

PROGRAMACIÓN ORIENTADA A OBJETOS

Proyecto Final – Mayo 2025

Agradecimiento especial a la estudiante María José Lozada por inspirarse y desarrollar la narrativa de la historia y los personajes y al profesor Mario Mora por definir el esqueleto del proyecto

En un mundo subterráneo, el Reino de Paradise, una colonia de hormigas vivía en armonía y prosperidad, trabajando duro para recolectar alimento y cuidar de su reina. Sin embargo, una amenaza acechaba desde las sombras. Hormigas que antes marchaban en fila perfecta de pronto actuaban desorientadas, parecían títeres moviéndose por inercia hasta que finalmente desaparecían. Un hongo letal, ha comenzado a propagarse desde las sombras.

Alertada y preocupada por esto la reina ha convocado a una hormiga exploradora llamada Ray, para que lleve a cabo una misión secreta para descubrir que ha pasado con sus compañeras sin causar pánico en el hormiguero. En su aventura Ray explora los túneles siguiendo pistas y completando tareas, llegando a los lugares más profundos y abandonados del hormiguero, donde descubre hacia donde se dirigen las hormigas que actúan de manera extraña, una guarida que alberga el cuerpo inerte de sus compañeras y un ente extraño que se alimenta de ellas a través de lo que parecen raíces.

Hay que acabar con ese hongo letal, de lo contrario el Reino de Paradise puede desaparecer.

Primer desafío: Navegar a través de un laberinto subterráneo, siguiendo pistas para descubrir el origen del problema, tras descubrir que las esporas son débiles ante la luz, el Ray deberá recolectar la luminiscencia presente en las paredes de los túneles, evitando que las esporas lo infecten. El Ray deberá usar **habilidades especiales** de defensa y curación para protegerse.

Derrotar al Hongo Letal en un último combate: Iluminado por las luminiscencias recolectadas en el camino, Ray se adentra a la guarida principal y enfrenta una batalla contra hormigas infectadas hasta llegar al líder final, el Rey Hongo, quien es la fuente. Durante la batalla, Ray debe destruir las raíces que dan vitalidad al Rey Hongo y derrotarlo.

REQUERIMIENTOS DEL PROYECTO

Desarrollar un juego de video en C++, con ayuda, no es obligatorio, de una o varias bibliotecas especializadas en gráficas y motores de video juegos. Se recomienda SFML. Tenga en cuenta que puede hacer el videojuego usando una interfaz con símbolos de texto o ASCII.

Los personajes obligatorios jugables son:

- **Ray:** es el personaje con el que se juega y es controlado por el jugador. Es una hormiga exploradora, que porta una máscara (una hoja) y un bastón que brilla en la punta superior.

Vitalidad: La vitalidad del personaje se refleja en el estado de su mascara, si el personaje es infectado varias veces pierde, renueva energías por medio de las semillas

Alimento: semillas (puntos que va recolectando)

Habilidades: Caminar hacia adelante, atrás, saltar, golpear con su bastón y dar golpes de luz con él

Poderes especiales: Super golpe de luz

Equipamiento: Un bastón que emite luz, que debe ir recargando a medida que avanza el juego con los hongos bioluminiscentes presentes en los túneles del hormiguero, este le ayuda a golpear y cuando esta en el punto máximo de luz emite un rayo de luz poderoso

Debilidades: Las esporas que disparan las hormigas infectadas, pero sobre todo las esporas que vienen directamente del hongo.

- **Rey Hongo:** También conocido como el Hongo Letal. Una figura de hongo amorfa y malvada que infecta a las hormigas, convirtiéndolas en títeres para manipularlas a dirigirse hacia él para robarles la energía y crecer.

Vitalidad: Requiere de la energía de las hormigas para su supervivencia y la luz lo lastima

Alimento: la energía de las hormigas

Habilidades: Liberar esporas que infligen daño

Poderes especiales: Control mental, propagación de esporas

Equipamiento: No tiene.

Debilidades: La luz

- **Personajes NPC (non-playable character):** Son personajes de un videojuego que no son controlados por el jugador. Mínimo implementar **Hormigas Normales** y **Hormigas infectadas** (que se comportan erráticamente).

