FIUBA - 75.07

Algoritmos y programación III

Trabajo práctico 2: AlgoFormers

1er cuatrimestre, 2016

(trabajo grupal de 4 integrantes)

Alumnos:

Nombre	Padrón	Mail
Funes Federico	98372	
Zaffaroni Torre Joaquin	98314	
Cano Matias	97925	
Song John	97038	

Fecha de entrega final: Jueves 23/06/2016 - acordar con ayudantes.

Tutor: Yolis Eugenio

Comentarios:

Introducción

Objetivo del trabajo

Aplicar los conceptos enseñados en la materia a la resolución de un problema, trabajando en forma grupal y utilizando un lenguaje de tipado estático (Java)

Consigna general

Desarrollar la aplicación completa, incluyendo el modelo de clases e interfaz gráfica. La aplicación deberá ser acompañada por pruebas unitarias e integrales y documentación de diseño. En la siguiente sección se describe la aplicación a desarrollar.

Descripción de la aplicación a desarrollar

Se deberá desarrollar una aplicación que implemente un juego relacionado con el clásico cómic de los Transformers. Todos los algoformers presentan las siguientes características en cualquiera de sus modos:

- Nombre
- Puntos de vida
- Ataque
- Distancia de ataque
- Velocidad de desplazamiento

Todos los algoformers tienen un modo humanoide (que siempre es terrestre) y su modo alterno, que será distinto para cada uno.

Autobots

1. Optimus Prime, líder de los Autobots. Su modo alterno es un Peterbilt 379 azul con llamas rojas (unidad terrestre)

OPTIMUS	Humanoide	Alterno
Ptos de Vida	500	
Ataque	50	15
Distancia de ataque	2	4
Velocidad	2	5

1. Bumblebee, el joven explorador de los Autobots y guardián de Sam. Su modo alterno es un reluciente Chevrolet Camaro Concept de 2006. (unidad terrestre)

Bumblebee	Humanoide	Alterno
Ptos de Vida	350	
Ataque	40	20
Distancia de ataque	1	3
Velocidad	2	5

1. Ratchet, Su modo alterno es un F22 raptor (unidad aérea)

Ratchet	Humanoide	Alterno
Ptos de Vida	150	1.
Ataque	5	35
Distancia de ataque	5	2
Velocidad	1	8

1. Los 3 combinados forman un Superion (unidad terrestre), la transformación dura 2 turnos propios hasta completarse. Queda a criterio del grupo definir la distancia mínima a la que tienen que estar los algoformers entre sí para formar un Superion.

SUPERION	modo único	
Ptos de Vida	∑ Ptos de vida de los algoformers que lo forman	
Ataque	100	
Distancia de ataque	2	
Velocidad	3	

Decepticons

1. Megatron, líder de los Decepticons. Su modo alterno es un jet cibertroniano. (unidad aérea)

MEGATRON	Humanoide	Alterno
Ptos de Vida	550	
Ataque	10	55
Distancia de ataque	3	2
Velocidad	1	8

1. Bonecrusher, el desbocado buscaminas de los Decepticons. Su modo alterno es un vehículo blindado Force Protection Industries Buffalo HMPCV buscaminas. (unidad terrestre)

Bonecrusher	Humanoide	Alterno
Ptos de Vida	200	

Ataque	30	30
Distancia de ataque	3	3
Velocidad	1	8

1. Frenzy, un pirata informático de los Decepticons. Su modo alterno es una Renault Duster. (unidad terrestre)

Frenzy	Humanoide	Alterno
Ptos de Vida	400	1.
Ataque	10	25
Distancia de ataque	5	2
Velocidad	2	6

1. Los 3 combinados forman un Menasor (unidad terrestre), la transformación dura 2 turnos hasta completarse. Queda a criterio del grupo definir la distancia mínima a la que tienen que estar los algoformers entre sí para formar un Menasor.

MENASOR	modo único	
Ptos de Vida	∑ Ptos de vida de los algoformers que lo forman	
Ataque	115	
Distancia de ataque	2	
Velocidad	2	

Ataques y distancia de ataques

Los algoformers de un mismo equipo **no** pueden atacarse entre ellos.

