

PROYECTO 2

GENETIC KINGDOM

Instituto Tecnológico de Costa Rica
Escuela de Ingeniería en Computadores
Algoritmos y Estructuras de Datos II (CE 2103)
I Semestre 2025

OBJETIVOS

GENERAL

- Diseñar e implementar un juego de estilo “Tower defense” en que se apliquen algoritmos genéticos y pathfinding

ESPECÍFICOS

- Implementar algoritmos genéticos y pathfinding en el lenguaje de programación C++.
- Diseñar la solución de un problema mediante OOP aplicada a C++.
- Implementar un diseño OOP en el lenguaje de programación C++.

REQUERIMIENTOS

El proyecto consiste en implementar un juego estilo “Tower Defense” ambientado en la edad media en C++ para desktop. El juego genera oleadas de enemigos de distintas clases y categorías. El jugador se encarga de colocar torres en lugares predeterminados para evitar que los enemigos crucen el puente del castillo. Después de cada oleada, los enemigos evolucionan haciendo más difícil evitar que los enemigos crucen el puente.

ID	DESCRIPCIÓN	PUNTOS
001	El mapa es una cuadrícula de tamaño fijo definida por el grupo de trabajo. El usuario puede colocar torres en cualquiera de los cuadros del mapa asegurando que no bloqueen todos los posibles caminos hacia el puente. El mapa tiene un único punto de ingreso de los enemigos y está en el lado contrario al puente del castillo	10
002	Cada torre tiene los siguientes atributos: daño, velocidad, alcance, tiempo de regeneración del poder especial, tiempo de recarga de ataque. Las torres atacan a los enemigos cuando están en su alcance. Con cada muerte de enemigo, se considera su categoría y tipo para devolver cierta	10

	cantidad de oro al jugador. La cantidad de oro retornado debe ser calculado de forma justa y consistente. El grupo de trabajo define esto.	
003	Todas las torres tienen 3 <i>upgrades</i> . Cada <i>upgrade</i> aumenta el daño que pueden causar y cada torre tiene ataques especiales que ocurren cada cierto tiempo con una probabilidad definida por los estudiantes. Los <i>upgrades</i> tienen un costo en oro. Cada <i>upgrade</i> es más caro que la anterior. Cada <i>upgrade</i> modifica los atributos en un valor definido por el grupo de trabajo.	10
004	Hay 3 tipos de torres: <ul style="list-style-type: none"> • Arqueros: bajo costo, alto alcance, poco daño, tiempo de recarga de ataque bajo • Magos: costo medio, alcance medio, daño medio, tiempo de recarga de ataque medio • Artilleros: costo alto, alcance bajo, daño alto, tiempo de recarga de ataque alto 	5
005	El jugador puede ir colocando las torres en cada lugar disponible. Al seleccionar un lugar disponible, puede escoger el tipo de torre que desea crear. Cada torre tiene un costo en oro.	10
006	Los enemigos aparecen por oleadas. Cada oleada es una generación que evoluciona. Los enemigos pueden ser los siguientes: <ul style="list-style-type: none"> • Ogros: son el enemigo más básico. Son resistentes a los arqueros y débiles contra la magia y la artillería. Son lentos. • Elfos Oscuros: son resistentes a la magia, pero débiles a los arqueros y a la artillería. Son muy rápidos • Harpías: solo pueden ser atacadas por magia y arqueros. Tienen una velocidad intermedia • Mercenarios: son débiles a la magia, pero resistentes a arqueros y artillería. <p>Cada enemigo tiene atributos como:</p> <ul style="list-style-type: none"> • Vida • Velocidad • Resistencia a flechas • Resistencia a la magia • Resistencia a la artillería 	20

007	Cada generación selecciona los individuos con el mejor fitness, los cruza e ingresar los nuevos individuos a la población. Pueden ocurrir mutaciones con cierto grado de probabilidad. El estudiante debe definir cada elemento del algoritmo genético y explicarlo adecuadamente en la documentación. Las oleadas son de tamaño variable y se generan con un intervalo parametrizable	10
008	Los enemigos utilizan Pathfinding A* para encontrar el camino hacia el puente del castillo.	20
009	El juego muestra un panel con estadísticas como: <ul style="list-style-type: none"> • Generaciones transcurridas • Enemigos muertos en cada oleada • Fitness de cada individuo de la oleada • Nivel de cada torre • Probabilidad de mutaciones y cantidad de mutaciones ocurridas 	5

ASPECTOS OPERATIVOS

- El trabajo se realizará en **grupos de tres personas**
- El uso de Git y Github es obligatorio
- La fecha de entrega será según lo especificado en el TEC Digital. Se entrega en el TEC digital, un archivo PDF con la documentación. Los estudiantes pueden seguir trabajando en el código hasta 15 minutos antes de la cita revisión oficial.

DOCUMENTACIÓN

- La documentación deberá tener las partes estándar:
 - o Portada
 - o Introducción
 - o Tabla de contenidos (con los títulos debidamente numerados)
 - o Breve descripción del problema
 - o Descripción de la solución
 - Por cada uno de los requerimientos, se deberá explicar cómo se implementó, alternativas consideradas, limitaciones, problemas encontrados y cualquier otro aspecto relevante.
 - o Diseño general: diagrama de clases UML con las clases relevantes que muestren el diseño orientado a objetos y los patrones de diseño aplicados
 - o Enlace al repositorio de Github

EVALUACIÓN

- El proyecto tiene un valor de 20% de la nota del curso
- Los proyectos que no cumplan con los siguientes requisitos no serán revisados:
 - Toda la solución debe estar integrada
 - La interfaz de usuario debe estar implementada e integrada
- El código tendrá un valor total de 80%, la documentación 10% y la defensa 10%. De estas notas se calculará la *Nota Final del Proyecto*.
- Aun cuando el código y la documentación tienen sus notas por separado, se aplican las siguientes restricciones
 - **Si no se entrega documentación en formato PDF, automáticamente se obtiene una nota de 0.**
 - Si no se utiliza un manejador de código se obtiene una nota de 0.
 - Si la documentación no se entrega en la fecha indicada se obtiene una nota de 0.
 - El código debe desarrollarse en C++, si no, se obtendrá una nota de 0.
 - La nota de la documentación es proporcional a la completitud del código. Es decir, si el código no está completo, aunque la documentación lo esté, no se dará el total del 10%
- La revisión de la documentación será realizada por parte del profesor, no durante la defensa del proyecto.
- Cada grupo tendrá 20 minutos para exponer su trabajo al profesor y defenderlo, es responsabilidad de los estudiantes mostrar todo el trabajo realizado, por lo que se recomienda tener todo listo antes de entrar a la defensa.
- Cada grupo es responsable de llevar los equipos requeridos para la revisión, si no cuentan con estos deberán avisar al menos 2 días antes de la revisión a el profesor para coordinar el préstamo de estos.
- Durante la revisión únicamente podrán participar los miembros del grupo, asistentes, otros profesores y el coordinador del área.