcolare. Infine, in casi molto particolari, la maschera di collisione può essere impostata su quella della forma di tutte le sotto-immagini di sprite combinate. Questo è di gran lunga, più lento. Tenere presente che le collisioni devono essere controllate 60 volte al secondo per ogni due oggetti che si trovano nelle vicinanze
In Game Maker, le collisioni semplici sono gestite dallo sviluppatore in due modi:

- Utilizzando l'evento di collisione per il quale lo sviluppatore può scrivere il codice di azione che si occuperà di cosa fare. Questo è il modo preferito quando le collisioni non sono usuali per un oggetto (per esempio una nave spaziale che viaggia nello spazio)
- Verifica di potenziali collisioni su azioni legate all'evento chiave. Questo è consigliabile se un oggetto è solitamente in collisione con un altro. Non utilizzare l'evento di collisione farà risparmiare un po' di tempo perché lo stesso può essere ottenuto dall'evento chiave controllando esplicitamente dal codice.

Game Maker include un motore di gioco fisico, che apre un modo totalmente nuovo di gestire i movimenti, le collisioni e l'interazione fisica tra gli oggetti. Quando si utilizza questa funzione, le collisioni sono totalmente gestite dal motore di gioco, semplificando enormemente lo sviluppo. L'inconveniente è che lo sviluppatore deve specificare le proprietà fisiche per ogni oggetto e stanza, e verosimilmente mettere a punto alcuni parametri per ottenere il risultato desiderato.

Co-funded by the Erasmus+ Programme of the European Union
The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Sprites - Sito web che fornisce l'accesso a un archivio di Sprites

Co-funded by the Erasmus+ Programme of the European Union

2.7 Esportazione e pubblicazione di videogiochi

L'esportazione del gioco si riferisce alla capacità di Game Maker di creare un programma in grado di funzionare ed eseguire sotto uno specifico hardware e software, mentre la pubblicazione è la disponibilità diffusa del gioco per essere acquistato dal grande pubblico. Precedentemente le due attività erano svolte da società totalmente diverse. Ma qui game Maker si occupa di entrambi i problemi, facilitando il percorso dello sviluppatore per far conoscere il suo lavoro ai giocatori.

La caratteristica principale di Game Maker è la possibilità di esportare su molte piattaforme di gioco diverse senza modificare nulla del gioco in fase di sviluppo. I target (piattaforme o sistemi per i quali lo sviluppatore del gioco può creare un gioco) includono Linux (Ubuntu), Mac OS X, Android, iOS, fireTV, Android TV, Windows desktop, Microsoft UWP, HTML5, PlayStation 4 e Xbox One. Ognuno di questi obiettivi richiede licenze specifiche. Game Maker può generare pacchetti che possono essere installati in uno qualsiasi dei sistemi sopra citati. Ma perché il gioco arrivi ai giocatori, dopo questo rimane ancora un lungo tratto da percorrere. Il gioco necessita di essere pubblicato, e diffuso, proprio come i libri

In questa era digitale, dove tutto è in rete, la pubblicazione di giochi ha subito una rivoluzione. I giochi stanno passando dal formato fisico a contenuti digitali scaricabili. I maggiori editori ora centralizzano la pubblicazione di questi giochi digitali. Il percorso di pubblicazione di un gioco è abbreviato per gli sviluppatori. I giochi possono essere disponibili per i giocatori non appena il loro sviluppo è completato. Gli sviluppatori di giochi hanno solo bisogno di un abbonamento alla piattaforma editoriale

Game Maker può pubblicare i giochi sulla piattaforma Steam. Con Steam i giochi possono usufruire di alcuni servizi forniti dalla piattaforma, come le classifiche, i contenuti scaricabili a pagamento, lo spazio di archiviazione su cloud, ecc.

Game Maker offre diverse licenze, in base alle piattaforme di destinazione ed anche in base all'abbonamento o all'integrazione con diverse piattaforme editoriali.

Co-funded by the Erasmus+ Programme of the European Union
The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Capitolo 3 - Casi pratici

3.1 - Introduzione ai casi pratici

In questo tutorial abbiamo un'introduzione a Game Maker operando su due casi pratici.
Entrambi gli esempi sono molto elementari e non richiedono conoscenze di programmazione precedenti.
Sono complementari e il secondo adotta un approccio diverso, per considerare concetti di base mancanti del primo caso.