SIMPLIFICACIÓN: Los algoformers pueden atacar en cualquiera de las formas, es todo igual:			
tierra-tierra	tierra-aire	aire-tierra	aire-aire

La distancia de ataque se mide en casilleros. Por ejemplo OPTIMUS en modo humanoide posee distancia de ataque = 2 significa que podrá atacar a cualquier otro algoformer que se encuentre en un casillero verde, no así en los celestes.

			OPTIMUS			

Superficies:

Tierra

1. **Rocosa**: Todas las unidades pueden atravesarla.
2. **Pantano**: En modo humanoide no es posible atravesarlo. En modo alterno las unidades terrestres tardan el doble que una superficie rocosa.
3. **Espinas**: Causa un 5% de daño a quien la atraviese, independientemente del modo en que se halle.

Aire

1. **Nube**: Todas las unidades aéreas pueden atravesarla.
2. **Nebulosa de andrómeda**: Las unidades aéreas quedan atrapadas 3 turnos.
3. **Tormenta psiónica**: Al pasar por una tormenta psiónica las unidades aéreas pierden poder de ataque (sólo del modo alterno) y el mismo queda disminuido en un 40 % **para siempre**. (En el contexto de una partida). El daño no es acumulable en caso de que el algoformer ya afectado vuelva a pasar por una tormenta, no lo afecta.

Bonus

Una vez que el algoformer (en cualquier modo) captura un bonus el mismo es consumido por el algoformer y desaparece del mapa.

1. **Doble Cañón**: El algoformer que se tope con este bonus, duplica su capacidad de ataque durante 3 turnos propios.
2. **Burbuja inmaculada**: El algoformer que se tope con este bonus, no recibe ningún daño por ningún tipo de ataque de otro algoformer durante 2 turnos propios.
3. **Flash**: El algoformer que se tope con este bonus triplica su velocidad de desplazamiento durante 3 turnos propios.

Ejemplo de Doble cañón:

Turno Autobots:

Optimus Humanoide captura doble cañón

Turno Decepticons

….

Turno Autobots:

Optimus Humanoide ataque = 100 (2 x 50)

Turno Decepticons

….

Turno Autobots:

Optimus Humanoide ataque = 100 (2 x 50)

Turno Decepticons:

….

Turno Autobots:

Optimus Humanoide ataque = 100 (2 x 50)

Turno Decepticons:

….

Turno Autobots:

Optimus Humanoide ataque = 50

Queda a criterio de cada grupo definir qué pasa cuando se combinan los 3 algoformers para formar un Menasor o Superion y alguno de ellos tiene un bonus activo.

Jugabilidad

Hay 2 jugadores, cada uno debe elegir un equipo antes de iniciar una partida. Cada jugador comienza la partida con sus 3 algoformers.
Es un juego por turnos. En cada turno el jugador debe elegir UN algoformer y solicitarle que realice una actividad (moverse, transformarse, atacar, combinarse, capturar chispa, etc…). Luego pasará el turno al contrincante y así sucesivamente hasta la captura de la chispa suprema.
El juego elige al azar qué jugador comienza. Cada jugador inicia en el extremo opuesto al otro con sus 3 algoformers juntos.

Tablero

El juego tiene lugar en un tablero compuesto de casilleros. El tamaño, forma y cantidad de casilleros del tablero queda a definir por cada grupo y acordado con su ayudante.
Todos los algoformers ocupan 1 casillero en cualquiera de sus modos. No puede haber más de 1 algoformer en un casillero. Hay 1 o 0, nunca 2, 3, etc.
Los algoformers se desplazan por el tablero de casillero en casillero. Cada punto de su velocidad de desplazamiento representa 1 casillero. Por ejemplo, OPTIMUS en modo alterno posee una velocidad de desplazamiento = 5 lo que nos lleva a la siguiente configuración:

Estado inicial:

OPTIMUS					

Optimus se mueve ⇒

Estado final:

					OPTIMUS

Otro ejemplo

Megatron en modo humanoide posee una velocidad de desplazamiento = 1. Con lo cual desde dónde está ubicado actualmente se puede mover a cualquiera de los casilleros verdes. Como puede verse, aplica la misma lógica que en distancia de ataque.

