PROYECTO V4T

CURSO E-LEARNING

Capítulo1 - Introducción a la programación de videojuegos.

Tecnologías disponibles - Part 1

La creación de videojuegos incluye varias áreas de trabajo y conocimiento, desde creación artística hasta programación a bajo nivel de computadores. En cada área, existen herramientas de ayuda o desarrollo y tecnología especializada. Muchas de estas herramientas pueden ser utilizadas en solitario, dando a los equipos de creación de videojuegos la posibilidad de trabajar por separado en la mayor parte del proceso de creación. En las últimas fases, todo el trabajo de los miembros del equipo se integra en un producto final. Para lograr esta integración, el equipo de desarrollo utiliza programas que toman todos los elementos del juego y generan el producto final. Estas herramientasson capaces de manipular imágenes, vídeos, sonidos, junto con un conjunto de reglas o programa y generar el Videojuego en sí.

Básicamente, crear un juego es programar, como se programa cualquier otro software. Sin embargo, conforme el software evoluciona y la especialización se acrecienta, las diferencias entre el desarrollo de juegos y de otro software se agrandan. Y las herramientas para asistir a los desarrolladores de software evolucionan y se especializan también. Frecuentemente, las empresas de software ofrecen nuevas herramientas para asistir a los desarrolladores de software, desarrolladores web, de software empotrado, etc. con promesas en mejoras, manejo más simple y mayor eficiencia en cada sector. Ocurre lo mismo en las herramientas de asistencia a la creación de videojuegos

Sin embargo, a diferencia de otros sectores de software, las tecnologías de ayuda al desarrollo de videojuegos van desde herramientas muy especializadas a otras muy abiertas. Hay herramientas que pueden usarse para crear juegos complejos de varios géneros, mientras que otras tienen como objetivo automatizar la creación de juegos de un tipo concreto. Existen incluso juegos como tal, que incorporan las funciones necesarias para expandir o ampliar el juego con nuevos personajes, niveles; siendo estos juegos una mezcla entre juego y herramienta de desarrollo.
Adicionalmente, el desarrollo de videojuegos comparte tecnologías clave con otras áreas de la ciencia y de la informática. Por ejemplo, las imágenes tridimensionales son una gran parte de la revolución en los videojuegos y su fundamento es también la base de la fabricación y diseño asistidos por ordenador (CAD/CAM), cirugía asistida por computador, visores de mapas y entorno

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.

(google Earth), y otros.

1.2 - Tecnologías disponibles - Parte 2

Hay una serie de campos científicos e informáticos que están directamente relacionados con la creación de videojuegos:

- Ingeniería del Software: como cualquier software, los videojuegos son desarrollados por equipos de personas que deben seguir las directrices y tecnologías de la ingeniería del software.
- Lenguajes de programación, que resultan esenciales para crear cualquier programa complejo.
- Imágenes bidimensionales y tridimensionales. Los videojuegos son el software más intenso en lo que a mostrar imágenes, animaciones, escenas, etc. Esto ha empujado a los fabricantes de Hardware a desarrollar tarjetas gráficas cada vez más potentes que permiten representar escenas complejas con movimientos suaves y fluidos. Como resultado, se han creado estándares para facilitar el uso de este hardware. El más universal es "OpenGL". Adicionalmente el software para modelar en el computador objetos tridimensionales ha madurado y se ha hecho más simple de utilizar.
- Inteligencia artificial y aprendizaje automático. Jugar contra el computador requiere que los desarrolladores doten a los videojuegos de una lógica para desafiar al jugador. Habitualmente, los videojuegos tienen "niveles de dificultad" para ajustar la dificultad ofrecida a la destreza del jugador. Existe, sin embargo, un esfuerzo por dotar a los videojuegos de la capacidad de adaptarse automáticamente a la destreza mostrada por el jugador, o para aprender automáticamente y batirlo.

Adicionalmente no podemos olvidarnos de otras tecnologías, conocimiento y desarrollos que soportan el trabajo no informático en el desarrollo de los videojuegos.

