



El Juego

Existirán los siguientes elementos

- Uno o más jugadores: deberán ser algún tipo de naves (aviones), cada nave tendrá las siguientes características vida (0 a 100), combustible (0 a 100) y munición (0 a 40). La velocidad de movimiento del jugador es constante, el mismo sólo se podrá mover sobre las aristas del grafo. Además del movimiento entre aristas, se puede contar con movimientos evasivos que permitan esquivar a los atacantes, este movimiento evasivo se definirá como un movimiento perpendicular a la dirección de desplazamiento y máximo 25 píxeles tanto izquierdo como derecho.
- Munición: Durante el transcurso del juego, se encontrará munición sobre el terreno, aparece de forma aleatoria, la munición tendrá diferentes tipos:
 - Básica: Después de cada disparo se debe esperar ciertos milisegundos para que el arma se enfríe y puede volver a disparar. Cada disparo es en línea recta.
 - Pro: Se puede disparar seguido, todo dependerá de la velocidad de presión del jugador. Cada disparo es en línea recta.
 - Difusión: Es de tipo Pro, enviará 5 balas cada una con un ángulo de 36 grados con respecto a la nave.
 - Mayhem: Destruye todo a su paso, después de ser disparada los jugadores quedan aturdidos durante 5 segundos.

El jugador puede cambiar de tipo de munición en el momento que lo desee.

- Puentes: Los puentes aparecen aleatoriamente sobre las aristas del grafo, estos deben ser destruidos para poder continuar, los puentes tendrán una vida aleatoria generada con una distribución de probabilidad gaussiana con una media en 15, cada disparo baja un punto de vida, si se choca contra el puente al jugador se le descontará el equivalente a un 35% de la vida del puente, una vez que se choca el puente desaparece.
- Naves enemigas básicas : Aparecen barcos, aviones, etc de manera aleatoria, la vida al igual que los puentes estará definida por una distribución de probabilidad gaussiana con una media de 3. Estas naves podrán disparar aleatoriamente y dispararán también de forma aleatoria una cantidad de balas por unidad de tiempo, cada bala bajará 2 puntos de vida. Chocar contra estas significa perder la cantidad de vida que estas tienen, cuando se choca, estas desaparecen, se mueven sobre determinadas aristas del grafo formando un ciclo del grafo, aparecen aleatoriamente.
- Naves Perseguidoras: Se encargan de perseguir al jugador sobre todo el grafo, el jugador puede destruirlas, se mueven sobre el grafo como si este fuera no dirigido. A cierta distancia del jugador inician los disparos, tienen munición infinita.
- Kamikaze: Son naves un poco más lentas y débiles, que aparecen aleatoriamente y se dirigen aleatoriamente hacia algún jugador, pueden aparecer n, si se choca contra ella se pierde un 50% de la vida, se destruyen con un disparo.
- Combustible: Cada 100 ms se pierden 2 puntos de combustible, aleatoriamente deberán aparecer puntos para cargar combustible en alguna de las ciudades (nodo del grafo), si se dispara a estos se destruyen y se encienden causando que si el jugador es alcanzado por el fuego pierda un 25% de su vida, la cantidad de combustible que contiene cada tanque es aleatoria gaussiana con una media en 50, la nave debe hacer contacto con el punto para cargar combustible.
- Vida: Cada 15 segundos aparecerá un paquete médico, el mismo tendrá una vida aleatoria

gaussiana con una media en 30, si se dispara a esta se destruye, se debe hacer contacto con esta para tomarla, una vez que aparece en la pantalla tiene una vida útil de 5 ms, si en este tiempo no se toca, se vuelve tóxica entonces al hacer contacto se pierde 5 puntos de vida.