			MEGATRON			

Fin del juego

Para ganar se debe capturar la chispa suprema. que será ubicada de forma aleatoria cerca del centro del tablero. Cualquier algoformer en estado Humanoide es capaz de capturar la chispa suprema, no así en su modo alterno. Los superion y menasor también pueden atrapar la chispa suprema. Si un jugador logra destruir a todos los algoformers del jugador contrario, también gana en ese caso.

[OPCIONAL] Fin del juego alternativo (suma puntaje extra)

Para ganar el juego el algoformer que captura la chispa suprema debe transportarla hasta el monte de la perdición donde es arrojada. ¡ Pero cuidado ! Si en el camino el algoformer que la transporta es destruido, la chispa suprema puede ser recapturada por el jugador contrario.

Interfaz gráfica

Se debe desarrollar una interfaz visual para la interacción entre los jugadores. En la misma se pondrá mucho énfasis y se evaluará como parte de la consigna la **USABILIDAD** de la misma.
Cada vez que le toque el turno a cada jugador la vista del mapa debe centrarse en el lugar donde utilizó a su último algoformer.

Entregables

1. Código fuente de la aplicación completa, incluyendo también: código de la pruebas, archivos de recursos
2. Script para compilación y ejecución (ant)
3. Informe, acorde a lo especificado en este documento

Formas de entrega

Habrá **4 entregas formales.** Las mismas tendrán una calificación de **APROBADO o NO APROBADO** en el momento de la entrega.

Aquél grupo que acumule 3 no aprobados, quedará automáticamente desaprobado con la consiguiente pérdida de regularidad en la materia. En cada entrega se debe traer el informe actualizado.

Evaluación

El día del vencimiento de cada entrega, cada ayudante convocará a los integrantes de su grupo, solicitará el informe correspondiente e iniciará la corrección mediante una entrevista grupal.
Es imprescindible la presencia de todos los integrantes del grupo el día de cada corrección.
Se evaluará el trabajo grupal y a cada integrante en forma individual. El objetivo de esto es comprender la dinámica de trabajo del equipo y los roles que ha desempeñado cada integrante del grupo. Para que el alumno apruebe el trabajo práctico debe estar aprobado en los dos aspectos: grupal e individual.
Dentro de los ítems a chequear el ayudante evaluará aspectos formales (como ser la forma de presentación del informe), aspectos funcionales: que se resuelva el problema planteado y aspectos operativos: que el TP funcione integrado.

Casos de prueba para cada entrega

1er Entrega Jueves 02/06/2016 - Lunes 06/06/2016 - *2 semanas desde fecha inicio* Lunes 30/05/2016 para el turno tarde

Turnos, Jugadores, Unidades, Tablero, casillero

1. Se ubica un algoformer humanoide en un casillero, se pide que se mueva, se verifica nueva posición acorde a su modo.
2. Se ubica un algoformer humanoide se lo transforma, se verifica que se pueda transformar en ambas direcciones.
3. Se ubica un algoformer en su modo alterno y se pide que se mueva y se verifica que su nueva posición sea acorde.
4. Crear una prueba de integración en la cual se pueda crear un juego, con 2 jugadores cada uno de ellos con sus 3 algoformers distribuidos en el tablero según el enunciado y la chispa suprema por el centro del tablero.
5. Combinaciones en modos de: Ubicar un autobot, ubicar un decepticon, pedir que se ataquen respetando (y no) las distancias verificando los daños (o no daños).

2da Entrega Jueves 09/06/2016 - Lunes 13/06/2016 - *3 semanas desde fecha inicio* Lunes 6/6/2016 para el turno tarde

1ra entrega + Interacción con superficies

1. Llenar una zona rocosa, verificar que todos los algoformers en todos sus modos la atraviesen sin problemas
2. Llenar una zona pantano, verificar que en modo humanoide no se pueda atravesar.
3. LLenar una zona pantano, verificar que en modo alterno las unidades terrestres tardan el doble que rocoso
4. LLenar una zona pantano, verificar que las unidades aéreas las atraviesan sin problemas
5. Llenar una zona de espinas verificar que todas las unidades terrestres pierden un 5% de sus vida por cada casillero de estos que atraviesen
6. LLenar una zona de espinas, verificar que unidades aéreas no tienen problemas al atravesarlas.
7. Llenar una zona con nubes, verificar que las unidades aéreas las atraviesan sin problemas
8. Llenar una zona de nebulosa de andrómeda, pasar una unidad aérea, corroborar que quede 3 turnos atrapada, sin moverse
9. Llenar una zona de tormenta psiónica, pasar un algoformer alterno aéreo, ver que baje su capacidad de ataque
10. test 9 + volver a pasar y ver que no bajó su capacidad de ataque.

