



Componente formativo

Verificación y ajustes de funcionalidad

Breve descripción:

El componente de formación presentado describe la forma de verificar el funcionamiento del prototipo digital elaborado, aplicando metodologías ágiles y herramientas propias del motor de Unity

Área ocupacional:

Ciencias Naturales

Junio 2023

Tabla de contenido

Introducción.....	3
1. Diseño del ciclo de iteraciones – backlog	4
2. Prototipado digital con paquetes prefabricados	14
3. Métricas del videojuego a partir de impresiones de detalles de eventos (debug.log).....	17
3.1. Métricas de Unity	18
3.2. Método debug.log	21
4. Verificación del videojuego	24
4.1. Verificación basada en el modelo SCRUM	24
4.2. Verificación basada en el modelo SUM	25
4.3. Verificación y ajustes en el entorno de desarrollo (consola, modo edición)...	28
Síntesis	30
Material complementario	31
Glosario.....	32
Referencias bibliográficas	33
Créditos.....	34

Introducción

Uno de los aspectos más importantes para el desarrollo de los videojuegos es verificar que cumple con las funcionalidades diseñadas en el concepto gráfico. El motor de Unity contiene una función de verificación que se incorpora cuando se instala el programa y es utilizada con mayor efectividad cuando el desarrollador crea las líneas de código para comprobar una mecánica específica, igualmente posee el servicio de análisis que verifica el funcionamiento actual en aras de poder hacer las mejoras y ajustes que requiera el juego.

No obstante es importante no solo contar con las herramientas de software sino también con las técnicas que permitan llevar a cabo comprobaciones por cada entregable del producto para ajustar inmediatamente lo que no está de acuerdo con las especificaciones iniciales o con las que van surgiendo a medida que evoluciona el producto.

En este contenido formativo se abordarán las metodologías ágiles SCRUM y SUM como técnicas para verificar con eficacia los requisitos establecidos, mecánicas o niveles de los videojuegos como se explica en el siguiente video:

VIDEO INTRODUCTORIO

Carpeta Anexos: CF010 Video Introduccion

¡Buen aprendizaje!

Desarrollo de contenidos

1. Diseño del ciclo de iteraciones – backlog

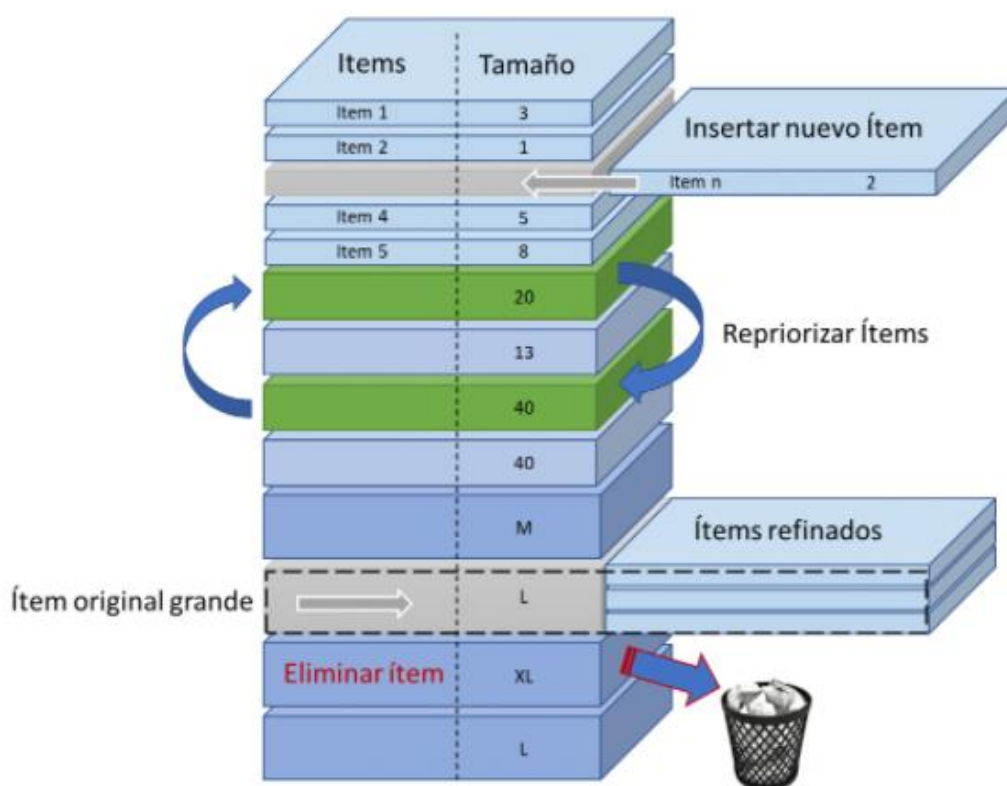
Según García (2019) backlog es una palabra de origen inglés que significa una pila de tareas almacenadas que están esperando para ser evacuadas, igualmente, este término es usado cuando se define el ciclo de vida de un sistema de información en sus fases de análisis, diseño, desarrollo e implementación, también está asociado a metodologías de trabajo ágiles para el desarrollo y entrega de productos en menor tiempo y con mayor eficiencia.