- Diseño de cómics, gráfico y cine. Los videojuegos tienden a representar a los personajes y el entorno de una forma impactante para producir en el jugador una experiencia intensa. Hay tecnologías y técnicas emergentes en la creación de películas que se aplican así mismo en los videojuegos. Los juegos imitan frecuentemente los planos y escenas cinematográficas.
- Psicología. Desde cierto punto de vista, jugar es evadirse de la realidad. Los diseñadores de videojuegos triunfan si su videojuego logra ser absorbente y evade al jugador del mundo en el que realmente vive. Algunos juegos van más allá e incitan a los sentimientos del jugador, tristeza, ira, miedo, etc. Los videojuegos toman lecciones de la psicología para cumplir mejor estas intenciones.
- Sociología. Los videojuegos multijugador masivos, explotan la sociabilidad de los jugadores

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.

y aprovechan la motivación adicional que tiene un jugador al saber que está jugando o compitiendo con o contra otros humanos. El juego se torna un instrumento para encontrar a otras personas (mientras se juega). La manera en la que el jugador interacciona es una cuestión de análisis, cómo hace amigos, enemigos, cómo planifica las partidas, etc. Hay elementos científicos de la sociología en estos videojuegos. Y las conclusiones obtenidas se aplican a nuevos juegos multijugador.

1.2 Entornos de desarrollo y lenguajes de programación – Parte 1

Llamamos "Entornos de desarrollo" o "Entornos de desarrollo integrados" (IDE a partir de ahora) a un software especial que ayuda a los desarrolladores a construir software y programas. Los IDE básicos incluyen un editor de código, un compilador y un depurador; aunque los IDE modernos incorporan otras herramientas de ayuda. Al tiempo que los diferentes tipos de software se vuelven más diferentes entre sí, los IDE replican esta diferenciación. Por tanto existe un amplio abanico de IDE que abarcan desarrollo de software en diferentes áreas. Los IDE para el desarrollo de videojuegos son de los más especializados de todos, con varias particularidades. Un elemento central en la concepción de un IDE es el lenguaje de programación. Algunos IDE están concebidos alrededor de cierto lenguaje, mientras que otros admiten ser usados con diferentes lenguajes de programación y compiladores.

Los entornos de desarrollo de videojuegos tienen varias características especiales, pero dos son realmente importantes. La primera es la integración de diferentes herramientas destinadas a crear distinto contenido del juego (gráficos, programas, sonidos, interconexión de redes, etc). Y la segunda gran característica es la especialización en el desarrollo de juego en que muchos de los elementos de un juego ya están prefabricados y previstos (niveles, mapas, personajes, animaciones, etc)

Un IDE para producir videojuegos debe permitir al desarrollador programar diferentes partes del juego con diferentes propósitos. Para soportar distintos requisitos, el IDE puede permitir al programador utilizar diferentes lenguajes. Por ejemplo, un juego de acción multijugador en primera persona necesita hacer uso intenso de la programación en 3D y además gestionar las comunicaciones a través de la red con algún ordenador remoto. Podríamos necesitar lenguajes de programación radicalmente diferentes para estas dos funciones del mismo juego. Adicionalmente el juego tendrá reglas sobre qué acciones están permitidas y prohibidas al personaje, o puntos de control por los que debe pasar. Este requisito necesita otro lenguaje de alto nivel donde estas reglas puedan ser fácilmente escritas. Considere también que el juego puede tener "bots"; esto es: jugadores virtuales gestionados por el propio juego. ¿Cómo se programa el comportamiento de los "bots"? Necesitamos lenguajes específicos para este requisito.
Muchos IDE incorporan su propio lenguaje de programación intentando cumplir diversos requisitos y

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.

tareas, al tiempo que se esfuerzan por mantener una simplicidad agradable; mientras que otros permiten el uso de lenguajes populares y asentados (típicamente C++) como los protagonistas del desarrollo. Los últimos no restringen el uso de otros lenguajes, sin embargo, ya que hay herramientas y formas de empotrar partes de un programa escrito en otro lenguaje en lenguajes genéricos como C++. Pero incluso lenguajes como C++, rust, python o lua, se amplían con nuevas funciones de bibliotecas que están fuertemente ligadas al uso particular de un IDE o motor de juego.

1.3 Entornos de desarrollo y lenguajes de programación – Parte 2

Una alternativa interesante a utilizar un IDE es utilizar un editor de niveles. Algunos juegos son distribuidos con editores incorporados que permiten modificar o ampliar el propio juego. Esta es la herramienta que los propios creadores del videojuego han utilizado para crear los niveles, mapas, fases del juego.

El jugador puede utilizar libremente el editor para crear sus propios niveles, incluyendo elementos gráficos, imágenes, sonidos, etc. Algunos editores permiten realizar cambios mínimos, pero otros se usan para modificar fuertemente el juego.