- Jefes: Cada 60 segundos, aparecerá un jefe, los jefes tendrán una vida aleatoria gaussiana con una media en 150. Cada jefe tendrá alguna característica que debe diseñar y documentar cada grupo, para el proyecto se solicitan al menos 5 jefes. Los jefes tienen dos formas, ofensivo y defensivo, cuando este aparece lo hace de forma defensiva, en este estado puede recibir daño (el tiempo que dura el estado es aleatorio) por esto este escapa sobre el grafo como si este fuera no dirigido, cuando el jefe cambia a estado ofensivo este persigue al jugador, en este estado prácticamente no puede recibir daño, si el jugador logra disparar al centro del jefe (un punto muy pequeño) podrá causar mucho daño. Los cambios de estado del jefe y duración son aleatorios, cuando aparecen jefes pueden haber otras naves enemigas, puentes, etc
- Niveles: Cada 60 segundos se pasa de nivel, cada nivel aumenta la cantidad de puentes, naves enemigas, kamikaze y además aumenta la velocidad del juego, también el grafo se hace mas grande.
- Ciudades: Cada ciudad corresponden a un nodo del grafo, existen ciudades móviles que cambian su posición en el mapa, esto afectará las rutas mas cortas calculadas. Las ciudades (aleatoriamente) pueden tener un domo, el mismo debe ser destruido para poder ingresar a está parte del grafo.
- Recompensas: Las recompensas son puntos que el jugador debe recolectar, estos se mueven sobre todo el grafo como si este fuera no dirigido. El valor de la recompensa es aleatorio.

Se deben proporcionar una forma de jugar este juego con el teclado, pueden existir N jugadores, se debe implementar una arquitectura cliente servidor que permite a varios jugadores conectarse al juego, cada vez que un jugador se conecta, aparece automáticamente una nave nueva. Un jugador pierde cuando se queda sin combustible o sin vida, puede continuar pero se penaliza con un 50% de los puntos que tiene hasta el momento. El juego termina cuando se acaban los niveles, se solicitan al menos 5 niveles.

Control

El mando remoto debe ser desarrollado en alguna tecnología para teléfonos celulares, eso queda a decisión de los estudiantes y el dispositivo que tengan, por facilidad en el lenguaje se les recomienda el uso de Android.

Básicamente se debe construir una aplicación para android que trabaje con el acelerómetro, entonces el teléfono será el mando para manejar la nave, mediante la detección de cambios de posición, la aplicación se comunicara con el juego mediante un socket y le indicará que se debe hacer un cambio de posición, la aplicación también contará con botones para poder disparar además de que reproducirá sonidos acorde con lo que sucede en el juego.

5. Documentación

1. Se deberá documentar el código fuente utilizando el estándar Javadoc.
2. Se deberá entregar un documento que contenga:
 - a. Manual de usuario: cómo ejecutar el programa, incluir requerimientos de Hardware y Software
 - b. Descripción de todas las bibliotecas y funciones utilizadas.
 - c. Descripción de los métodos implementados.
 - d. Descripción de las estructuras de datos desarrolladas.
 - e. Descripción detallada de los algoritmos desarrollados.

- f. Problemas conocidos: En esta sección se detalla cualquier problema que no se ha podido solucionar en el trabajo.
 - g. Actividades realizadas por estudiante: Este es un resumen de las bitácoras de cada estudiante (estilo timesheet) en términos del tiempo invertido para una actividad específica que impactó directamente el desarrollo del trabajo, de manera breve (no más de una línea) se describe lo que se realizó, la cantidad de horas invertidas y la fecha en la que se realizó. Se deben sumar las horas invertidas por cada estudiante, sean concientes a la hora de realizar esto el profesor determinará si los reportes están acordes al producto entregado.
 - h. Problemas encontrados: descripción detallada, intentos de solución sin éxito, solución encontradas con su descripción detallada, recomendaciones, conclusiones y bibliografía consultada para este problema específico.
 - i. Conclusiones y Recomendaciones del proyecto.
 - j. Bibliografía consultada en todo el proyecto
3. Bitácora en digital o en papel, donde se describen las actividades realizadas, desde reuniones con el compañero de trabajo, investigaciones, consultas, etc. Esta se puede encontrar hecha a mano, se debe describir todo por mas insignificante que sea, esto demostrará si ustedes están trabajando en realidad. Este es su diario de trabajo, llevan seguimiento de todo en el tiempo, imaginen que si un compañero los releva en su trabajo, le bastaría con leer sus bitácoras para seguir el trabajo.
 4. Diagrama de clases y un documento que explique el porqué del diseño.
 5. Se debe utilizar SVN o algún otro manejador de código fuente. Se recomienda Assembla. Por favor agregar las cuentas asistente.nereo@gmail.com o kevin+asistente@gmail.com a sus repositorios. Recuerden hacer comentarios en cada commit.