En el contexto del método ágil el diseño de los ciclos de trabajo o iteraciones se acuña el término “Product backlog”, el cual García define como una lista ordenada y priorizada de tareas a ejecutar en un periodo de tiempo corto para completar un entregable del producto y los ciclos se repiten hasta que se obtenga el producto final.

El backlog que se observa en la Figura 1, está ordenado de acuerdo con las funciones o requisitos más importantes considerados por el cliente o dueño del producto y colocados en la parte superior del backlog, dejando en la base los de menor relevancia:

Figura 1

Backlog



Nota. Adaptado del Conjunto de tareas priorizadas del backlog, García (2019).

<p>Durante la iteración se pueden cambiar las prioridades de las tareas según el valor que aporten al cliente las diferentes funciones que se estén desarrollando en el ítem (tarea).</p>	<p>El tamaño de los ítems (requisitos) depende del grado de detalle o refinamiento proporcionado por el cliente, ya que se deben ejecutar muy pronto, entre más pequeños sean mayor es el refinamiento o detalle, por eso los de tamaño más grande están en la parte inferior.</p>	<p>Durante el ciclo iterativo se puede decidir que un requisito no se va a ejecutar y se puede eliminar tranquilamente de la pila de requisitos o adicionar uno que no se había tenido en cuenta para completar la entrega.</p>
---	--	---

El backlog se materializa en un documento que contiene los requisitos necesarios para obtener el producto final esperado por el cliente. Los aspectos que lo caracterizan se pueden resumir a continuación:

Los ítems o requisitos considerados se deben colocar en una lista en orden de importancia para el cliente, siendo los primeros los de mayor relevancia.

El backlog se puede modificar a lo largo del ciclo iterativo adicionando o eliminando ítems de acuerdo con las prioridades que vayan surgiendo durante el proceso de desarrollo.

El refinamiento de los ítems depende de los detalles proporcionados por el cliente, siendo los más refinados aquellos que se colocan al comienzo de la lista.

El tamaño de los ítems depende del grado de refinamiento (detalles) que posean, siendo los más pequeños aquellos que estén provistos del mayor número de detalles y por el contrario, los de mayor tamaño son los de menor refinamiento.

Cada ítem del backlog debe asociar la función que se va a desarrollar para cumplir el requisito en un lenguaje claro y entendible por el cliente.

Cada ítem del backlog debe ser independiente uno del otro para poder cambiar el orden de ejecución si así se requiere.

Cuando se desarrolla un videojuego utilizando metodologías ágiles como por ejemplo SCRUM se debe tener en cuenta:

Lo primero que se diseña para empezar las tareas es el Product backlog; que en este caso sería la lista priorizada de las características principales del juego.

El Product Owner (PO) es el que decide la prioridad de dichas características, ya que es el responsable ante el cliente de la calidad del producto terminado. Para establecer dicha prioridad es necesario identificar los aspectos más importantes para el jugador en su experiencia con el producto.

El Scrum Team (SC) o equipo desarrollador del videojuego establecerá los ítems de la lista que se trabajarán en cada ciclo iterativo que en adelante se llamará Sprint; por lo tanto, cada Sprint es el tiempo de duración de cada ciclo de tareas que puede ser de 1 a 4 semanas, dependiendo del grado de complejidad de los ítems.

Antes de comenzar cada Sprint, el Scrum Team revisa la lista de ítems a desarrollar, selecciona los ítems del Sprint definiendo su objetivo y cada persona del equipo asume una responsabilidad sobre los componentes de cada ítem que se compromete a desarrollar, colocando estos elementos en una nueva lista llamada Sprint backlog.

Cuando se desarrolla un proyecto de un videojuego, no necesariamente se hace una entrega después de terminar un Sprint; puesto que pueden necesitarse varios Sprints para lograr terminar una funcionalidad o mecánica del juego, de manera que tenga sentido para el jugador que lo va a probar.

Cada vez que se realiza una entrega se pueden determinar los bugs o errores que puede arrojar el producto cuando sea comprobado por el cliente, lo cual garantiza que antes de que esté 100% terminado debe contener el mínimo de fallos.

En el siguiente gráfico se observa el ciclo completo de un Sprint en la metodología SCRUM:

GRÁFICO INTERACTIVO

Anexo: CF010_1_ciclo iterativo

Para la elaboración de los Product backlog y los Sprint backlog se pueden utilizar diferentes maneras:

a. Se puede diseñar un formato para elaborar el Product backlog, el cual debe indicar claramente la prioridad y la descripción del ítem que debe desarrollar el Scrum Team, observar el ejemplo de la tabla a continuación:

Tabla 1

Ejemplo Product Backlog

Nombre del videojuego: La isla de los tesoros escondidos	
P rioridad del ítem	Descripción

1	Interfaz principal donde se visualice la isla y los activos que la componen, colocación de los tesoros en el nivel más bajo para que no se vean.
2	Diseño del personaje principal que buscará los tesoros escondidos. Crear animaciones con acciones de correr, saltar, mirar mapa, caer, caminar.
3	Mapa de navegación que es consultado por el personaje para ubicarse en la isla.
4	Efecto del día y de la noche.
5	Diseño de animales feroces que atacan al personaje principal, movimientos y funciones de ataque.
6	Diseño del primer tesoro, caja contenedora, elementos brillantes.
7	Recompensa de 300 monedas por encontrar el primero de 5 tesoros.
8	Mundo 2: otra perspectiva de la isla pasa al siguiente nivel después de encontrar el primer tesoro.

