

**SÍLABO
DESARROLLO DE VIDEO JUEGOS****ÁREA CURRICULAR: CIENCIAS DE LA COMPUTACIÓN****I. DATOS GENERALES**

1.1	Departamento Académico	: Ingeniería y Arquitectura
1.2	Semestre Académico	: 2019-I
1.3	Código de la asignatura	: 091126E4040
1.4	Ciclo	: Electivo
1.5	Créditos	: 4
1.6	Horas semanales totales	: 6
	1.6.1 Horas lectivas (Teoría, Práctica, Laboratorio)	: 4 (T=4, P=0, L=0)
	1.6.2 Horas no lectivas	: 2
1.7	Condición de la asignatura	: Electivo
1.8	Requisito(s)	: 09013707050 Ingeniería de Software II
1.9	Docentes	: Ing. Fred Duarte

II. SUMILLA

El curso es de naturaleza especializada, dirigiéndose a un público objetivo con gusto por los videojuegos y la curiosidad para crearlos, así como la teoría que involucra la temática del mismo. Los alumnos aprenderán la parte teórica que involucra los conceptos generales de juegos y de desarrollo siendo estos desarrollados en un entorno bidimensional y tridimensional.

En el curso se realiza un proyecto en grupo, el mismo que exige a los estudiantes cubrir todas las fases de desarrollo hasta la implementación y evaluación de un primer prototipo.

Unidades: 1. Conceptos generales de juegos, desarrollo e introducción al entorno de desarrollo. 2. Lógica y desarrollo de juegos retro sin enemigos. 3. Lógica y desarrollo de juegos con inteligencia artificial de alto impacto empresarial.

III. COMPETENCIAS Y SUS COMPONENTES COMPRENDIDOS EN LA ASIGNATURA**3.1 Competencia**

- Aplica conocimientos de computación y matemáticas apropiadas para los resultados del estudiante y las disciplinas enseñadas.
- Analiza un problema e identifica y define los requerimientos apropiados para su solución.
- Diseña, implementa y evalúa un sistema basado en computadoras, procesos, componentes o programa que satisfagan las necesidades requeridas.
- Usa técnicas, destrezas, y herramientas modernas necesarias para la práctica de la computación.

3.2 Componentes**Capacidades**

- Describe y explica los conceptos básicos relacionados con el diseño lógico de algoritmos y video juegos.
- Aplica los conceptos básicos sobre el diseño lógico de un video juego.
- Identifica y reconoce las estructuras lógicas para el diseño y desarrollo de video juegos.
- Aplica las estructuras lógicas las estructuras lógicas para el diseño y desarrollo de video juegos.

Contenidos actitudinales

- Aprende a trabajar en equipo.
- Aprende de sus propios errores a partir de su propia experiencia
- Entiende que conocimientos debe lograr para aprender los contenidos de manera más eficiente
- Es responsable y cumple con las actividades asignadas por el docente

IV. PROGRAMACIÓN DE CONTENIDOS

UNIDAD I: CONCEPTOS GENERALES DE JUEGOS, DESARROLLO E INTRODUCCION AL ENTORNO DE DESARROLLOS					
CAPACIDAD: <ul style="list-style-type: none"> Identificar los principales paneles de la plataforma de desarrollo. Conocer la nomenclatura de los objetos en la plataforma de desarrollo. Identificar el sistema de coordenadas en la plataforma de desarrollo. Posicionamiento y escala de objetos simples. 					
SEMANA	CONTENIDOS CONCEPTUALES	CONTENIDOS PROCEDIMENTALES	ACTIVIDAD DE APRENDIZAJE	HORAS	
				L	T.I.
1	Primera sesión ¿Qué es un juego? ¿Por qué jugamos? ¿Qué es un videojuego? Videojuegos de ayer, hoy y proyecciones futuras. Tipos de videojuegos. Segunda sesión Tipo de jugadores en los videojuegos: los 4 cuadrantes de Richard Bartle. Conceptos que involucran la atracción de un usuario a los juegos: Onboarding, Scaffolding, Paths to mastery. Formas de motivar a los jugadores a permanecer en los juegos ya desarrollados. Laboratorio Manejo del IDE a emplear con el lenguaje de programación correspondiente.	- Explica los conceptos principales de los video juegos. - Explica e identifica los conceptos básicos sobre los game object - Utiliza activamente el IDE (Entorno de Desarrollo) a emplear en el curso	Lectivas (L): - Desarrollo del tema – 2h - Ejercicios en Laboratorio - 2h De trabajo Independiente (T.I): - Desarrollo de ejercicios - 2 h	4	2
2	Primera sesión Roles y etapas en el desarrollo de videojuegos. El arte de ser un Game Designer de juegos y los conceptos alrededor del mismo. Introducción a la plataforma de desarrollo, conociendo los paneles: Escena, Juego, Jerarquía, Proyecto e Inspector. Manejo del panel escena para cambios de vista 2D – 3D. Creando mi primer GameObject. Segunda sesión Creando mi primer proyecto en la plataforma de desarrollo. Modificando GameObjects utilizando la vista Inspector, añadiendo componentes manualmente a los GameObject.	- Explica el uso del IDE y como interactuar con los objetos del videojuego - Inserta el uso de Sprites en el IDE - Explica conceptos de Colisiones y Objetos Rígidos en el IDE	Lectivas (L): - Desarrollo del tema – 2h - Ejercicios en laboratorio – 2h De trabajo Independiente (T.I): - Desarrollo de ejercicios - 2 h	4	2

