

**ESCUELA COLOMBIANA DE INGENIERIA**  
**PROGRAMACIÓN ORIENTADA A OBJETOS**  
**POOBthogen 2014-02**

Propuesta : Nicolas Gomez y Miguel Romero

**PATHOGEN**

**Pathogen** es un revolucionario juego de estrategia para dispositivos móviles y tablets. Diseñado con la clásica experiencia de juegos de mesa, **Pathogen** brinda la mecánica y el estilo de juegos clásicos como Ajedrez y Go<sup>1</sup> a un nivel completamente nuevo, fácil de aprender, pero con una gran variedad de estrategias. Las características de **Pathogen** son: <sup>2</sup>

- Estrategia Interminable
- Extensa campaña de un jugador
- Múltiples modos de juego y mapas
- Editor de mapas
- Desafío de amigos en persona u online

El comportamiento del juego se puede observar más detalladamente en los siguientes videos:

- <https://www.youtube.com/watch?v=ZgPXOgsjmb8>
- [https://www.youtube.com/watch?v=4Zrbp\\_kRv](https://www.youtube.com/watch?v=4Zrbp_kRv)



**POOBthogen**

Es un juego de estrategia para dos jugadores cuyo objetivo es cubrir la mayor porción de un tablero rectangular con el “virus” aliado. El tablero está compuesto de casillas rectangulares de igual tamaño que puede contener una ficha enemiga, aliada o neutral. Las fichas neutrales pueden ser infectadas por cualquier jugador para aumentar su distribución en el tablero..

El juego se desarrolla por turnos en los que cada uno de los jugadores coloca un virus en el tablero con el fin de ocupar la mayor porción de este.

Los jugadores disponen de diferentes tipos de virus cada uno se caracteriza por las condiciones que requiere para poder ubicarse en una posición, el efecto que causa, si existía un virus en la casilla que él ocupó, y su propia transformación al ubicarse.

Cada vez que se coloca un virus sobre otro este “evoluciona” lo que significa que se van a crear nuevos virus en las celdas adyacentes al mismo (verticales y horizontales) del mismo tipo que el original replicándose el efecto.

Los bloques son generados por la transformación de algunos tipos de virus, estos elementos no pueden ser eliminados de ninguna forma y perduran hasta el final del juego.



<sup>1</sup> <http://es.wikipedia.org/wiki/Go>

<sup>2</sup> <http://pathogen.co/>

## JUGADORES

Los jugadores pueden ser controlados por un usuario o por el computador. Los jugadores automáticos pueden asumir diferentes perfiles, entre ellos:

1. Tímido: en lo posible, no ataca al jugador enemigo
2. Ofensivo: busca causar en cada momento el mayor daño posible.
3. Irreflexivo: realiza cualquier movimiento

Los movimientos que pueden realizar los heroes son los siguientes:

Colocar virus	Coloca un nuevo virus en una casilla
Pasar turno	Decidir no hacer ningún movimiento si no lo requiere.
Rendirse	Decidir no seguir jugando. En este caso el oponente gana sin importar el estado del juego.

## VIRUS

El sistema puede permitir cualquier tipo de virus cuyo comportamiento será implementado en una clase con su nombre. Los cuatro virus básicos están descritos en los documentos previos.

---

## REQUISITOS FUNCIONALES

La aplicación debe:

- Permitir generar aleatoriamente tres tipos de tableros con un número de casillas determinado
- Permitir importar y exportar un campo de juego
- Permitir abrir y salvar el estado de un juego
- Permitir seleccionar el tipo de oponente, bien sea persona o un tipo específico de máquina
- Permitir a los jugadores elegir el color deseado
- Permitir establecer, si se desea, el tiempo máximo de juego
- Permitir establecer, si se desea, la cantidad de turnos
- Permitir indicar cual jugador va a realizar el primer movimiento.
- Permitir realizar los movimientos
- Ilustrar permanentemente el estado del de juego y establecer la proporción de casillas ocupadas por cada jugador
- Permitir que el jugador termine el juego en cualquier momento
- Terminar el juego y comunicar su causa

## REQUISITOS DE DISEÑO

### De extensión

- Permitir generar nuevas versiones de la aplicación que incluyan otros tipos de virus (por ejemplo, el extremo) y otros perfiles de jugador computador (por ejemplo, en la competencia, el nuevo perfil con la estrategia)

### De visualización

- El tablero debe tener una representación gráfica adecuada que permita conocer el estado del juego.
- El jugador que realiza el primer movimiento siempre se ubica al sur del tablero.

### De manejo de excepciones

- Deben definir mínimo una nueva clase excepción para manejar las excepciones propias.
- Los métodos correspondientes a las acciones deben lanzar una excepción si la acción solicitada no es válida explicando claramente su causa.
- Cuando ocurra una excepción no esperada o una propia grave se debe escribir esta información en el log de errores para los programadores y terminar la ejecución del mismo.

## REQUISITOS DE ENTREGA

<b>Versión uno</b> <b>Persistencia</b>	Generar tablero Importar y exportar tablero Salvar guardar estado de juego	CAPA APLICACIÓN Diagrama de clases Diagramas de secuencia Código programa Código pruebas JUnit	A par Martes 11 de noviembre
	Realizar movimiento		
<b>Versión dos</b> <b>Presentación</b>	Leer tablero Generar tablero Salvar guardar estado de juego  Configurar un juego Juego persona-persona	<b>CAPA PRESENTACIÓN</b> <b>Boceto del la interfaz gráfica</b> <b>Diagrama de clases</b> <b>Código</b> CAPA APLICACIÓN Diagrama de clases Diagramas de secuencia Código programa Código pruebas JUnit	A par Martes 25 de noviembre
<b>Versión tres</b>	Funcionamiento completo <b>PRUEBAS DE ACEPTACIÓN</b>	CAPA PRESENTACIÓN Boceto del la interfaz gráfica Diagrama de clases Código CAPA APLICACIÓN Diagrama de clases Diagramas de secuencia Código programa Código pruebas	A evaluador Inicial Lunes 1 de diciembre Final Viernes 5 de diciembre
<b>COMPETENCIA</b>	Es requisito para participar en la competencia que el equipo se haya presentado a todas las revisiones de pares y que todas las entregas del proyecto hayan sido aprobadas. El equipo ganador tiene 5.0 en la nota del tercer tercio.		Viernes 12 de diciembre