La lista puede contener muchos ítems de acuerdo con la magnitud del proyecto a desarrollar y, para una elaboración detallada se debe consultar el documento de diseño del juego, ya que es allí donde se encuentra la lógica de la historia y todos los detalles de los elementos y mecánicas que la componen.

También se puede diseñar el formato del Sprint Backlog con el cual el equipo de desarrollo determinará las tareas a ejecutar en cada Sprint, como se observa en la siguiente tabla:

Tabla 2

Sprint Backlog

Objetivo Sprint #1	Realizar las funciones del mundo 1			Fecha inicio del Sprint	dd/mm/aaaa	
				Fecha final del Sprint	dd/mm/aaaa	
				N° de días del Sprint	30	
				N° horas de trabajo x día	8	
Ítem del Product Backlog		Prioridad en el Product Backlog	Horas estimadas totales	Horas de trabajo /día	Responsable	Estado (sin iniciar, progreso, terminado)
Interfaz principal		1				

rea	Ta	Estru ctura de contenedore s		3	3	XXXXXXXXXXX	T
rea	Ta	Selecc ión de activos		2	2	H HHHHHHHHH	T
rea	Ta	Integr ación de activos		5	3	VVVVVVVV	P
rea	Ta	Confi gurar nivel de componente s		4	0	GGGGGGGGG	S
Personaje principal			2				
rea	Ta	Selecc ionar personaje		3	2	KKKKKKKKKK	P
rea	Ta	Crear 6 animaciones		3 0	8	LLLLLLLLL	P

El Sprint Backlog está compuesto por tareas que son medidas en horas, teniendo en cuenta las disciplinas de las personas comprometidas en cada tarea y el número de personas con las que el equipo cuenta para llevar a cabo el Sprint.

Por ejemplo, si se deben crear 6 animaciones para el juego y solo se cuenta con una persona que se demora 8 horas por día, quiere decir que tardará más de 4 semanas para su elaboración, sobrepasando la duración del Sprint; lo que quiere decir que esta tarea se debe dividir en subtareas que harán parte de otros Sprint o en su defecto, contactar otro especialista para repartir el trabajo.

Se pueden utilizar herramientas digitales gratuitas como Jira Software para la elaboración de los Product Backlog y los Sprint Backlog. Las plantillas son proporcionadas por el programa:

En la siguiente figura se observa el Product Backlog del juego “La isla de los tesoros escondidos”.

Figura 2

Plantilla Jira Software

Proyectos / Realizar las funciones del mundo 1

Backlog

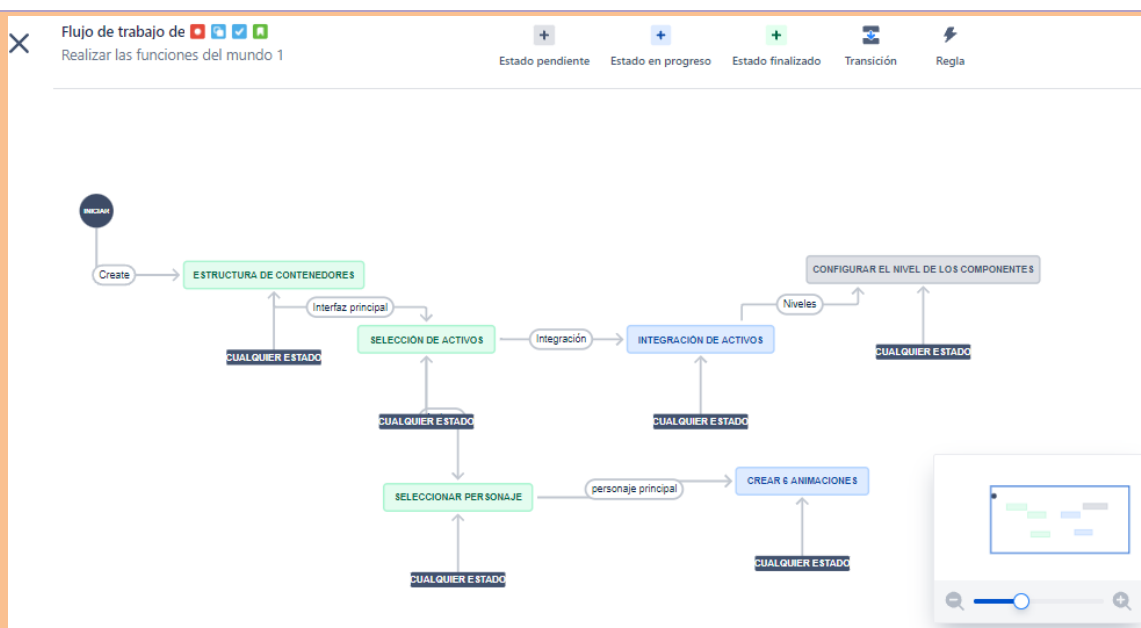
Q	OC	+	Epic	Insights
RLFDM1-1	Interfaz principal donde se visualice la isla y los activos que la componen, colocación de los tesoros en el nivel más bajo para que no...	TAREAS POR HACER		
RLFDM1-2	Diseño del personaje principal que buscara los tesoros escondidos. Crear animaciones con acciones de correr, saltar, mirar mapa, caer, camin...	EN CURSO		
RLFDM1-3	Mapa de navegación que es consultado por el personaje para ubicarse en la isla.	EN CURSO		
RLFDM1-4	Efecto del día y de la noche	EN CURSO		
RLFDM1-5	Diseño de animales feroces que atacan al personaje principal, movimientos y funciones de ataque.	EN CURSO		
RLFDM1-6	Diseño del primer tesoro, caja contenedora, elementos brillantes.	EN CURSO		
RLFDM1-7	Recompensa de 300 monedas por encontrar el primero de 5 tesoros.	EN CURSO		
RLFDM1-8	Mundo 2: otra perspectiva de la isla pasa al siguiente nivel después de encontrar el primer tesoro.	EN CL		Quickstart