UNIDAD II : LÓGICA Y DESARROLLO DE JUEGOS RETRO SIN ENEMIGOS

CAPACIDAD:

- Entender las mecánicas y reglas que involucran a un juego.
- Abstracta las mecánicas de juegos retro sin enemigos en base a juego continuo.
- Conocer la plataforma de desarrollo en el entorno de juegos.
- Plasmar las mecánicas extraídas en lógica de programación.
- Aprender los scripts necesarios para plasmar las mecánicas extraídas.

SEMANA	CONTENIDOS CONCEPTUALES	CONTENIDOS PROCEDIMENTALES	ACTIVIDAD DE APRENDIZAJE	HORAS	
				L	T.I.
3	Primera sesión Conociendo la plataforma de desarrollo. Eventos en los juegos start y update, características. Conociendo el script GameObject y modificando nombre de objetos con código. Primera práctica calificada Segunda sesión Jugando Arkanoid a fin de entender las reglas que involucran a los juegos. ¿Qué es una mecánica? Extracción de mecánicas del juego	<ul style="list-style-type: none"> • Manejo de Eventos con la herramienta uso de Visual Studio para agregar código • Sincronizar Unity con el Código del Script 	Lectivas (L): - Desarrollo del tema – 2h - Ejercicios en laboratorio – 2h De trabajo Independiente (T.I.): - Desarrollo de ejercicios - 1 h	4	1
4	Primera sesión Configuración de la plataforma de desarrollo para iniciar a desarrollar Arkanoid. ¿Qué es un sprite? Creación de sprites necesarios para iniciar la programación. Configuración de los Sprites. Segunda sesión Programando al Jugador, leyendo datos desde el teclado para mover al jugador.	<ul style="list-style-type: none"> • Creación de Sprites con imágenes descargadas de internet para poder hacer una animación • Agregar control en las secuencias con el teclado y hacer que el videojuego interprete los comandos del script 	Lectivas (L): - Desarrollo del tema – 2h - Ejercicios en laboratorio – 2h De trabajo Independiente (T.I.): Desarrollo de ejercicios - 2h	4	2
5	Primera sesión Creación de escenario estático para el juego, limitación de movimiento de jugador a fin de no salir del escenario. Lógica del jugador., Segunda sesión Creando colisionadores para el jugador. Creando un objeto (pelota) para interactuar con el jugador, programación de mecánicas y reglas de colisionamiento con el jugador Control de lectura 1	<ul style="list-style-type: none"> • Delimitación del escenario en el IDE • Asociación de las colisiones en unity 	Lectivas (L): - Desarrollo del tema – 2h - Ejercicios en laboratorio – 2h De trabajo Independiente (T.I.): Desarrollo de ejercicios - 2 h	4	2

6	<p>Primera sesión Lógica de la pelota. Evento de programación Awake(). Generación de números aleatorios para evitar lo previsible de un juego.</p> <p>Segunda sesión Creando parentescos en la plataforma de desarrollo. (Parent – Child), uso de scripts para crear y desvincular parentescos en la plataforma. Uso de Tags para la búsqueda de gameObjects.</p>	<ul style="list-style-type: none"> • Codificación de los eventos en los scripts • Uso de aleatorios en el script 	<p>Lectivas (L):</p> <ul style="list-style-type: none"> - Desarrollo del tema – 2h - Ejercicios en laboratorio – 2h 	4	2
7	<p>Primera sesión Creando otros elementos y lógica de choque entre ellos y la pelota. Sistema de puntuaciones. Destruyendo elementos de la escena del juego con scripts.</p> <p>Segunda sesión Cambios de escenario, eventos mediante interfaces de entrada (mouse), generación de Build del aplicativo.</p> <p>Segunda Práctica Calificada: Primer avance del proyecto</p>	<ul style="list-style-type: none"> • Codificación de Destroy en el código • Uso de la generación de video juego para instalarse en un dispositivo. 	<p>Lectivas (L):</p> <ul style="list-style-type: none"> - Desarrollo del tema – 2h - Ejercicios en laboratorio – 2h <p>De trabajo Independiente (T.I): Desarrollo de ejercicios - 1 h</p>	4	1
8	Examen Parcial				