3er Entrega: Jueves 16/06/2016 - Lunes 20/06/2016 - *4 semanas desde fecha inicio* Lunes 13/6/2016 para el turno tarde

2da entrega + Bonus, Interfaz gráfica inicial

1. Ubico un algoformer, ubico un bonus doble cañón,ubico otro algoformer enemigo, el algoformer captura el bonus y ataca al enemigo verificando que causa el doble de daño durante 10 turnos.
    a. Repetir para el modo alterno.
2. Ubico un algoformer, ubico un bonus burbuja, ubico un otro alfoformer enemigo, el algoformer captura el bonus, el otro algoformer ataca al primer algoformer, este no recibe daños, repetir hasta 2 turnos propios, continuar y verificar que en el 3ro sí reciba daño.
    a. Realizar el mismo test en modo alterno
3. Ubico un algoformer, ubico un bonus flash, verifico que se mueve 3 veces más rápido durante 3 turnos propios.
    a. Repetir en modo alterno
    b. Repetir en modo humanoide-alterno-humanoide
4. Test boundary cases (Si ya tiene un bonus de un tipo que no pueda agarrar otro del mismo tipo, Atrapar 2 bonus distintos verifcar ambos comportamientos, etc…)

4ta y última Entrega: Jueves 23/06/2016 - Lunes 27/06/2016 - *5 semanas desde fecha inicio* Lunes 20/6/2016 para el turno tarde

Trabajo Práctico completo funcionando, con interfaz gráfica final, sonidos e informe completo.

Tiempo total de desarrollo del trabajo práctico: 5 semanas completas.

Informe

Supuestos

Para la resolucion del TP, utilizamos los siguientes supuestos:

- Cuando se crea un Algoformer, su modo por defecto es Humanoide

- Siempre el primer turno corresponde al jugador 1

- Los Algoformers no pueden desplazarse a una posicion fuera del tablero

- La lista de los movimientos que puede realizar un algoformer es valida, son adyacentes y existen (va a ser contemplado en la interfaz visual)

- Las distintas superficies son ubicadas en posiciones aleatorias en el tablero

Modelo de dominio

Para la resolucion del TP, lo desarrollamos a traves de TDD con las pruebas unitarias que fuimos construyendo a medida que avanzaba en el TP ya que de esta manera podriamos trabajar todos los integrantes buscando un comportamiento comun de la aplicacion. El primer paso para encarar el TP fue separar los comportamientos y crear las clases correspondientes para una buena estrategia de diseño.
En primer lugar, para realizar una representación del juego a desarrollar, se decidió implementar una clase Juego que administre todo lo relacionado con el juego. Es decir, es la clase principal de nuestro modelo, y se encarga de controlar tanto a los jugadores, como al tablero o a los personajes. El Juego contiene a dos jugadores y a un tablero. Estos últimos dos conceptos también requirieron la creación de clases respectivas.
Por un lado la clase Jugador conoce al Tablero y a sus personajes. De esta forma, es el encargado de realizar los movimientos de los personajes en el Tablero. Por otro lado, la clase Tablero, esta compuesta por posiciones. El Tablero se modelo como un HashMap con posiciones (para las cuales se creo una clase Posición), en cada posición puede haber un personaje, un bonus o nada. Estos tres tienen en común que ocupan posiciones en el Tablero, por ello se decidió realizar una interfaz, llamada Ubicable, de la cual implementan tres clases: Algoformer, Bonus y Vacio.
La clase Algoformer, podría decirse que es la más importante de estas tres, representa a los distintos personajes del juego. Para empezar, los Algoformers se crean mediante una clase Fabrica de Algoformers, que conoce los distintos tipos de Algoformers (junto con sus estadísticas y particularidades) y devuelve un Algoformer con las características especificadas. La clase Algoformer es una clase abstracta, que tiene por hijos a los Autobot y a los Decepticon. Una particularidad de los Algoformer es que se pueden mover en el tablero, mientras que los otros Ubicables no.
La clase Bonus, representa a todo aquello que ocupa una posición en el tablero pero que no se puede mover y que al ser “activada” por un Algoformer produce un cambio de estado. Para ser activada un Algoformer debe pararse sobre ella. Para ello, se realizo una clase reemplazar que se encarga de realizar la acción esperada según el bonus (la clase Algoformer también la tiene y devuelve una excepción indicando que no puede ocuparse una misma posición por dos Algoformer). Para la representación propuesta se considero a la Chispa Suprema como un caso particular de Bonus.

