Proyecto 1 - Herencia y polimorfismo

Yuen Law

Semestre I, 2023

Objetivos

1. Aplicar conceptos de ingeniería de software.
2. Aplicar conceptos básicos de POO: Herencia y polimorfismo

Descripción

El objetivo de este proyecto es desarrollar un juego/simulación de microorganismos utilizando herencia y polimorfismo para modelar los diferentes microorganismos. Para esto, se debe implementar un organismo básico que tenga comportamientos y atributos generales. De este organismo heredarán los otros organismos.

Microorganismos NPC

El comportamiento de los microorganismos que queremos simular es como sigue: los organismos buscan alimento para poder defenderse de los otros mircroorganismos. Todos los organismos tienen las siguientes características: energía, visión, velocidad y edad. Estas caracerísticas deben tener máximos y mínimos.

Estas características funcionan de la siguiente manera:

1. La energía permite al organismo moverse
2. La energía incrementa cuando los organismos se alimentan,
3. La energía disminuye con cada movimiento.
4. La visión permite al organismo ver en un radio a su alrededor y detectar otros organismos y alimento.
5. La visión incrementa con ciertos alimentos.
6. La visión decrementa con la edad.
7. La velocidad determina cuántos pasos puede moverse el organismo en un solo turno.
8. La velocidad incrementa con ciertos alimentos.
9. La velocidad decrementa conforme aumenta la energía.
10. La edad aumenta con cada turno.

11. El grado en el que las características incrementan o decrementan debe ser configurable al inicio de la aplicación.

Los microorganismos buscan alimentarse ya sea con alimento o de otros microorganimos. Cuando los organismos se encuentran, la decisión de quién come a quién se determina de la siguiente manera:

1. El organismo con mayor energía gana.
2. Si los 2 tienen igual energía, gana el de mayor velocidad.
3. Si los 2 tienen igual velocidad, gana el de mayor edad.
4. Si los 2 tienen igual edad, se selecciona un ganador aleatoriamente.

Además cuando un organismo consume a otro obtiene la mitad de sus características: energía, visión y velocidad.

Cada organismo se moverá y tomará decisiones independientemente, pero basado en información de su entorno, y según su personalidad. Algunos organismos priorizarán visión o velocidad, pero todas tratarán de incrementar energía. Todos huirán de situaciones de peligro.

Jugador

El jugador controla un microorganismo con las mismas características que los NPC. La diferencia es que el comportamiento lo determina el jugador y no un algoritmo.

Alimento

Los organismos consumen también partículas que les proveen energía, mejoran la velocidad o la visión. Estas partículas se mantienen en su posición hasta que sean consumidas. Las partículas que dan energía vienen en diferentes "tamaños".

Mapa

El mapa es un tablero de al menos 50×50. Los organismos ocupan 1 sola casilla y se mueven en 4 direcciones: arriba, abajo, izquierda y derecha. Las partículas de alimento ocupan una casilla también.

Simulación

No es necesario que la simulación corra en tiempo real, es decir, no es necesario usar hilos. Se recomienda tener estructuras de datos, por ejemplo listas, que se recorren en ciclos para determinar la

siguiente acción de cada organismo. Aquí es necesario aplicar polimorfismo, es decir, solo se puede usar una estructura para todos los agentes y el procesamiento llamará al mismo método de cada uno. No debe haber código de la forma (if (tipo==a) ejecute x(); else ejecute y()). Una vez que se hayan procesado todas las estructuras, se actualiza el mapa para reflejar todos los cambios.

La simulación completa se ve entonces de la siguiente manera:

1. Se genera el mapa, incluyendo al jugador, partículas de alimento.
2. Se crea la lista de organismos en posiciones aleatorias en el mapa.
3. Después de cada movimiento del jugador:
   a) Se recorre la estructura de organismos para determinar su siguiente acción, i.e. hacia donde se moverán, etc.
   b) Si una partícula de alimento es consumida, se genera una nueva para reemplazarla.
   c) Si un organismo es consumido se genera uno nuevo para reemplazarlo.
   d) Se actualiza la interfaz gráfica.

GUI

La interfaz gráfica no debe ser muy compleja. Los organismos y partículas pueden ser representados simplemente con un color diferente en la casilla. Sí debe ser posible ver información de los organismos para validar su compartamiento durante la revisión del proyecto.

Requerimientos

- Se debe programar usando el paradigma orientado a objetos.
- Se debe aprovechar al máximo la herencia y el polimorfismo, recuerde, no debe haber código de la forma (if (tipo==a) ejecute x(); else ejecute y()).
- Debe preparar y presentar un avance y presentarlo al menos 2 semanas antes de la entrega final. El avance debe incluir un diagrama de clases lo más completo posible con un plan de desarrollo.
- Debe implementar una interfaz gráfica
- Para el control de desarrollo se debe usar la pizarra de proyecto en Github y el sistema kanban para registrar el progreso de las actividades realizadas.

Entregables

1. Documentación PDF, usando una plantilla de Overleaf (sugerencia) con una descripción general de la solución implementada y el diagrama de clases y una breve explicación de cada clase. Además deben incluir una sección donde detallen cómo se aplicó la herencia y polimorfismo, así como ventajas y desventajas que observaron durante el desarrollo del proyecto. Por último debe incluir una sección de referencias donde detalle los recursos externos utilizados para el desarrollo del proyecto, por ejemplo, tutoriales, ejemplos de código, videos, etc.
2. Código en un repositorio de Git. El último commit realizado debe ser antes de la fecha y hora límite de entrega.

Aspectos administrativos

1. Fecha de entrega: Viernes 31 de marzo
2. El pdf debe ser enviado por medio de Teams
3. El proyecto debe ser desarrollado en Java.
4. Es obligatorio presentar el avance.
5. Después de la entrega, el proyecto será revisado en conjunto con el profesor.

Evaluación

Rubro	Valor
Documento PDF	(15)
- Descripción	2.5
- Diagrama de clases	5
- Herencia y polimorfismo	5
- Referencias	2.5
Código	(65)
- GUI	5
- Mapa	5
- Clase organismos	7.5
- Clase alimento	7.5
- Uso de la Herencia	7.5
- Comportamiento de los organismos	10
- Simulación y juego	
– Uso correcto de polimorfismo	7.5
– Estructura de datos	5
– Procesamiento	5
– Actualización	5
Avance	(10)
Workflow	(10)
- Uso de git	5
- uso del board de github	5
Total	100