6. Evaluación:

1. El juego tendrá un valor de un 60% de la nota final.
2. El mando remoto tendrá un valor de un 30% de la nota final.
3. La documentación ya sea para el juego o para el mando remoto tendrá un valor de entre un 20% y un 50% que será informado a los estudiantes el día de la revisión.
4. Cada grupo recibirá una nota en cada uno de los siguientes apartados, Documentacion - Juego, Código - Juego, Documentación - Mando Remoto y Código - Mando Remoto
5. De las notas mencionadas en el punto anterior se calculará la Nota Final del Proyecto.
6. Se debe enviar el código y la documentación a más tardar a las 23:59 del día de la fecha de entrega a la carpeta compartida del profesor. **El formato del email del subject del email es <Carnet>-<Carnet>-<Carnet>-<Carnet> - Proyecto Crazy River Ride, en el body del email se debe ingresar los carnet y nombres de cada uno de los estudiantes, si el proyecto es en grupos se debe copiar en el email a todos los miembros del grupo.**
7. Las citas de revisión oficiales serán determinadas por el profesor durante las lecciones o mediante algún medio electrónico.
8. Los estudiantes pueden seguir trabajando en el código (no la documentación) hasta 15 minutos antes de la cita revisión oficial, momento en el cuál deben enviar un email con el código fuente, si no lo hacen se asumirá que la versión oficial del código fue la enviada el día de la fecha de entrega junto con la documentación. **El formato del email del subject del email es <Carnet>-<Carnet>-<Carnet>-<Carnet> - Codigo Crazy River Ride, en el body del email se debe ingresar los carnet y nombres de cada uno de los estudiantes, si el proyecto es en grupos se debe copiar en el email a todos los miembros del grupo.**
9. Aún cuando el código y la documentación tienen sus notas por separado, se aplican las siguientes

restricciones

- a. Sí no se entrega documentación, automáticamente se obtiene una nota de 0.
 - b. Sí no se utiliza un manejador de código se obtiene una nota de 0.
 - c. Sí no se entrega el punto 4 de la documentación se obtiene una nota de 0.
 - d. Sí el código y la documentación no se entregan en la fecha indicada se obtiene una nota de 0.
 - e. Sí el código no compila se obtendrá una nota de 0, por lo cuál se recomienda realizar la defensa con un código funcional.
 - f. El código debe ser desarrollado en Java para GNU/Linux, en caso contrario se obtendrá una nota de 0.
 - g. Sí no se siguen las reglas del formato de email se obtendrá una nota de 0.
10. La revisión de la documentación será realizada por parte del profesor, no durante la defensa del proyecto. El único requerimiento que se consultará durante la defensa del proyecto es el diagrama de clases, documentación interna y la documentación en el manejador de código.
 11. Cada grupo tendrá como máximo 20 minutos para exponer su trabajo al profesor y realizar la defensa de éste, es responsabilidad de los estudiantes mostrar todo el trabajo realizado, por lo cuál se recomienda tener todo listo antes de ingresar a la defensa.
 12. Cada excepción o error que salga durante la ejecución del proyecto y que se considere debió haber sido contemplada durante el desarrollo del proyecto, se castigará con 2 puntos de la nota final del proyecto.
 13. 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.
 14. Durante la revisión podrán participar asistentes, otros profesores y el coordinador del área.