Por ultimo, en todo lugar del tablero que no haya un Algoformer o un Bonus, habrá Vacio.
Todos los Ubicables presentados, se caracterizan por tener una Posición asociada a cada momento.

Diagramas de clases

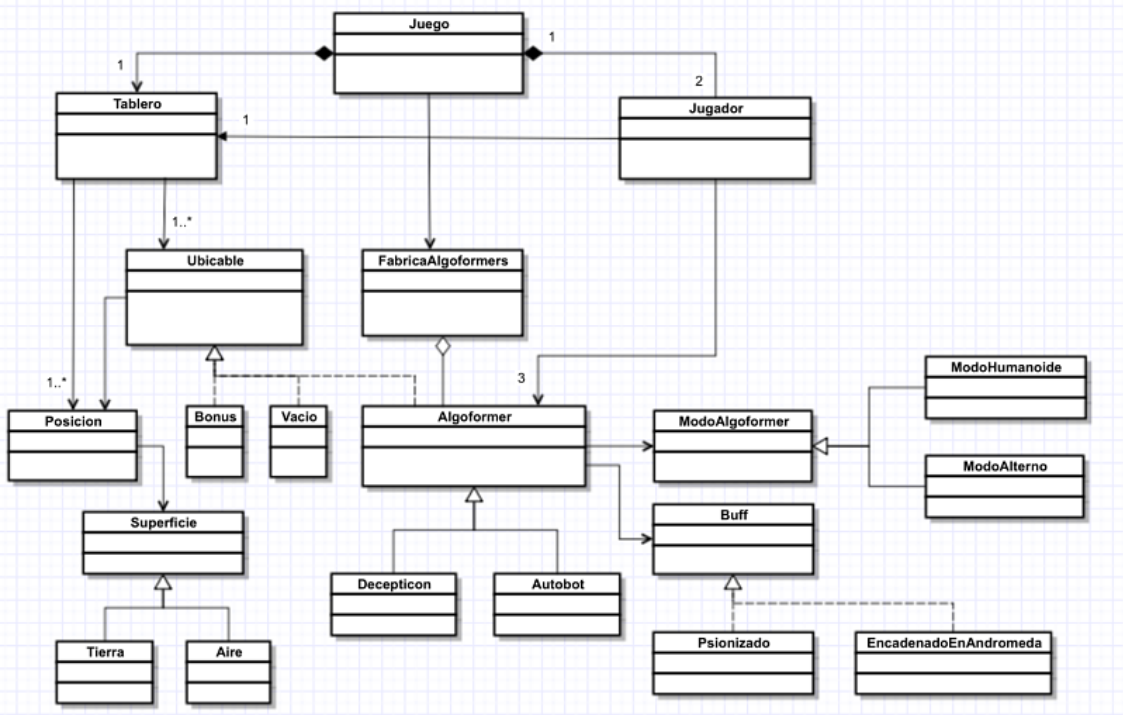

Figura 1: Diagrama de Clases General (sin detallar)