UNIDAD III : LÓGICA Y DESARROLLO DE JUEGOS CON INTELIGENCIA ARTIFICIAL DE ALTO IMPACTO EMPRESARIAL

CAPACIDAD:

- Dar a conocer al estudiante la fama de los videojuegos exitosos y cuál es el valor agregado que tienen.
- Abstraer las mecánicas de juegos exitosos y plasmarlas en el entorno de desarrollo.
- Crear varios enemigos con inteligencia artificial a fin de dar mayor versatilidad al juego.
- Crear un juego con escenario inmóvil y múltiples elementos a fin de que el alumno entienda las mecánicas involucradas

SEMANA	CONTENIDOS CONCEPTUALES	CONTENIDOS PROCEDIMENTALES	ACTIVIDAD DE APRENDIZAJE	HORAS	
				L	T.I.
9	Primera sesión Juegos exitosos de ayer, hoy y proyecciones futuras, montos facturados aproximados, razón de su éxito. Segunda sesión Identificación de las mecánicas en los juegos exitosos, uso de mecánicas similares en otros juegos exitosos, valores agregados	<ul style="list-style-type: none"> • Codificación de los eventos en los scripts • Manipulación de los atributos en unity para las acciones • Pruebas de la implementación con la herramienta unity. 	Lectivas (L): - Desarrollo del tema – 2h - Ejercicios en laboratorio – 2h De trabajo Independiente (T.I): Desarrollo de ejercicios - 1 h	4	1
10	Primera sesión Plantas vs Zombies. Abstracción de primeras mecánicas. División de elementos a fin de dar inicio a la programación Segunda sesión Unidades elementales del juego plantas vs zombies. Establecimiento de unidades en una matriz gráfica 5x9. Control de lectura 2	<ul style="list-style-type: none"> • Codificación de los eventos en los scripts • Manipulación de los atributos en unity para las acciones • Pruebas de la implementación con la herramienta unity. 	Lectivas (L): - Desarrollo del tema – 2h - Ejercicios en laboratorio –2h De trabajo Independiente (T.I): Desarrollo de ejercicios - 1 h	4	1
11	Primera sesión Utilizando sprites liberados del juego original, Introducción a las animaciones. Animando elementos del juego manualmente. Creación del girasol. Segunda sesión Creando colisionadores para el girasol. Lógica del girasol. Introducción a las corrutinas.	<ul style="list-style-type: none"> • Codificación de los eventos en los scripts • Manipulación de los atributos en unity para las acciones • Pruebas de la implementación con la herramienta unity. 	Lectivas (L): - Desarrollo del tema – 2h - Ejercicios en laboratorio –2h De trabajo Independiente (T.I): Desarrollo de ejercicios - 1 h	4	1
12	Primera sesión Creando el Sol. Lógica del sol y efectos de luz y colores. Manejo de colores y transparencias mediante código de programación. Segunda sesión Creación de Peashooter. Mecánica involucrada, animación de peashooter y estados.	<ul style="list-style-type: none"> • Codificación de los eventos en los scripts • Manipulación de los atributos en unity para las acciones • Pruebas de la implementación con la herramienta unity. 	Lectivas (L): - Desarrollo del tema – 2h - Ejercicios en laboratorio –2h De trabajo Independiente (T.I): Desarrollo de ejercicios - 1 h	4	1