Los géneros de juegos que usualmente incorporan un editor de niveles son "estrategia en tiempo real" y juegos de acción en primera persona. Desde cierto punto de vista, un juego editable o expandible contiene los siguientes elementos

- Un conjunto de recursos como imágenes, sonidos, animaciones, etc. que componen el aspecto básico de los objetos del juego.
- Un conjunto de niveles, fases o etapas que son uno en los muchos desafíos que el jugador debe superar para completar el juego
- Un programa que toma los elementos anteriores y genera el juego real integrando mapas y recursos. Este elemento es llamado "El motor del juego"

Para crear un juego puede que sólo necesite elegir un videojuego con un editor integrado, y entonces, diseñar los niveles y recursos para ellos. Muchos, si no todos, los editores incluyen algún lenguaje de programación que permiten especificar el comportamiento de diferentes elementos del juego (enemigos, proyectiles, "power-ups", etc)

Esta aproximación a la creación de juegos ha sido llevada más allá. Para crear juegos realmente complejos, los programadores establecen dos fases con diferentes productos. Inicialmente crean un "Motor de Juego" que puede hacer ejecutar (o correr) un juego pero los contenidos del juego son creados por separado. El Motor está emparejado con un editor especializado en generar niveles, gráficos, personajes, etc. En una segunda fase se usa este editor para dar contenido al juego y

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

completar la historia y sus niveles.

Cuando el juego está terminado, hay dos productos disponibles para el público: El juego en sí, con todo su contenido y el motor que lo ejecuta. Y el editor que puede ser usado por entusiastas y aficionados para crear nuevo contenido. Ambos se pueden vender por separado dando como resultado que algunos desarrolladores utilizan el editor y el motor para desarrollar otro juego completamente diferente al inicial. Este editor y el motor, no pueden considerarse una herramienta de "propósito general" ya que están orientadas a un tipo de juego concreto desde el inicio. Esto es diferente de simplemente "un videojuego con un editor integrado". Hay un buen y creciente conjunto de Motores de Juego.

1.4 Entornos de desarrollo y lenguajes de programación – Parte 3

Finalmente, y debido a la tendencia de otras tiendas online de software, como "Steam" de Valve, Play de Google, PlayStation Store de Sony, etc. Algunos IDE facilitan la publicación digital de los videojuegos. El IDE puede utilizar alguna cuenta del desarrollador en varias de estas tiendas de software para publicar el juego sin complicaciones

Como conclusión a este punto, debemos considerar la conveniencia de utilizar alguna herramienta de desarrollo que esté orientada a la simplicidad y dedicada a los principiantes con poca destreza previa en programación. Y de la larga listas de software disponible, GameMaker del desarrollador "Yo Yo Games" es una muy buena herramienta para este proyecto. Es una solución probada con una comunidad grande de usuarios y varios casos de éxito.

Capítulo 2 - Game Maker

2.1 Características del entorno Game Maker. Tipos de Juego.

"GameMaker Studio" ("GameMaker" en este tutorial) es un entorno completo e integrado para desarrollar juegos. Inicialmente fue concebido para permitir a no iniciados crear juegos simples sin tener conocimiento de programación. Para cumplir este objetivo, ofrecía diferentes tecnologías para componer visualmente el equivalente a un programa escrito de computador. Aunque si es necesario, el desarrollador puede escribir código utilizando un lenguaje similar a C con algunas ideas tomadas de javascript. Esto también resulta en un lenguaje que es suficientemente fácil para no iniciados.

Sus principales características son:

- Integrado. Todas o la mayoría de etapas en el desarrollo pueden ser realizadas con

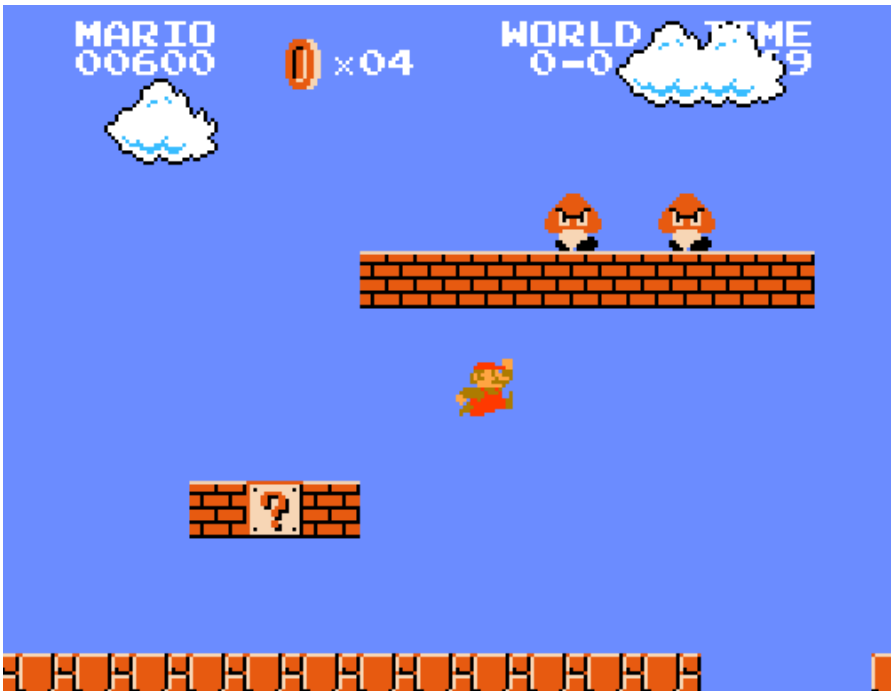
Co-funded by the Erasmus+ Programme of the European Union
The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made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