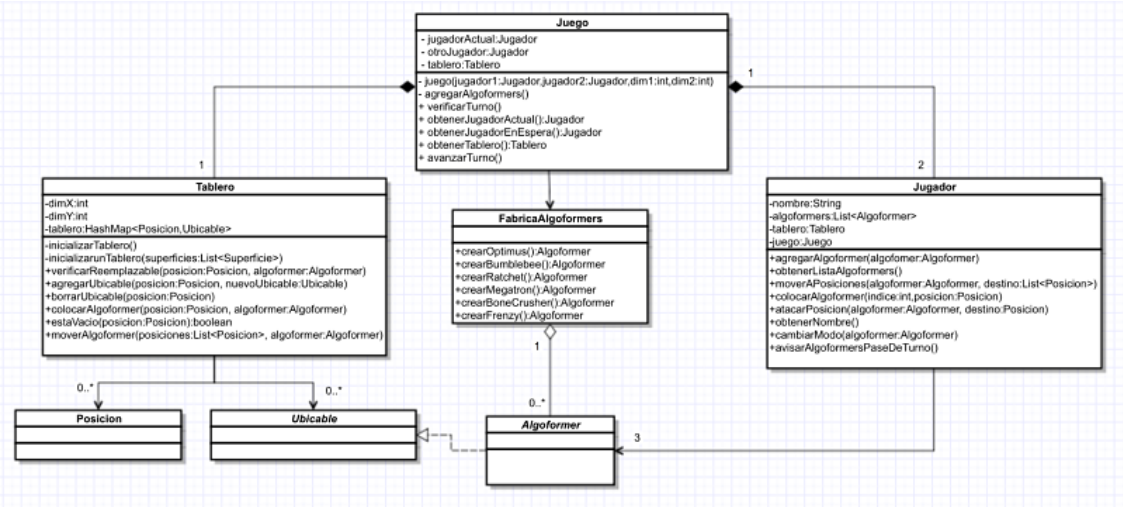

Figura 2: Diagrama de Clase en Detalle del Juego

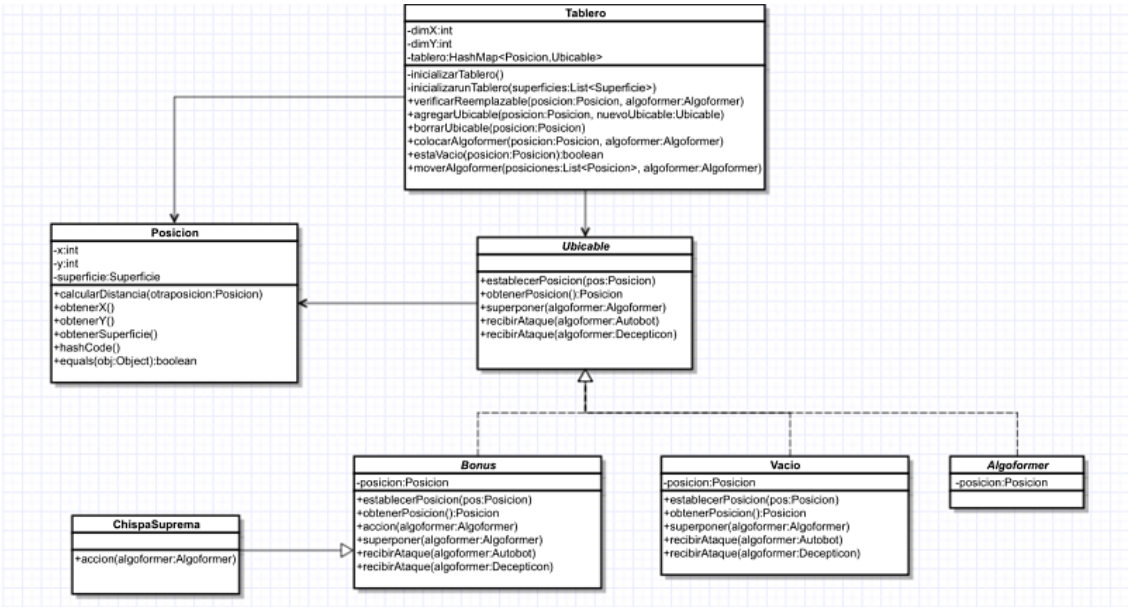

Figura 3: Diagrama de Clases en Detalle del Tablero

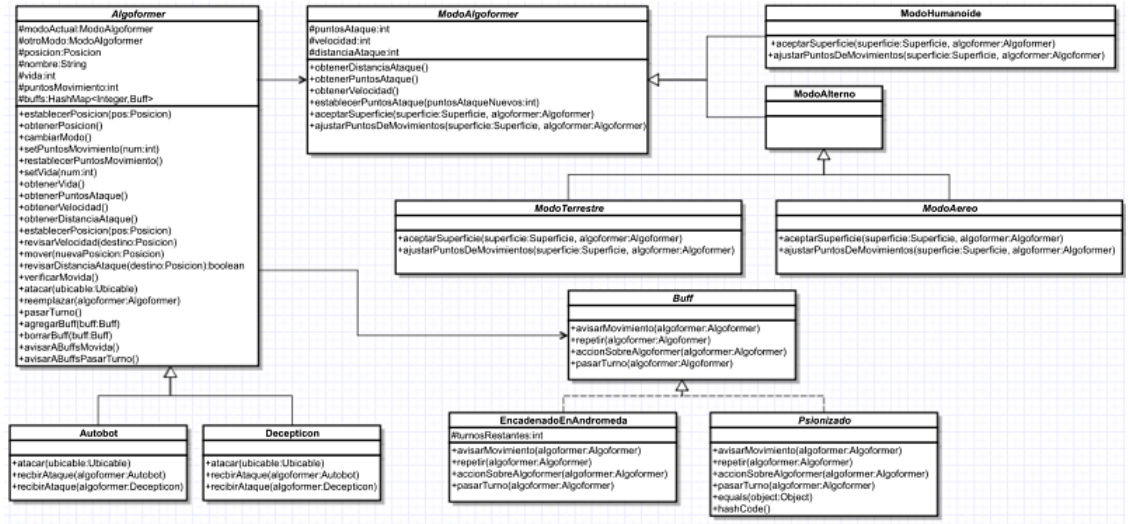