Nota. Backlog del juego La isla de los tesoros escondidos.

Las tareas que se han definido en el Sprint Backlog el programa lo llama Flujo de trabajo, cuyo objetivo es realizar las funciones del mundo 1. Los estados de las tareas se indican por colores en la convención que se encuentra en la parte superior de la figura:

Figura 3

Flujo de trabajo



Nota. Estado de las tareas del Sprint.


Para profundizar en el tema consultar el material complementario especialmente los artículos Artefactos Scrum: las 3 herramientas clave de gestión y El backlog de producto.

2. Prototipado digital con paquetes prefabricados

Los prefabricados o prefabs como se les conoce en los prototipos de los videojuegos son un tipo de asset que guardan copias de objetos a los que se les ha colocado características y propiedades que pueden ser reutilizadas en el videojuego cuantas veces sea necesario. Los prefabs actúan como plantillas que se pueden utilizar en cualquier momento para no repetir las características de un objeto.

Se debe recordar que los asset son cualquier tipo de recurso que se utilice en el videojuego como, por ejemplo, luces, sistema de partículas, imágenes, texturas, materiales, sonidos, objetos dentro del escenario como rocas, pasto, entre otros.

Paquetes prefabs

<p>(Utilizar imagen relacionada a: free y signo pesos)</p>	<p>(utilizar imagen con diferentes iconos relacionados a categorías como automóviles, personajes, plantas, etc.)</p>	<p>(Utilizar imagen similar de diferentes personajes de videojuegos)</p> 
<p>Cuando se habla de paquetes prefabricados se hace referencia a que no es necesario crear los prefabs, sino que ya el sector de la industria de videojuegos ha desarrollado una serie de objetos que pueden ser utilizados de manera libre o paga, si lo que se busca no está dentro los que se</p>	<p>Estos paquetes están agrupados por categorías de acuerdo con lo que puede requerir un juego para armar las escenas de la historia; por ejemplo, sistema de partículas, medio ambiente, vehículos, efectos especiales,</p>	<p>Si se está utilizando el motor de Unity para diseñar y desarrollar un juego, desde las opciones del menú el programa permite la creación de prefabs o la importación de paquetes prefabricados estándar e igualmente gratuitos o comprados de</p>

pueden descargar de manera gratuita.	terrenos, edificaciones, entre otros.	los que ofrece la tienda Unity Asset Store.
--------------------------------------	---------------------------------------	---

Crear un prefabs desde las opciones del menú de Unity

Para este prefabs se va a suponer que la escena necesita varios cubos iguales colocados en el mismo espacio, para lo cual se debe crear un prefabs del cubo de acuerdo con los siguientes pasos:

Slider B

Anexo: CF010_2_Crear prefabs

Importar un paquete prefabricado estándar

El motor de Unity tiene incorporados asset que actúan como recursos prefabricados llamados activos estándar, los cuales facilitan la creación de una escena, proporcionando elementos tales como medio ambiente, personajes y sistemas de partículas entre otros:

Slider B

Anexo: CF010_2_Importar paquetes prefabs

Importar paquete de Asset Store

Cuando en los paquetes estándar no están los recursos necesarios para el videojuego se pueden importar de la tienda de Unity, esta acción se puede realizar estando en el motor de Unity:

Slider B

Anexo: CF010_2_Importar de Asset Store

Para profundizar en el tema consultar en el material complementario los siguientes artículos:

Instanciar un objeto en Unity 3D y Prefabs

3. Métricas del videojuego a partir de impresiones de detalles de eventos (debug.log)

Es un conjunto de herramientas de análisis que proporcionan información acerca de la experiencia que tienen los usuarios con el uso del producto y de esta forma medir su calidad. Esta información puede ser utilizada para realizar mejoras en todos los aspectos del juego.

