

1. Objetivo General

- Desarrollar una aplicación que permita reafirmar el conocimiento de los **paradigmas de programación imperativo y orientado a objetos**.

2. Objetivos Específicos

- Desarrollar una aplicación en el lenguaje de programación C y java.
- Aplicar los conceptos de programación imperativa y orientada a objetos.
- Crear y manipular listas como estructuras de datos.

3. Datos Generales

- El valor del proyecto: 10% (5% C y 5% Java)
- Nombre código: pa**CE** man.
- La tarea debe ser implementada en grupos de no más de 3 personas.
- La **fecha de entrega** es: **7 de Agosto 2020**.
- Cualquier indicio de copia será calificado con una nota de 0 y será procesado de acuerdo al reglamento.

4. Descripción del juego.

Pac Man es uno de los videojuegos más importantes de la historia. El protagonista del videojuego Pac-Man es un círculo amarillo al que le falta un sector, por lo que parece tener boca. Aparece en laberintos donde debe comer puntos pequeños (llamados «Pac-dots» en inglés), puntos mayores y otros premios con forma de frutas y otros objetos. El objetivo del personaje es comer todos los puntos de la pantalla, momento en el que se pasa al siguiente nivel o pantalla. Sin embargo, cuatro fantasmas o monstruos, Shadow (Blinky), Speedy (Pinky), Bashful (Inky) y Pokey (Clyde), recorren el laberinto para intentar capturar a Pac-Man (Wikipedia, 2020).

Pueden jugarlo en línea para que se den una mejor idea en (Pac-Man, 2020).

- 4.1. **El servidor:** deberá estar programado en C y mantendrá la lógica del juego. Será el encargado de crear los fantasmas en momentos aleatorios (cuando el administrador lo decida), pastillas (en el lugar y momento que el administrador lo decida), frutas con un valor aleatorio (introducido por el administrador), asignación de vidas (cada vez que se sumen 10 mil pts), control de niveles (cada vez que se acaben los pequeños puntos), control de velocidad de los

fantasmas (cada vez que el administrador lo desee se puede modificar la velocidad) y el control de la puntuación cada pequeño punto asigna 10pts a la puntuación del jugador.

- 4.2. **El cliente Jugador:** deberá estar programado en java, se refiere a la interfaz gráfica del juego controla a paCE Man que utiliza el usuario, interpretará la estructura enviada por el servidor y le informa al servidor cuando paCE Man come un punto o un fantasma.
- 4.3. **El cliente Observador:** deberá estar programado en java. Se podrá unir a una partida existente pero sólo podrá observar lo que sucede.
- 4.4. **Conexión cliente servidor:** La conexión entre C y Java debe realizarse utilizando Sockets (chuidiang.org, 2020)
- 4.5. **Aspectos de implementación en C:** Recuerden se evaluará que todas las constantes estén en un archivo aparte. Se evaluará el uso de structs.
- 4.6. **Aspectos de implementación en Java:** Recuerden se evaluará que todas este programado según los conceptos de OO (Clases, paquetes, patrones, herencia, polimorfismo).

5. Entregables

- 5.1. Código fuente comentado.
- 5.2. Manual de usuario.

6. Documentación

1. Se deberá entregar un documento que contenga:
 - 1.1. **Descripción de las estructuras de datos desarrolladas.**
 - 1.2. **Descripción detallada de los algoritmos desarrollados (como funcionan los fantasmas).**
 - 1.3. Problemas sin solución: En esta sección se detalla cualquier problema que no se ha podido solucionar en el trabajo.
 - 1.4. Plan de Actividades realizadas por estudiante: Este es un planeamiento de las actividades que se realizaran para completar la tarea, este debe incluir descripción de la tarea, tiempo estimado de completitud y responsable a cargo.
 - 1.5. Problemas encontrados: descripción detallada, intentos de solución sin éxito, soluciones encontradas con su descripción detallada, recomendaciones, conclusiones y bibliografía consultada para este problema específico.
 - 1.6. Conclusiones y Recomendaciones del proyecto.
 - 1.7. Bibliografía consultada en todo el proyecto
2. Bitácora en digital, donde se describen las actividades realizadas, desde reuniones con el compañero de trabajo, investigaciones, consultas, etc. Está se puede encontrar hecha a mano, se debe describir todo por más 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.

7. Evaluación

1. El proyecto tendrá un valor de un 70% de la nota final, debe estar funcional.
2. La documentación tendrá un valor de un 20% de la nota final, cumplir con los requerimientos

- especificados en la documentación no significa que se tienen todos los puntos, se evaluará que la documentación sea coherente, acorde al tamaño del proyecto y el trabajo realizado, no escatimen en documentación.
3. La defensa tendrá un valor de 10%, todos los integrantes del grupo deben participar.
 4. Cada grupo recibirá una nota en cada uno de los siguientes apartados Código, Documentación y defensa.
 5. El profesor no sólo evaluará la funcionalidad del proyecto, esto quiere decir que aunque el proyecto este 100% funcional esto no implica una nota de un 100, ya que se evaluarán aspectos de calidad de código, aplicación del **paradigma imperativo y orientado a objetos**, calidad de documentación interna y externa y trabajo en equipo.
 6. No se revisarán funcionalidades parciales, ni funcionalidades no integradas.
 7. Es responsabilidad de cada miembro del grupo conocer su código, el profesor puede preguntar a cualquier miembro del grupo que le explique alguna funcionalidad/porción de código.
 8. De las notas mencionadas en el punto 4 se calculará la Nota Final del Proyecto.
 9. Las citas de revisión oficiales serán determinadas por el profesor durante las lecciones o mediante algún medio electrónico.
 10. Aun cuando el código, la documentación y la defensa tienen sus notas por separado, se aplican las siguientes restricciones
 - 10.1. Si no se entrega documentación, automáticamente se obtiene una nota de 0.
 - 10.2. Si no se entrega el punto 4 de la documentación se obtiene una nota de 0.
 - 10.3. Si el código y la documentación no se entregan en la fecha indicada se obtiene una nota de 0.
 - 10.4. Si el código no compila se obtendrá una nota de 0, por lo cual se recomienda realizar la defensa con un código funcional.
 - 10.5. Si el grupo no cuenta con los equipos necesarios para realizar la revisión y no avisó al profesor de esta situación obtendrá una nota de 0.
 - 10.6. El código debe ser desarrollado en **el lenguaje de programación C** utilizando el **paradigma de programación imperativo** y java utilizando el **paradigma de programación orientado a objetos**, en caso contrario se obtendrá una nota de 0.
 - 10.7. **NO** presentarse a la defensa se obtendrá una nota de 0.
 11. Cada grupo tendrá como máximo 30 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 cual 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.
 14. Durante la revisión únicamente podrán participar los miembros del grupo, asistentes, otros profesores y el coordinador del área.
 15. Las revisiones se realizan con los estudiantes matriculados en el curso, cualquier persona fuera de estos y los mencionados en el punto 13, no pueden participar en la revisión.
 16. Después de enviada la nota final del proyecto el estudiante tendrá un máximo de 3 días hábiles para presentar un reclamo siempre y cuando la funcionalidad esté completa.

8. Referencias

chuidiang.org. (2020, 3 21). *Socket entre C y java*. Retrieved from Socket entre C y java:

http://www.chuidiang.org/java/sockets/cpp_java/cpp_java.php

Pac-Man. (2020, 04 02). *Pac Man*. Retrieved from webpacman: <https://www.webpacman.com/>

Wikipedia. (2020, 04 02). *Pac-Man*. Retrieved from Wikipedia: <https://es.wikipedia.org/wiki/Pac-Man>