El equipo de trabajo debe definir:

1. Número de vidas
2. Los poderes de los personajes.
3. El Lucky
4. El equipamiento de Ray
5. La salud de Ray y del Rey Hongo
6. La jugabilidad en general
7. Cómo se obtienen los puntos.

Solo pueden utilizar contenedores tipo Vector y Map.

BONUS: Implementar persistencia de Jugadores y Top 10 de mejores puntuaciones. Esta persistencia debe implementarse usando archivos JSON y recuperarse cada vez que se inicie el juego. Debe permitir al jugador antes de iniciar y al finalizar ver el Top 10. Puede encontrar aquí una guía sobre como utilizar archivos json elaborada por el profesor Yoan

<https://github.com/yoanpinzon/POO/blob/master/poo3-persistencia.md>

CRONOGRAMA GENERAL DE ENTREGAS vía GitHub:

Mayo 13: Diagrama de clases versión 1

Mayo 14: Diagrama de clases versión 2

Mayo 20- Mayo 21: Entorno gráfico o de consola - Implementación de los tres escenarios mínimos (El Primer Desafío, Antesala para encontrar el Rey Hongo y el escenario del combate final)

MAYO 27: Entrega y sustentación final. Pondremos a prueba los video juegos. Para las sustentaciones es importante tener un computador disponible en el que pueda hacer modificaciones, eliminaciones, adiciones al proyecto entregado.

Condiciones adicionales

- Los entregables deben ser subidos al repositorio de GitHub del equipo. El documento de autoevaluación debe ser subido al Brightspace. Los documentos deben tener una calidad de **redacción, ortografía y presentación esperadas a nivel universitario**.
- Equipos de 3 personas. Todos los miembros del equipo deben participar activamente en la codificación y ser responsables de diferentes partes del proyecto.
- Puede hacer uso de IA en el proceso de desarrollo del proyecto siempre y cuando pueda dar cuenta de todo lo que está incluido en la solución, así como de las decisiones de diseño que tomen.
- Todos los miembros del equipo deben participar en **todas las etapas** del desarrollo (diseño, codificación, documentación y pruebas funcionales)
- Pueden solicitarme espacios para aclarar dudas o asesorías

Entregables

- **Video de Presentación del Proyecto:** Video de 6 minutos, donde se muestre el proyecto funcionando y se explique cómo se abordaron las mejoras que implementaron. Cada miembro del equipo debe participar y explicar su contribución. Subir link de video en los entregables del proyecto.
- Los documentos deben tener una calidad de **redacción, ortografía y presentación esperadas a nivel universitario**.
- Diagrama UML actualizado como Mermaid dentro del README del proyecto
- Código fuente del programa. Use este github classroom para iniciar el proyecto:
- README en el repositorio [Manual técnico]: Con presentación general del proyecto, imágenes y textos que muestren el cumplimiento de los requerimientos, diagrama de clases UML en formato .MERMAID incluido en el README(uno por equipo).

Organización del Proyecto en GitHub

- Utilizar el esquema de ramas sugerido por GitHub Flow. Cada miembro del equipo debe crear una rama de trabajo por cada funcionalidad o mejora que desee implementar.
- No se deben eliminar las ramas al hacer *merge* con la rama principal, para facilitar la revisión posterior del trabajo realizado.
- Se espera que el proyecto tenga como mínimo 5 pull requests y 20 issues que reflejen la gestión de tareas en la ejecución del proyecto.

Integración Continua (CI/CD):

- Implementar GitHub Actions para configurar una integración continua (CI) que compile el proyecto automáticamente al realizar un commit o merge en la rama principal. Esto ayudará a verificar que el proyecto se mantenga funcional en todo momento.