A grandes rasgos la información que se puede extraer de un videojuego es:

Datos de los usuarios del juego

Se refiere a los datos de las personas que ingresan al juego, como por ejemplo, nombre, edad, género, lugar de nacimiento, entre otros. Esta información permite segmentar a los jugadores de acuerdo con su perfil para tomar decisiones respecto al desarrollo de funciones que despierten su interés.

Datos de hardware y software utilizados

Se refiere a los datos que se puedan obtener de los dispositivos, sistemas operativos e infraestructura que posean los usuarios para ejecutar el juego, puesto que de esta parte técnica depende también su experiencia de uso. No se puede lanzar un producto de videojuego en un lugar donde no funcionan adecuadamente las comunicaciones.

Datos de jugabilidad

Hace referencia a la experiencia del usuario en las diferentes sesiones del juego, se analiza la comprensión de las mecánicas a través de las diferentes sesiones jugadas, el recorrido por los niveles, cuántos logra superar, el tiempo que demora en avanzar y el tiempo total que invierte en el juego por día, semana y mes.

3.1. Métricas de Unity

El motor de videojuegos Unity utiliza el sistema Analytics, servicio que se configura automáticamente al ser instalado el programa y que contiene las siguientes métricas:

Métricas del jugador

Proporciona información sobre el número de jugadores que inician una sesión durante las 24 horas del día; con lo cual se pueden obtener las estadísticas del porcentaje mensual de participantes activos tanto nuevos como los que regresan.

Métricas de sesión

Se refiere al número total de sesiones que juegan en promedio los usuarios por día y el tiempo de juego promedio de estos.

Métrica de retención

Proporciona el porcentaje de jugadores que regresan al juego al siguiente día de jugar por primera vez, los que regresaron a los siete días y los que regresaron a los 30 días.

Métrica de monetización

Hace referencia a los ingresos promedio por usuario que juegan en un día, los cuales pueden ser obtenidos de los anuncios. Para esto Unity hace la verificación mediante la función Analytics Transaction.

Métrica de Ads

O métrica de anuncios, hace referencia a los ingresos promedios por anuncios que ven los jugadores durante el juego, al igual que los ingresos totales. Los anuncios de video que se reproducen en el juego y los ingresos estimados por 1000 impresiones de anuncios.

Generador de segmentos

Dentro de la base de datos de los jugadores, Unity forma grupos diferenciados aspectos como:

Segmentos del ciclo de vida: número de días que jugaron desde que ingresaron por primera vez: (1-3) días; (4-7) días; (8-14) días; (15-30) días; (31 -90) días.

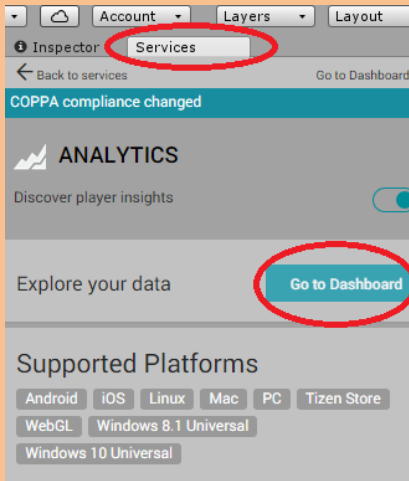
Segmentos geográficos: de acuerdo con la dirección IP desde donde se conecta el jugador Unity lo geolocaliza. El desarrollador puede definir las zonas geográficas o países que va a incluir.

Segmentos de monetización: agrupa los jugadores de acuerdo con la cantidad de dinero que gastan en el juego.

Segmentos por plataforma: clasifica a los jugadores de acuerdo con dos plataformas, los que utilizan Android o los que utilizan IOS.

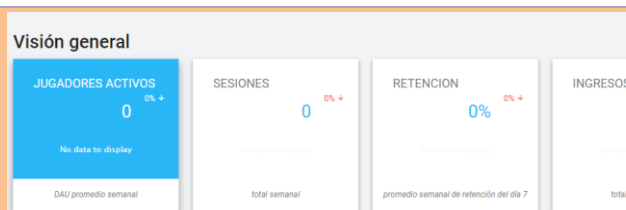
Segmentos personalizados: es una categoría de jugadores que se crea combinando reglas de los segmentos anteriores.

Para hacer uso de las métricas de Unity el juego debe estar por lo menos en la versión Beta, en la cual se hacen pruebas de usuario para medir su experiencia. Los pasos a seguir son:

<p>1. Los desarrolladores deben abrir el proyecto del videojuego y en la ventana del inspector hacer clic en la pestaña Servicio:</p>	 <p>Servicio de análisis del videojuego</p>
<p>2. Una vez cargada la pestaña de servicios de Unity se hace clic en el botón ir al tablero: Go to dashboard</p>	

para acceder a las diferentes métricas de análisis.

Las métricas que se observan en la figura, aparecen en (0), debido a que el juego se está desarrollando y todavía no se han convocado usuarios para comprobar las funciones 1.