herramientas que Game Maker incorpora, sin ayuda de terceros programas.

- Tiene su propio lenguaje ("Game Maker Language" o GML), que es familiar a usuarios con conocimientos de C, javascript y otros. Está fuertemente integrado con el motor del juego y con el entorno de desarrollo
- Para juegos simples no es necesario escribir código. Game Maker permite a los desarrolladores utilizar "DnD" ("Drag and Drop" o "Arrastrar y soltar") como alternativa al lenguaje GML.
- Los juegos pueden crearse para varios Sistemas Operativos y plataformas. La integración con plataformas digitales de distribución también está resuelta.
- Está especialmente orientado a algunos géneros o tipos de juego, pero se puede adaptar para la creación de otro tipos de juego con un poco de trabajo extra.

Game Maker es excelente en la simplicidad y facilidad para crear juegos de plataforma y de tipo "shoot'em up". Aunque otros tipos de juego también se contemplan

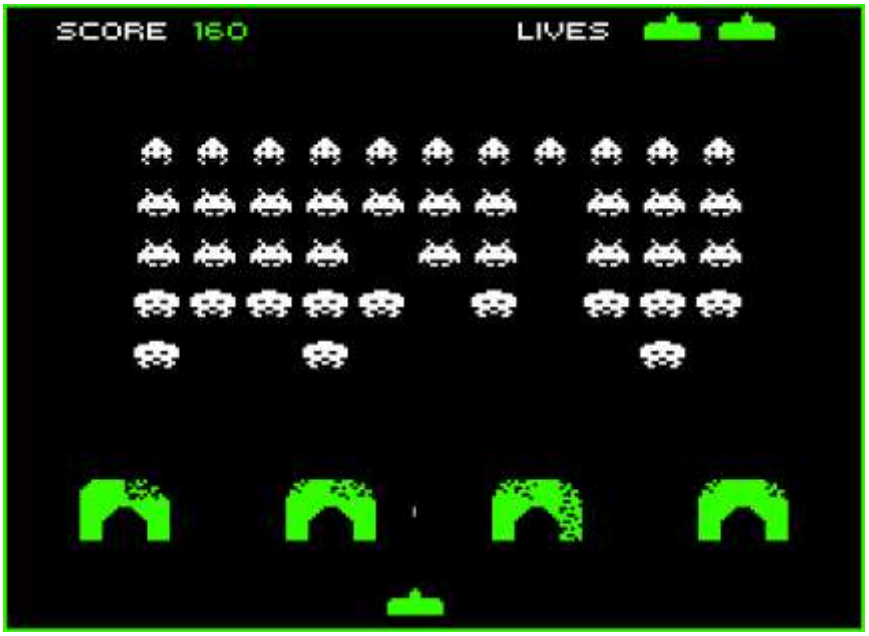
- En los juegos de plataformas, el personaje se mueve en un entorno bidimensional, capturando objetos, evitando obstáculos y destruyendo enemigos. Un ejemplo muy popular es "Mario Bros"



- En los juegos "Shoot'em up", el jugador está contínuamente avanzando en un mapa, bien volando o caminando. Puede desplazarse y alcanzar diferentes zonas de la pantalla. Los enemigos aparecen desde la zona opuesta y en dirección contraria, avanzando hacia el jugador, quien tiene que evitarlos o matar esos enemigos. Un ejemplo eterno es "Space

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.

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.