3.2 - Creare un gioco Shoot'em up

Il primo caso pratico ha lo scopo di creare un gioco "shoot'em up" basato su un'astronave controllata dal giocatore.
Questa nave può sparare proiettili per distruggere i nemici, che a loro volta sono anche astronavi più piccole.
Le navi nemiche appariranno dalla parte superiore dello schermo e si muoveranno verso il basso e spareranno dove il giocatore di solito si trova.

Potete fare riferimento a questa guida passo dopo passo per capire il processo necessario per creare un piccolo gioco di "shoot'em up".

Co-funded by the Erasmus+ Programme of the European Union
The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

3.3 - Creazione di un gioco di piattaforma

Il secondo progetto mira a creare un gioco di piattaforma. In esso, la maggior parte degli stessi punti inizialmente affrontati nella prima attività vengono esaminati e approfonditi.
Alcune caratteristiche simili che entrambi i giochi condividono sono risolte con un approccio diverso, dando al lettore una più ampia gamma di metodi per ottenere la programmazione del gioco.
I giochi di piattaforma sono stati molto popolari nel corso degli anni. Per la maggior parte, sono giochi lineari che richiedono solo abilità nel muovere il personaggio, sparare, saltare e così via, all'interno di un mondo bidimensionale.
Per questa attività, andremo direttamente a costruire un livello minimo giocabile senza decorazioni, che verrà aggiunto in seguito.
Il giocatore sarà in grado di muovere un personaggio con soli tre tasti, sinistra, destra e spazio, che farà saltare il personaggio giocatore.

Potete fare riferimento a questa guida passo dopo passo per capire il processo necessario per creare un piccolo gioco di piattaforma.

3.4 - Un sistema autore per creare videogiochi e applicazioni interattive: Unity

Unity è una potente piattaforma 3D di sviluppo per videogiochi e applicazioni interattive, la più utilizzata al mondo anche a livello professionale. Si tratta quindi di uno strumento di lavoro estremamente diffuso, sia nel mondo dei videogiochi che in quello delle App, ma anche nella Realtà Virtuale ed Aumentata, nello sviluppo dei Punti informativi e nello scenario dell'Internet Of Things. Con Unity è possibile creare videogiochi in tre dimensioni per le piattaforme più diffuse (Desktop o mobile). E' disponibile una versione gratuita denominata personal (https://unity3d.com/get-unity/download), che può essere utilizzata agevolmente anche nelle scuole.

Il funzionamento di Unity è incentrato sulla **Scene View**, la rappresentazione dell'ambiente di gioco, che visualizza gli oggetti di gioco in maniera simile ai programmi di modellazione 3D. La Scene View contiene sia oggetti "tangibili" presenti in scena sia luci, camere e controller degli oggetti stessi. La scena è visibile sia su piani cartesiani sia in prospettiva.

Gli **oggetti** sono tutti gli elementi del nostro gioco. Telecamere, luci, modelli 3D, sprites, effetti particellari, elementi dell'interfaccia utente. Un progetto può essere composto da una o più scene. La Camera è la vista soggettiva sul videogioco. Nella modalità di gioco l'azione è ripresa esattamente da dove si trova la telecamera. In Unity è possibile creare oggetti semplici, ma per sviluppare oggetti complessi è bene operare con software di modellazione (come SketchUp) e poi importare i modelli

Co-funded by the Erasmus+ Programme of the European Union
The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

3d e le textures, che appariranno tra gli assets della scena, dove è possibile spostarli, scalarli e ruotarli. Gli oggetti presenti nella Scene View sono tutti da considerare “game object”.

I game object possono essere dotati di **funzionalità interattive** mediante un linguaggio di programmazione semplificato che permette di effettuare interazioni di base. Ogni Game Object ha un nome, alcune proprietà di base, può essere posizionato nello spazio 3D e dotato di script. I Game Objects possono essere di tipo molto diverso: possono essere visibili o invisibili, emettere suoni o essere interattivi. Mentre la Scene View offre una visuale operativa sulla scena 3D nella quale si svolge il gioco, per visualizzare quello che i giocatori vedranno durante l’azione è disponibile la Game View, che mostra il punto di vista della videocamera di gioco.

Quando sul Game panel viene premuto **Play**, Unity inizia ad eseguire il gioco, che può infine essere esportato tramite la funzione build, in un gran numero di piattaforme e device, compresi PC, smartphone, tablet e visori di realtà aumentata e virtuale.

Co-funded by the Erasmus+ Programme of the European Union
The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.