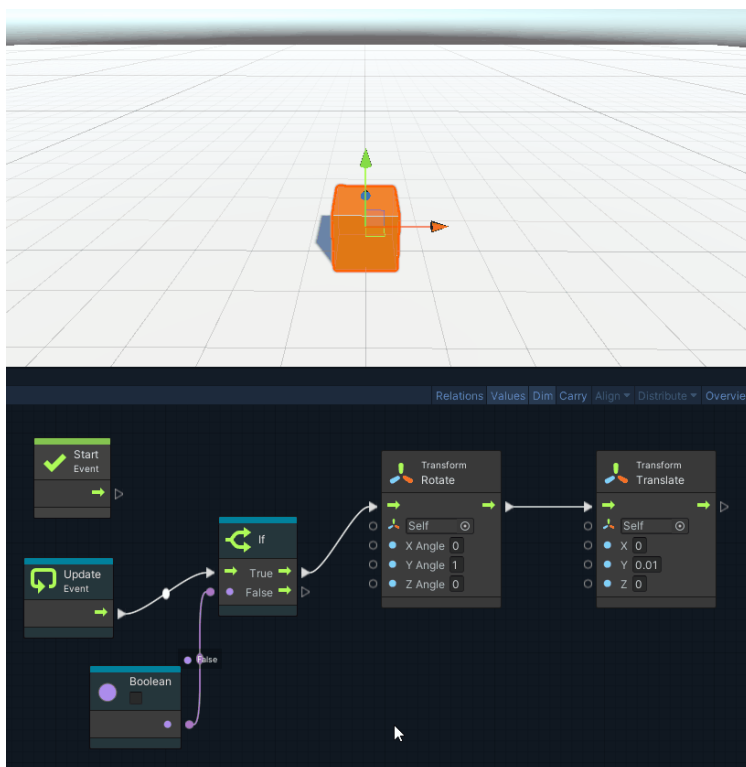
Métricas de un juego en fase de diseño

3.2. Método debug.log

En Unity el método debug.log consiste en una función que se utiliza para imprimir mensajes en la ventana de la consola con el objeto de visualizar los eventos del juego, en cuanto al comportamiento de los objetos y componentes que se han colocado en las escenas, inspeccionando el flujo de las acciones con el propósito de depurarlas, por ejemplo, se va a inspeccionar el evento A: rotación del cubo en la escena del juego de acuerdo con las coordenadas iniciales (X=0, Y=1, Z=0) a las coordenadas finales (X = 0, Y = 0.01, Z = 0), como se muestra a continuación:

Figura 4

Evento A



Nota. Flujo del evento A. Hektor docs, (2021). Tipos de datos.

<https://docs.hektorprofe.net/unity/visual-scripting/tipos-datos/>

Función debug.log

El flujo de transformación se puede imprimir en la ventana de la consola de Unity a través de la función `debug.log`, como se puede detallar en la siguiente imagen interactiva, donde se observan términos que corresponden a las líneas de código relacionadas con el evento.

IMAGEN INTERACTIVA

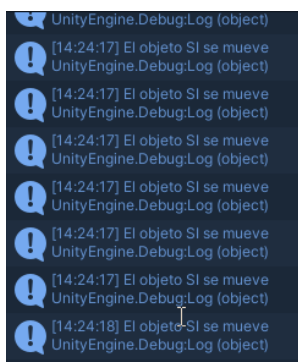
Anexo: CF010_3_ Metodo Debug.Log

Nota. adaptado de la Utilización de la función debug.log para inspeccionar el evento A. Hektor docs, (2021). Tipos de datos. <https://docs.hektorprofe.net/unity/visual-scripting/tipos-datos/>

Obsérvese en la figura que el mensaje en la ventana de la consola indica que el cubo efectivamente ha rotado de una coordenada inicial a una final: “el objeto SI se mueve”:

Figura 5

Mensajes consola



Nota. Impresión de mensajes del evento A en la ventana de la consola. Hektor docs, (2021). Tipos de datos. <https://docs.hektorprofe.net/unity/visual-scripting/tipos-datos/>

Crear Script

Quando se necesita manipular el objeto e imprimir el flujo del evento con mensajes propios se debe crear el Script mediante los siguientes pasos:

PASOS B

Anexo: CF010_3_Crear script

Para profundizar en el tema consultar en Unity Documentation: “Métricas, segmentos y terminología de Analytics” y “Depurar” que se encuentran en el material complementario.

4. Verificación del videojuego

Para que la comprobación y los ajustes de un videojuego se realicen rápidamente es importante hacer uso de un método ágil, que permita un trabajo fluido y un incremento del proyecto a través de cada entregable aprobado por el cliente.

4.1. Verificación basada en el modelo SCRUM

Al utilizar un método como SCRUM se deben planear no solo el diseño y desarrollo sino también la verificación del producto, llevando a cabo las diferentes fases del método:

INFOGRAFÍA

Anexo: CF010_4.1_Modelo Scrum

Para ampliar los conocimientos relacionados con este tema consultar el artículo “Metodología Scrum” que se encuentra en el material complementario.