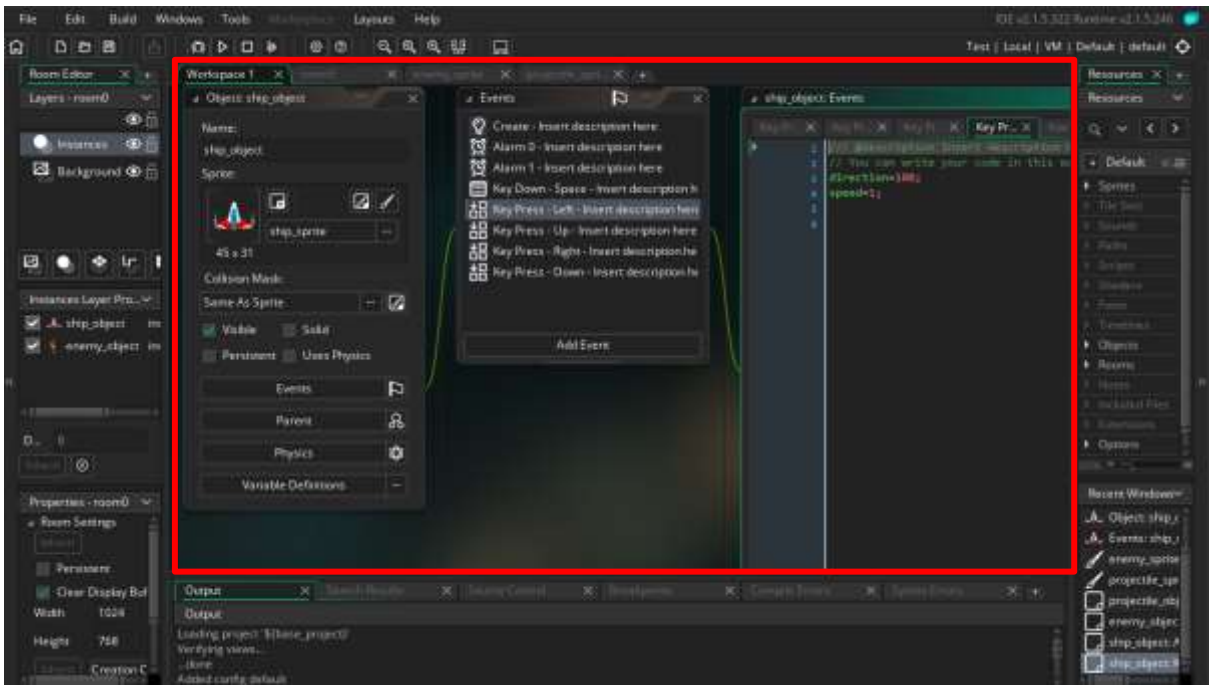
2.2 Entorno de desarrollo

Game Maker es un entorno completo de desarrollo. Incluye todas las herramientas necesarias para desarrollar y probar juegos sin ayudas externas, dispositivas o configuraciones complejas. El interfaz de usuario está diseñado alrededor de un marco central que permite alojar diferentes áreas de trabajo. Estas áreas de trabajo son paneles o tableros donde se colocan y organizan los recursos del juego durante el desarrollo y que permiten una personalización. Las áreas de trabajo incluyen ellas mismas ventanas internas con diferente contenido, desde objetos genéricos y sus propiedades, hasta listas de eventos y código GML. Todo elemento de ventana es parte o está enlazado con algún otro elemento.

Adicionalmente, alrededor del área central de trabajo, existen paneles secundarios. Muchos de los cuales son enrollables para permitir expandir el área central. Estos paneles muestran:

- Una lista en forma de árbol de todos los recursos del juego. Este árbol aparece siempre y cambia cuando se añaden o eliminan recursos al juego.
- Una lista de controles para cambiar las propiedades de un recurso concreto. Estos paneles cambian acorde al objeto seleccionado..
- Una ventana que muestra el resultado del juego final cuando se compila y ejecuta.

Otras ventanas pueden aparecer solitariamente, encima de las restantes como una pestaña o como parte del área central de trabajo cuando se solicita alguna herramienta. Por ejemplo, el editor de imágenes es raramente usado, pero cuando se decide editar una imagen, se activa y trae al frente como una pestaña más en el área central.



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

Game maker ofrece dos maneras de crear un juego::

- Drag and Drop ("DnD"). Esta forma está orientada a no iniciados que no saben o no quieren programar y pretenden crear juegos estándar. Game Maker permite la definición de la lógica del juego utilizando una herramienta visual . De esta forma, el desarrollador, arrastra y suelta elementos que están clasificados y especifican el comportamiento de cualquier elemento.
- Proyectos hechos con Game Maker Language. Este es el modo por defecto y estándar. La lógica del juego y de todos sus elementos está definida como programa escrito en GML.

En este tutorial utilizaremos la opción del lenguaje GML, porque la complejidad de los juegos hechos es poca y consecuentemente no necesitaremos una programación compleja pero al terminar tendremos un futuro con más posibilidades.