Figura 4: Diagrama de Clases en Detalle de Algoformer

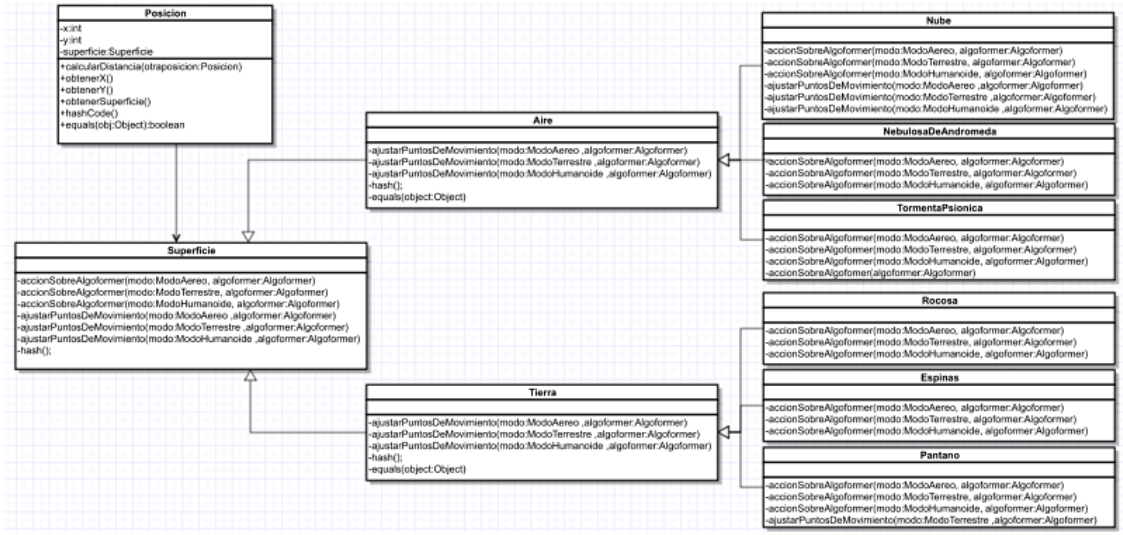

Figura 5: Diagrama de Clases en Detalle de Superficie

Diagramas de secuencia

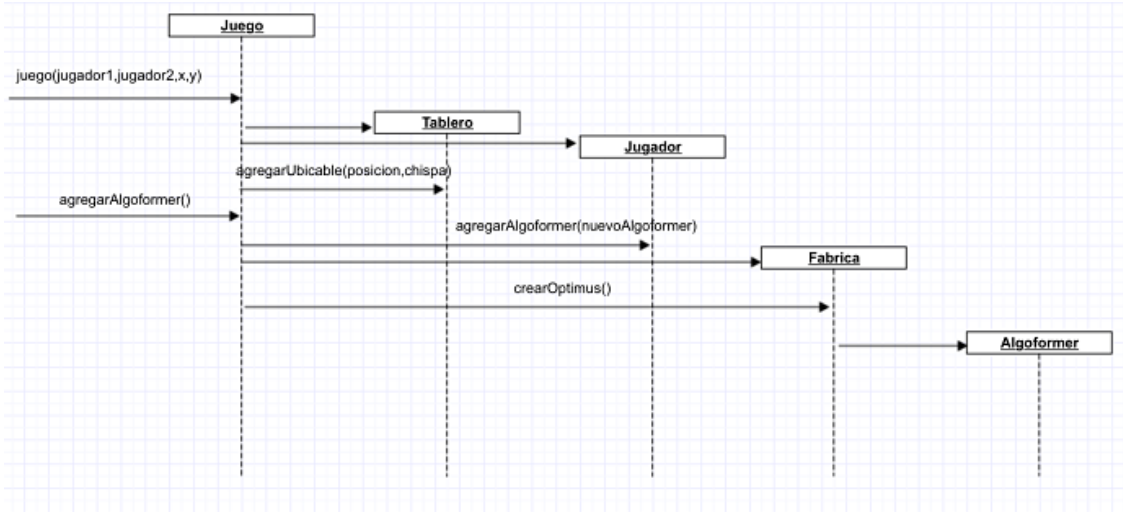