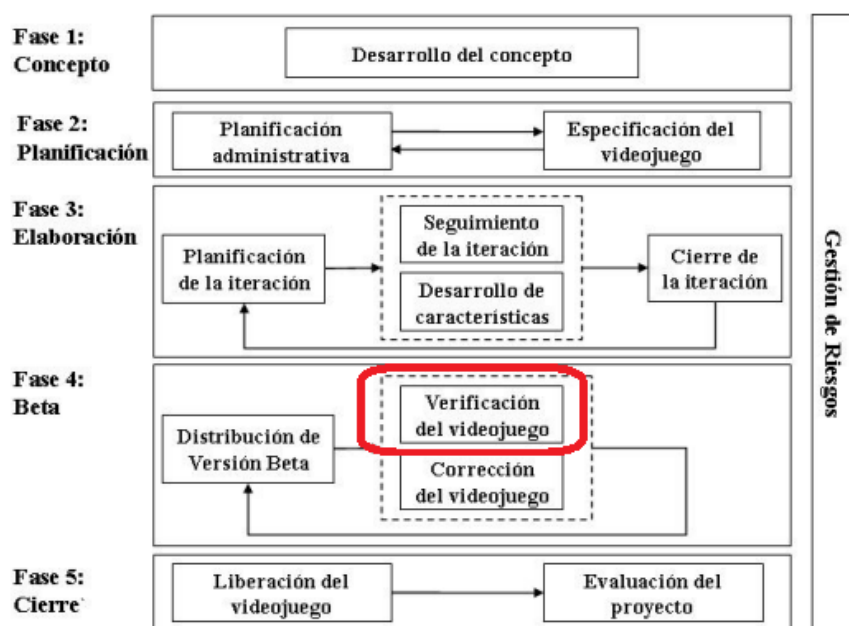
4.2. Verificación basada en el modelo SUM

Bajo la metodología SUM el videojuego se desarrolla en cinco fases donde en cada una se debe lograr un objetivo específico, realizando un proceso de mejora continua para obtener resultados administrando eficientemente los recursos y los riesgos.

La etapa Beta es la primera versión completa del videojuego y en la siguiente figura se observa que en la “Fase 4: Beta” se encuentra la “Verificación del videojuego”, puesto que ya las funcionalidades se pueden probar:

Figura 6

Metodología SUM



Nota. Fases de la metodología SUM. Adaptado de Acerenza. Et al. (2009). Una metodología para desarrollo de videojuegos.

https://www.fing.edu.uy/sites/default/files/biblio/22811/asse_2009_16.pdf

La fase 4 está constituida por los siguientes elementos:

Verificadores Beta

Son los designados para hacer la verificación de las funciones del videojuego, pueden tener experiencia en probar el software de videojuegos, o no poseerla, haber jugado o no.

Cliente

Está en casi todas las etapas del diseño y desarrollo del videojuego, puesto que se encarga de definir el concepto del juego, especificar y priorizar las características y las tareas que le dan más valor, evalúa el producto obtenido al finalizar cada iteración, prioriza la corrección de los errores y valida las versiones del producto.

Equipo de desarrollo

Está conformado por diseñadores estructurales, diseñadores gráficos y programadores. Aportan a la definición del concepto, realizan la estimación del tiempo, definen las tareas, las asignan y realizan para producir el videojuego. Evalúan el proceso y colocan los correctivos a los problemas.

Aspectos por verificar

Son las características funcionales y no funcionales que deben comprobar los verificadores Beta.

Videojuego

Es la versión ejecutable del contenido que es reproducida para la verificación de los aspectos funcionales y no funcionales.

Evaluación y errores encontrados

Es la comprobación del funcionamiento del juego, en la cual se genera una lista de los aspectos a corregir, planificándose una iteración para priorizar las correcciones a los errores encontrados.

Lista de cambios priorizados

Es la lista de errores encontrados en el funcionamiento del juego y generada en la evaluación. Los errores se deben ordenar de acuerdo con el nivel de importancia y afectación en las funcionalidades del juego.

Realizar ajustes

Es la corrección de los errores que se encuentran en la lista de cambios priorizados, teniendo en cuenta la prioridad y los costos del ajuste a realizar.

Para los conocimientos en el tema consultar el documento SUM para el desarrollo de videojuegos que se encuentra en el material complementario.

4.3. Verificación y ajustes en el entorno de desarrollo (consola, modo edición)

La verificación de un videojuego desarrollado en el motor Unity consiste en la comprobación de la calidad y el funcionamiento de sus mecánicas, así como de los elementos y los recursos utilizados en él. Para ello, el programa tiene incorporado un paquete Package Manager que puede utilizar el desarrollador para realizar las pruebas que necesita.

ANIMACIÓN 2D

Anexo: CF010_4.3_Verificacion y ajustes

Los desarrolladores pueden crear pruebas para testear cualquier cosa en los objetos que han colocado en una escena del juego. En el siguiente ejemplo se observa una escena que contiene una esfera de color verde encima de un cubo de color blanco que hace de piso; se quiere verificar que la esfera caiga bajo el efecto de la gravedad sobre el piso que es el cubo, mediante los siguientes pasos:

PASOS B

Anexo: CF010_4.3_Ejemplo Verificacion

Para profundizar en el tema con algunos consejos sobre cómo usar Unity Test Framework, consultar el tema “Prueba el código de tu juego con Unity Test Framework” que se encuentra dentro del material complementario.