2.3 Sprites, rooms, objects y collisions.

Los juegos muestran diferentes elementos en una pantalla cambiante y dinámica. Lo que el jugador ve son simplemente "cosas" moviéndose de aquí a allá dentro de un área o "mundo". Habitualmente estos elementos chocan o colisionan con otros elementos. Tanto Game Maker, como los desarrolladores y los jugadores tienen nombres específicos para cada elemento visual. El área o "mundo" donde se desarrolla el juego se llama "room" en Game Maker ("habitación" o "estancia"). Las animaciones del juego se llaman "sprites". Las imágenes no animadas, estáticas o con fines decorativos son el "background" ("fondo"). Cualquier elemento del juego que tenga algún comportamiento o influencia lleva el nombre de "objeto". Finalmente, "colisión" es el nombre habitual del evento que ocurre cuando dos objetos se encuentran físicamente (dentro del juego). Estos elementos son la base de muchas características de Game maker..

Sprites, rooms, objects and collisions - An in depth guide addressing the main elements of a Game

2.4 Eventos y acciones

A game, when running or being played, is a dynamic system. It is something that changes, often quickly from the player point of view. These constant change in the game is actually the result of small very frequent changes, perceived as a greater, more complex turn of events. The key to

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.

determine how the game evolves is to understand the basics changes and how establish them from their components called "events" and "actions".

Para entender mejor los eventos y acciones, debemos primero considerar que un videojuego es realmente un programa que se está ejecutando contínuamente. A veces se conoce a este programa como "Motor del juego". El programa en ejecución está continuamente monitorizando la actividad del teclado, ratón y otros dispositivos; es consciente del paso del tiempo, sabe la posición de todos y cada uno de los objetos del juego,etc. Por tanto, El motor del juego es el primero en saber que algo ha ocurrido, tanto si es una tecla que se ha pulsado o si ha transcurrido un lapso de tiempo, o bien dos objetos han colisionado en la pantalla, etc. Game Maker denomina "eventos" a estas cosas que pueden terminar desencadenando cambios en el juego. Muchos de los eventos son el resultado de la interacción del jugador con la computadora. Otros eventos ocurren por la evolución del juego por sí mismo.

El motor del juego tiene el control del mismo y puede cambiar el estado de cualquier objeto, en cualquier momento. Puede reposicionar un objeto, añadir daño al mismo, crear un nuevo objeto, etc. Estos cambios ocurren tal como lo planeó el programador mediante la programación realizada. El motor del juego es tan sólo el elemento que aplica la programación. Game Maker denomina "acciones a los cambios especificados por el desarrollador del juego. La clave aquí es que todas las acciones ocurren como resultado de un evento, incluso acciones aleatorias, están enlazadas a eventos con un componente de aleatoriedad en ellos. El objetivo del desarrollador es establecer qué acciones ocurren como consecuencia de qué eventos. Analizaremos primero los eventos y después veremos como se establecen acciones asociadas a ellos

Existe una lista larga de eventos preestablecidos en Game Maker, pero el desarrollador es libre de asociar cualquier acción a cualquiera de ellos.

Events and Actions - Una guía que trata en profundidad los eventos y acciones en Game Maker

2.5 Acciones condicionales

En ocasiones, las acciones deben ser llevadas a cabo bajo condiciones complejas para las cuales no existe un evento predefinido. En estos casos, el programador necesita escribir una condición particular bajo la cual se ejecutará la acción. El motor del juego deben verificar la posibilidad de que se cumpla la condición a cada momento. Cuando ello ocurre, la acción se ejecuta. Esto ofrece gran flexibilidad al desarrollador. Sin embargo, las acciones condicionales son en última instancia acciones que necesitan ser enlazadas a algún evento (así es como funciona Game Maker). Si la condición necesita ser verificada a cada instante, entonces, el evento natural al que asociarse, es el evento

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.

"Step". Y la acción es simplemente la comprobación de la condición que se debe verificar.

Acciones Condicionales - Ampliación del tratamiento de acciones condicionales

2.6 Animation and advanced collision

La animación en videojuegos es crucial para una experiencia inmersiva. Desde el inicio de los videojuegos, los programadores y artistas han trabajado duro para producir las apariencias más realistas y atractivas en Juegos. Esto requiere replicar el movimiento que muchos objetos tienen en la ivda real. Por ejemplo, conducir un coche viéndolo desde arriba o desde detrás, no sólo debe resultar en percibir su movimiento por la carretera, sino en ver las ruedas girar, el humo salir del tubo de escape, y la carretera sacudiendo el coche o las ruedas. Hay diferentes técnicas de efectos de animación en videojuegos, algunas de las cuales están integradas adecuadamente en Game Maker.

