

Trabajo Práctico - Taller de Programación Avanzada

El otro Pac-Man

[Consigna](#)

[Mecánica del juego](#)

[Requisitos \(en forma de historias de usuario\)](#)

[Fechas de Entrega](#)

[Modalidad de trabajo](#)

Consigna

El trabajo consiste en la realización de una aplicación cliente-servidor, la cual simulará el funcionamiento del juego Pac-Man en modo multijugador.

Mecánica del juego

El juego repite la mecánica del [Pac Man](#), juego clásico de los '80. Sin embargo, se le agrega la posibilidad de jugar en modo multijugador.

Una partida dura 1 minuto, durante el cual los jugadores tienen puntaje asignado en función a su desempeño:

- Si el jugador que maneja a Pac Man consume todos los puntos del escenario, gana la partida.
- Caso contrario, gana el fantasma que más veces haya “comido” al Pac Man.

Para que todo esto funcione de manera equilibrada, se agregan las siguientes reglas al juego:

- No se puede jugar de a menos que tres jugadores (dos fantasmas, un solo Pac Man). Los roles se asocian en forma aleatoria.
- Al “comer” a Pac Man, éste se regenera en un sitio alejado de todos los fantasmas (en promedio). **No debe ser** un sitio aleatorio.
- Si los fantasmas se “tocan”, rebotan un casillero y por dos segundos no pueden moverse ni “comer” al Pac Man.
- Los fantasmas se mueven a la mitad de la velocidad que el Pac Man.



Requisitos (en forma de historias de usuario)

- Como potencial jugador quiero registrarme para participar del juego
- Como jugador quiero ingresar al sistema para jugar
- Como jugador quiero elegir mi nick
- Como jugador quiero ver mis estadísticas de victorias, uso de cada tipo de personaje y tiempo jugado en total
- Como jugador quiero poder unirme a una partida
- Como administrador quiero crear una partida, para que otros jugadores se unan a la misma
- Como administrador quiero ver cualquier partida en curso, a pesar de no estar jugando
- Como jugador quiero utilizar el teclado para moverme por el escenario
- Como jugador quiero disfrutar de una física realista en el juego, que detecte colisiones

Fechas de Entrega

- Diagrama de clases: 26/09/2015
Se pide el diagrama de clases de dominio del sistema.
- Pantallas con Swing: 10/10/2015
Se piden las pantallas principales, armadas con la biblioteca de Swing.
- Primera y segunda funcionalidad: 31/10/2015
Se piden dos funcionalidades a elección, terminadas
- Pre-entrega: 14/11/2015
Una entrega cuasi-formal, para ver pequeños ajustes restantes
- Entrega final: 21/11/2015
Entrega y defensa del trabajo realizado

Modalidad de trabajo

Se trabajará en equipos de **4 personas**.

Documentación para la Entrega Final

- Se deberá entregar los diagramas de clases correspondientes utilizando una herramienta CASE.
- Se entregará las métricas y estimaciones del proyecto.
- Se debe entregar un documento con el análisis y diseño de la aplicación.
- Se debe entregar las pruebas realizadas de la aplicación. Definir escenarios, descripción de las pruebas y cuáles fueron sus resultados. En caso de errores indicar si fueron solucionados.
- Se debe entregar todo el código fuente del proyecto. La demo de la entrega se debe realizar con jar ejecutables. No se permite la demo desde la IDE de desarrollo.
- Se debe agregar documento con la instalación y cualquier configuración que la aplicación necesite para poder ejecutarse.
- Se permite para la demo una ejecución armando su propia red. (servidor y 2 clientes como mínimo) Puede utilizarse las PC del lab, recordar que no es posible instalarse software ya que no tenemos usuario administrador. (Para el caso de la Base de Datos)
- Todo se debe entregar en formato digital, la documentación es obligatoria y forma parte de la aprobación del TP.