Autoevaluación y coevaluación

- Informe de autoevaluación en un documento individual en PDF **[SUBIDO EN EL BRIGHTSPACE]** en el que explique:

- Qué problemas tuvo al hacer su proyecto, qué aprendió, qué le gustó, que no le gustó,
- Cómo uso la IA, qué beneficios y dificultades le trajo el uso de la IA
- Qué hizo cada uno de los miembros del equipo
- Evaluación y co-evaluación: califique de manera individual y para cada compañero cada uno de los siguientes elementos.
 - - Colaboración y trabajo en equipo
 - - Responsabilidad y compromiso
 - - Contribución al Desarrollo del Trabajo
 - - Uso de asistentes de IA en el desarrollo del proyecto
 - - Nota consolidada (promedio de las anteriores)

Tenga en cuenta la siguiente rúbrica

Colaboración y trabajo en equipo

- **Excelente (5)** El estudiante participa activamente y mejora la dinámica del equipo mediante una colaboración efectiva, mostrando respeto y apertura hacia las ideas de los demás.
- **Bueno (4)** Participa regularmente y colabora bien, aunque en ocasiones puede ser pasivo. Respetuoso con los compañeros la mayoría del tiempo.
- **Adecuado(3)** Participa sin tomar un rol activo, realiza lo mínimo necesario para no obstaculizar el equipo. Puede mejorar en respeto y colaboración.
- **Insuficiente(2)** Participación mínima o negativa, a menudo no colabora o desestima a los demás, afectando negativamente la dinámica del equipo.

Responsabilidad y compromiso

- **Excelente (5)** Cumple con todas las tareas asignadas a tiempo, mostrando un alto nivel de compromiso y capacidad para tomar iniciativas adicionales.
- **Bueno (4)** Generalmente cumple con las tareas en los plazos establecidos y muestra un buen nivel de compromiso.
- **Adecuado(3)** Cumple con las tareas básicas, pero raramente toma iniciativas adicionales y a veces no cumple con los plazos.
- **Insuficiente(2)** Falta frecuentemente a sus responsabilidades, mostrando poco compromiso y afectando el rendimiento del equipo.

Contribución al Desarrollo del Trabajo

- **Excelente (5).** Aporta ideas y soluciones que son esenciales para el proyecto. Participa de manera activa en la planificación y ejecución de tareas clave.
- **Bueno(4).** Realiza contribuciones que benefician al proyecto. Participa en la planificación y ejecución de tareas.
- **Adecuado(3)** Realiza contribuciones que cumplen con los requisitos básicos del proyecto. Participación regular en la planificación y ejecución de tareas.
- **Insuficiente(2)** Realiza pocas o ninguna contribución al desarrollo del trabajo. Participa poco o nada en la planificación y ejecución de tareas.

Uso de asistentes de IA en el desarrollo del proyecto (Usarlo solo en la autoevaluación)

- **Excelente (5).** Utiliza el asistente de codificación de manera estratégica, integrándolo efectivamente en el flujo de trabajo para mejorar significativamente la productividad y

la calidad del código. Demuestra una comprensión profunda del código generado, realizando adaptaciones precisas que mejoran el proyecto. Muestra un juicio crítico excelente, seleccionando cuándo y cómo utilizar el asistente para optimizar resultados.

- **Bueno (4).** Emplea el asistente de codificación regularmente con una integración generalmente efectiva, contribuyendo positivamente al desarrollo del proyecto. Comprende bien el código generado y realiza ajustes necesarios para alinear el código con los objetivos del proyecto. Demuestra un buen juicio al evaluar la utilidad y precisión del código sugerido, aunque hay espacio para una evaluación más crítica en algunas situaciones.
- **Adecuado (3).** Utiliza el asistente de codificación de forma inconsistente, integrándolo solo parcialmente en el proyecto. Entiende en términos generales el código generado, pero no siempre realiza las modificaciones necesarias para que se ajuste completamente a las necesidades del proyecto. A veces depende demasiado de las sugerencias del asistente sin una evaluación crítica suficiente, afectando la optimización del desarrollo.
- **Insuficiente (2).** Hace un uso esporádico e inefectivo del asistente de codificación, con poca evidencia de que su uso contribuya al proyecto. Incorpora código sin entenderlo adecuadamente, resultando en problemas de integración y funcionalidad. Falta un criterio crítico adecuado, aceptando sugerencias del asistente sin una evaluación pertinente de su aplicabilidad o corrección.

Rúbrica de evaluación

La nota de la profesora dependerá de la sustentación según los criterios que se describen a continuación y su capacidad para sustentar su trabajo.