Síntesis

A continuación, la síntesis del componente formativo abordado:



Material complementario

Tema	Referencia APA del Material	Tipo de material	Enlace del Recurso o Archivo del documento material
Entidades de información digital.	Ecosistema de Recursos Educativos Digitales SENA. (2022). <i>La tecnología en la información digital</i> . [Video]. YouTube. https://www.youtube.com/watch?v=puMO4kf6PRI	Video	https://www.youtube.com/watch?v=puMO4kf6PRI
Metadatos.	Micro aprendizaje: internet. (2019). <i>Qué son los metadatos</i> . [Video]. YouTube. https://www.youtube.com/watch?v=W9df022FHIE	Video	https://www.youtube.com/watch?v=W9df022FHIE
Registro de información.	Ecosistema de Recursos Educativos Digitales SENA. (2021). <i>Administración y normativa de registros de información: introducción</i> . [Video]. YouTube. https://www.youtube.com/watch?v=ytQ19oNM_Ro	Video	https://www.youtube.com/watch?v=ytQ19oNM_Ro

Glosario

Búsqueda: conjunto de procedimientos y operaciones que un usuario realiza para obtener la información que necesita.

Búsqueda automatizada interactiva: técnica de búsqueda consistente en explotar un ordenador de manera que se desarrolle una serie de preguntas y respuestas alternas entre un usuario y el ordenador de manera parecida a un diálogo entre dos personas.

Gestión de la Información (GI): denominación convencional de un conjunto de procesos por los cuales se controla el ciclo de vida de la información, desde su obtención (por creación o captura), hasta su disposición final (su archivo o eliminación).

Herramienta digital: paquetes informáticos que están en las computadoras, o en dispositivos electrónicos como celulares y tabletas, entre otros.

Hardware: conjunto de elementos físicos o materiales que constituyen una computadora o un sistema informático

Información: conjunto de datos, ya procesados y ordenados para su comprensión, que aportan nuevos conocimientos a un individuo o sistema sobre un asunto, materia, fenómeno o ente determinado.

Metadatos para la gestión de documentos: información estructurada o semiestructurada que permite la creación, gestión y uso de los documentos a lo largo del tiempo y entre diferentes dominios.

Seguridad de la información: conjunto de medidas preventivas y reactivas de las organizaciones y sistemas tecnológicos que permiten resguardar y proteger la información buscando mantener la confidencialidad, la disponibilidad e integridad de datos.

Software: conjunto de programas y rutinas que permiten a la computadora realizar determinadas tareas.

Referencias bibliográficas

Archivo General de la Nación (2022). Guía para la formulación de un esquema de metadatos para la gestión de documentos.

https://www.archivogeneral.gov.co/sites/default/files/Estructura_Web/5_Conulte/Recursos/Publicaciones/GuiaDeMetadatos.pdf

Instituto Colombiano de Normas Técnicas y Certificación. (2022). Normas técnicas de la información y seguridad de la familia ISO 27000. Icontec. <https://www.icontec.org/servicio-educacion/modulares-de-educacion/>

Ministerio de Educación. (2022). Guía para la clasificación de la información. https://www.mineducacion.gov.co/1759/articles-407695_galeria_14.pdf

Ministerio de Tecnologías de la Información y Comunicaciones - MinTIC. (2022). Entidades del sector. MinTIC. <https://www.mintic.gov.co/portal/inicio/Ministerio/Entidades-del-sector/>

Ministerio de Tecnologías de la Información y Comunicaciones - MinTIC. (2022). Interoperabilidad. MinTIC. <https://www.mintic.gov.co/arquitecturati/630/w3-propertyvalue-8117.html>

Créditos

Nombre	Cargo	Regional y Centro de Formación
María Camila García Santamaria	Líder Ecosistema de Recursos Educativos Digitales	Dirección General
Rafael Neftalí Lizcano Reyes	Responsable Equipo Contenido Instruccional	Regional Santander – Centro Industrial del Diseño y la Manufactura
	Experto Temático	Sena Distrito Capital Grupo De Apoyo Administrativo - CGI
	Diseñador Instruccional	Regional Norte de Santander – Centro de la Industrial, la Empresa y los Servicios
	Asesora Metodológica	Regional Distrito Capital - Centro de Diseño y Metrología.
	Corrector de estilo	Regional Distrito Capital- Centro de Diseño y Metrología.
Francisco José Lizcano Reyes	Responsable Equipo Diseño y Desarrollo	Regional Santander – Centro Industrial del Diseño y la Manufactura
	Diseño PDF Descargable	Regional Santander – Centro Industrial del Diseño y la Manufactura
	Validación de Contenido Componente Descargable	Regional Santander – Centro Industrial del Diseño y la Manufactura
Tatiana Villamil	Responsable Repositorio	Regional Tolima – Centro de Comercio y Servicios