La técnica más extendida para producir animaciones, resulta de utilizar un conjunto de imágenes similares que se van mostrando alternadamente de forma cíclica. Si el ritmo al cual cambian las imágenes es el adecuado, el jugador percibirá dichas imágenes como una animación. La dificultad reside en cómo dibujar u obtener dichas imágenes.

Consideramos que un sprite tiene sub-imágenes. Tome por ejemplo, la siguiente tira de imagens. El sprite mostraría una chica a la vez, cambiando a la siguiente en la tira de una forma acelerada, dando la impresión de que la chica está moviendo sus piernas (la posición de la imagen también cambiaría)

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

Game Maker es muy capaz en la creación y utilización de sprites. Puede leer una imagen como la anterior de un fichero, y trocearla fácilmente en diferentes sub-imágenes, creando así el sprite casi automáticamente. Puede manipular imágenes cuyas subimágenes están organizadas en otras disposiciones (como una matríz)

La utilización de sprites en Game Maker da buenos resultados y es muy capaz. Pero crear los sprites es otra historia. Para iniciarse en la creación de juegos, existen muchos repositorios de sprites en

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.

sitios web. Uno de ellos es http://opengameart.org.
Los sprites determinan la forma visual del objetos y dependiendo del tamaño relativo a la room puede ser inaceptable manipular colisiones simplemente a partir de las coordenadas de la posición de los objetos. Por ejemplo, considere un objeto pequeño moviéndose hacia un objeto grande. Muchas de las veces, la colisión debe evitar que unos objetos se superpongan con otros o penetren dentro de ellos. Por tanto la colisión no debe tomar únicamente la posición de los objetos sino la forma y dimensiones

Cuando se define un sprite, también es útil y directo establecer la "máscara de colisión" del sprite, la cual se aplicará a los objetos que usen este sprite. La máscara de colisión es un área de una forma potencialmente arbitraria. Típicamente y por razones de eficiencia de cómputo, la máscara de colisión es un rectángulo que puede rotarse. Círculos y elipses también pueden elegirse como máscaras de colisión, pero su uso requiere más cálculos. Finalmente una máscara de colisión puede ser definida a partir de la superposición de todas las sub-imágenes que componen el sprite. Esto es, evidentemente, la opción más lenta. Tenga en cuenta que las colisiones deben ser verificadas 60 veces por segundo para todos los objetos de la room

En Game Maker, las colisiones simples están gestionadas por el programador de dos formas

- Utilizando el evento colisión, para el cual el programador escribe el código de la accion que decidirá qué hacer. Esta es la manera preferida cuando las colisiones no son habituales para un objeto (por ejemplo en una nave espacial que se mueve por el espacio)
- Verificando posibles colisiones en acciones asociadas al evento "step". Esto es preferible si el objeto está continuamente colisionado con otros. De esta forma ahorramos algo de tiempo.

Game Maker incluye un motor de física del juego, que abre un mundo de posibilidades mayor para manipular movimientos, colisiones e interacciones físicas entre objetos. Cuando se utiliza esta característica, las colisiones son totalmente gestionadas por el juego, simplificando enormemente el desarrollo. La contrapartida, es que el desarrollador debe especificar las propiedades físicas de cada objeto y de la room; y probablemente ajustar algunos parámetros para obtener el resultado deseado

Sprites - Website que proporciona un repositorio de sprites

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.

2.7 Exportación y publicación de videojuegos

La "Exportación" de videojuegos se refiere a la habilidad de GameMaker the producir un programa capaz de ser ejecutado bajo un hardware y software concreto, mientras que la "publicación" es la difusión del videojuego y la disponibilidad del público para aquirirlo y utilizarlo. Tradicionalmente estas dos actividades han sido llevadas a cabo por empresas totalmente diferentes. Pero aquí, Game Maker facilita estas dos etapas, facilitando el camino del desarrollador para lograr que su juego alcance al público

Una característica fundamental de Game Maker es la capacidad de exportar el juego a diferentes plataformas, hardware y software sin alterar nada el desarrollo del videojuego. Los destinos (plataformas para los que se puede hacer disponible el juego) incluyen Linux (Ubuntu), MacOs X, Android, iOS, fireTv, AndroidTv, Windows desktop, Microsoft UWP, HTML5, Playstation 4 y Xbox One. Cada una de estas plataformas puede requerir licencias específicas. Game maker puede producir paquetes instalables para cada uno de esos sistemas. Pero, para lograr que el público utilice el juego, éste necesita ser publicado, al igual que los libros.