	5	4	3	2	1	0
Entregables (10%)	Los informes contienen toda la información solicitada y tiene alta calidad en cuanto a estilo y formato.	Se entregaron todos los informes. En términos de contenido están completos pero podría mejorar en cuanto a estilo o formato	Se entregaron todos los informes. En términos de contenido están completos pero podría mejorar en cuanto a estilo Y formato	Faltan algunos de los informes pero los entregados tiene buen estilo y formato	Faltan algunos de los informes y los entregados necesitan mejoras de estilo y formato	No se entregaron los informes
Diseño (30%)	El diseño responde a los requisitos. Se detectaron todas las clases importantes y para cada clase se detectaron los atributos y métodos importantes. Usa las relaciones correctas	El diseño responde a los requisitos. Se detectaron todas las clases importantes y para cada clase se detectaron los atributos y métodos importantes. Tiene algunas relaciones incorrectas o faltantes en el diseño de los métodos como por ejemplo los atributos.	El diseño responde a los requisitos. Faltó detectar algunas clases importantes. Faltó detectar algunos de los atributos y métodos importantes. Tiene algunas relaciones incorrectas. El diseño no corresponde a lo codificado.	Faltó detectar la mayoría de clases importantes Faltó detectar muchos de los atributos y métodos importantes. Tiene muchas relaciones incorrectas	El diseño no satisface los requisitos	No se entregó el diseño
Funcionalidad (30%)	Cumplió con todos los requisitos.	Fueron desarrollados mínimo el 75% de los requisitos	El diseño responde al 75% de los requisitos / podría mejorarse	Fueron desarrollados mínimo el 25% de los requisitos	Fueron desarrollados menos del 25% de los requisitos	La funcionalidad no responde a lo solicitado
Estilo de codificación (5%)	El código se encuentra correctamente indentado, los nombres de los atributos y las funciones cumplen con el estándar de nombramiento. El código tiene documentación interna para facilitar la revisión.	La mayoría del código se encuentra correctamente indentado- La mayoría de los nombres de los atributos y las funciones cumplen con el estándar de nombramiento. La mayoría del código tiene documentación interna para	Falta alguno de los ítems de calidad del estilo de codificación o alguna se cumple con mala calidad	Faltan dos de los ítems de calidad del estilo de codificación o dos se cumple con mala calidad	No hay código fuente suficiente para evaluar el estilo de codificación.	No cumple con el estándar de nombramiento No se encuentra correctamente indentado No está dividido adecuadamente

		facilitar la revisión				
Mejores prácticas (5%)	El código muestra mejores prácticas de desarrollo siempre. Reúso, separación de operaciones, bu en manejo de ciclos, simplicidad, programación defensiva validaciones.	El código muestra mejores prácticas de desarrollo en la mayoría de los casos, pero falta mejorar algunos de los siguientes aspectos Reúso, separación de operaciones, bu en manejo de ciclos, simplicidad, programación defensiva validaciones	El código muestra buenas prácticas de desarrollo pero falta mejorar dos de los siguientes aspectos Reúso, separación de operaciones, bu en manejo de ciclos, simplicidad, programación defensiva validaciones	El código aplica pocas buenas prácticas de desarrollo. Falta mejorar tres de los siguientes aspectos: Reúso, separación de operaciones, bu en manejo de ciclos, simplicidad, programación defensiva validaciones.	No hay código fuente suficiente para evaluar las buenas prácticas.	No se entregó proyecto o el proyecto no cubre al menos el 25% de las funcionalidades. No es posible evaluarlo .

Rúbrica de propiedad intelectual

	1	0.8	0.6	0.4	0.2	0
Sustentación	Es evidente que el estudiante entiende el código que desarrolló lo explica con claridad y responde correctamente a las preguntas.	La sustentación es buena pero se evidenció inseguridad del estudiante para explicar algunas partes del trabajo desarrollado o para responder algunas preguntas.	La sustentación es aceptable se evidencia que el estudiante desarrolló el código pero le cuesta trabajo explicar aspectos del código.	La sustentación es regular se evidenció inseguridad del estudiante para explicar gran parte del trabajo desarrollado o para responder muchas de las preguntas. Parece que el código no hubiera sido desarrollado por el estudiante.	El estudiante demuestra que entiende partes del código, pero no tiene claro cómo se relacionan con la funcionalidad solicitada.	Se evidencia que el estudiante no entiende el código desarrollado, no es capaz de responder a las preguntas formuladas de manera correcta.