SEMANA	CONTENIDOS CONCEPTUALES	CONTENIDOS PROCEDIMENTALES	ACTIVIDAD DE APRENDIZAJE	HORAS	
				L	T.I.
13	Primera sesión Introduccion a Raycast, lógica de Peashooter, Deteccion de peashooter con otros objetos Tercera Práctica Calificada. Segundo avance del proyecto. Segunda sesión Creación de enemigo. Animaciones de enemig y detección de peashooter al enemigo	<ul style="list-style-type: none">Codificación de los eventos en los scriptsManipulación de los atributos en unity para las accionesPruebas de la implementación con la herramienta unity.	Lectivas (L): - Desarrollo del tema – 2h - Ejercicios en laboratorio –2h De trabajo Independiente (T.I): Desarrollo de ejercicios - 1 h	4	1
14	Primera sesión Creando el proyectil para peashooter. Relación de Proyectil-Peashooter y Zombie. Mecánicas y scripts involucrados. Segunda sesión Programación de reglas de derrota del enemigo (zombie) de acuerdo a la cantidad de impactos y daño del peashooter. Control de lectura 3	<ul style="list-style-type: none">Codificación de los eventos en los scriptsManipulación de los atributos en unity para las accionesPruebas de la implementación con la herramienta unity.	Lectivas (L): - Desarrollo del tema – 2h - Ejercicios en laboratorio –2h De trabajo Independiente (T.I): Desarrollo de ejercicios - 1 h	4	1
15	Primera sesión Programacion de animaciones de acuerdo a estados de elementos de juego: Planta, Zombie. Creacion de HUD para colocación de plantas adicionales. Segunda sesión Reglas del tablero respecto a la colocación de las plantas, estado del zombie de acuerdo a derrota, Deadzone, colocación de mas plantas desde el HUD. Exposición de proyectos. Tercer avance del proyecto	<ul style="list-style-type: none">Codificación de los eventos en los scriptsManipulación de los atributos en unity para las accionesPruebas de la implementación con la herramienta unity.	Lectivas (L): - Desarrollo del tema – 2h - Ejercicios en laboratorio –2h De trabajo Independiente (T.I): Desarrollo de ejercicios - 1 h	4	1
16	Examen final.				
17	Entrega de promedios finales y acta de la asignatura				

V. ESTRATEGIAS METODOLÓGICAS

- **Método Expositivo** – Interactivo. Comprende la exposición del docente y la interacción con el estudiante.
- **Método de Discusión Guiada**. Conducción del grupo para abordar situaciones y llegar a conclusiones y recomendaciones.
- **Método de Demostración – Ejecución**. Se utiliza para ejecutar, demostrar, practicar y retroalimentar lo expuesto.

VI. RECURSOS DIDÁCTICOS

- **Equipos**: Computadora, écran y proyector multimedia.
- **Materiales**: Material docente, prácticas dirigidas de laboratorio y textos bases (ver fuentes de consultas).
- **Lenguaje de Programación**: C#
- **Software**: Unity3d , Git, Visual Studio Community

VII. EVALUACIÓN DEL APRENDIZAJE

El promedio final (PF) de la asignatura se obtiene con la siguiente fórmula:

$$PF = (2*PE+EP+EF) / 4$$

EP = Examen Parcial

EF = Examen Final

PE = Promedio de Evaluaciones

El promedio de evaluaciones (PE) se obtiene de la siguiente manera:

$$PE = (W1 + W2 + W3 + W4) / 4$$

W1, W2, W3, W4 = Trabajo individual y en equipo como avance de su proyecto

VIII. FUENTES DE CONSULTA.

8.1 Bibliográficas

- Universidad de Castilla la Mancha (2015). Programación gráfica de videojuegos.
- Kiaw, A. (2014). Unity 4.x Game AI Programming. Estados Unidos.
- Werbach Kevin - Gamification - Coursera. University of Pennsylvania (2013)
- Stagner, A. (2013). Unity Multiplayer Games. Estados Unidos.
- Jesse Schell - The Art of Game Design. (2008).
- Bernards Suits, The Grasshopper, Games Life and Utopia (2005).

IX. APOORTE DE LA ASIGNATURA AL LOGRO DE RESULTADOS

El aporte de la asignatura al logro de los Resultados del Estudiante (*Student Outcomes*) en la formación del graduado en Ingeniería de Computación y Sistemas, se establece en la tabla siguiente:

K = clave **R** = relacionado **Recuadro vacío** = no aplica

a.	Habilidad para aplicar conocimientos de computación y matemáticas apropiadas para los resultados del estudiante y las disciplinas enseñadas.	K
b.	Habilidad para analizar un problema e identificar y definir los requerimientos apropiados para su solución.	K
c.	Habilidad para diseñar, implementar y evaluar un sistema basado en computadoras, procesos, componentes o programa que satisfagan las necesidades requeridas.	K
d.	Habilidad para trabajar con efectividad en equipos para lograr una meta común.	
e.	Comprensión de los aspectos y las responsabilidades profesional, ética, legal, de seguridad y social.	R
f.	Habilidad para comunicarse con efectividad con un rango de audiencias.	
g.	Habilidad para analizar el impacto local y global de la computación en los individuos, organizaciones y la sociedad.	R
h.	Reconocer la necesidad y tener la habilidad para comprometerse a un continuo desarrollo profesional.	R
i.	Habilidad para usar técnicas, destrezas, y herramientas modernas necesarias para la práctica de la computación.	K
j.	Comprensión de los procesos que soportan la entrega y la administración de los sistemas de información dentro de un entorno específico de aplicación.	K