En esta era digital, donde todo está en red, la publicación de juegos ha sufrido una revolución. Hoy en día los juegos han abandonado el formato físico (Discos) por los contenidos descargables. Las principales editoriales centralizan la publicación de juegos digitales y el camino a los desarrolladores se abrevia. Los juegos pueden ser publicados tan pronto su desarrollo finaliza. Sólo se necesita disponer, como desarrollador, de una licencia en la plataforma elegida

Game Maker puede publicar directamente juegos en la plataforma Steam de Valve. Lo cual le permite aprovecha algunas características de la misma, como ranking de jugadores, contenido descargable de pago, almacenamiento en la nube, etc.

Game Maker ofrece varias licencias dependiendo de las plataformas destino

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

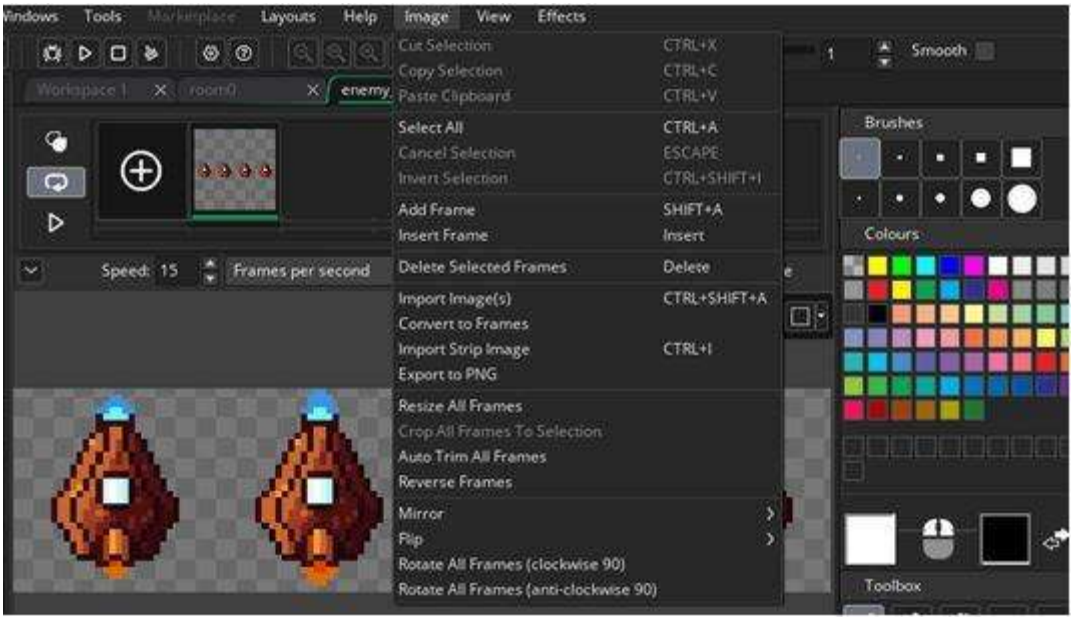
Capítulo 3 - Casos prácticos

3.1 - Introducción a los casos prácticos

En este tutorial vamos a ver una introducción a Game Maker mediante la práctica con dos casos de ejemplo. Ambos son muy básicos y no requieren conocimiento previo den programación. Son casos complementarios y el segundo sigue una aproximación diferente al primero para cubrir sus carencias.

3.2 - Crear un juego Shoot'em up

El primer caso práctico tiene la misión de crear un juego de tipo "shoot'em up" basado en una nave espacial controlada por el jugador. Esta nave puede lanzar proyectiles para destruir enemigos que resultan ser también naves espaciales más pequeñas. Éstas aparecerán por la parte superior de la pantalla y se moverán hacia abajo donde se encuentra el jugador



Puede referirse a la guía que describe el proceso paso a paso para comprender totalmente le proceso necesario para crear un pequeño juego

3.3 - Crear un juego de Plataformas

El segundo caso práctico pretende crear un juego de tipo "Plataformas". En él muchos de los puntos tratados en el primer caso práctico son revisitados y tratados con más detalle. Algunas características compartidas de ambos

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.

casos son resueltas con una aproximación distinta, dando al lector un rango más amplio de maneras de solucionar futuros problemas de programación.

Los juegos de plataforma han sido muy populares a lo largo de los años. En su mayor parte son juegos lineales que requieren algo de habilidad moviendo el personaje, disparando , saltando y demás a lo largo de un mundo bidimensional.

Para esta actividad, se irá directamente a construir un nivel mínimamente jugable sin decoración, la cual será añadida más tarde. El jugador controlará el personaje con sólo tres teclas, izquierda, derecha y espacio, que logrará hacer saltar al personaje

Refiérase a esta guía para entender el proceso necesario para crear un juego sencillo de plataformas.

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