Figura 6: Diagrama de Secuencia de la creacion del Juego

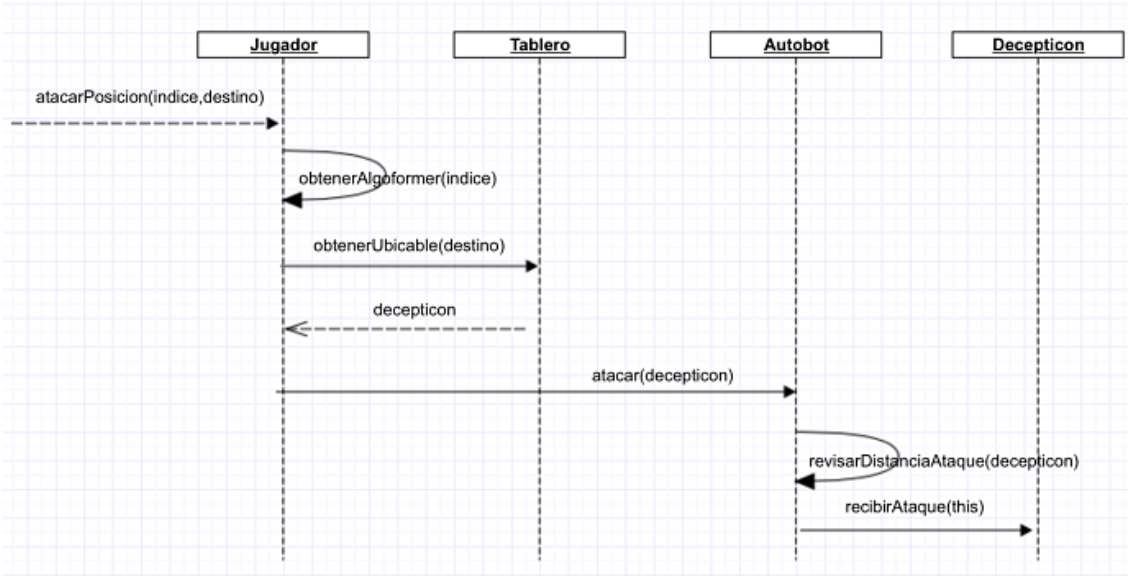

Figura 7: Diagrama de Secuencia del ataque de un Autobot a un Decepticon realizado por el jugador

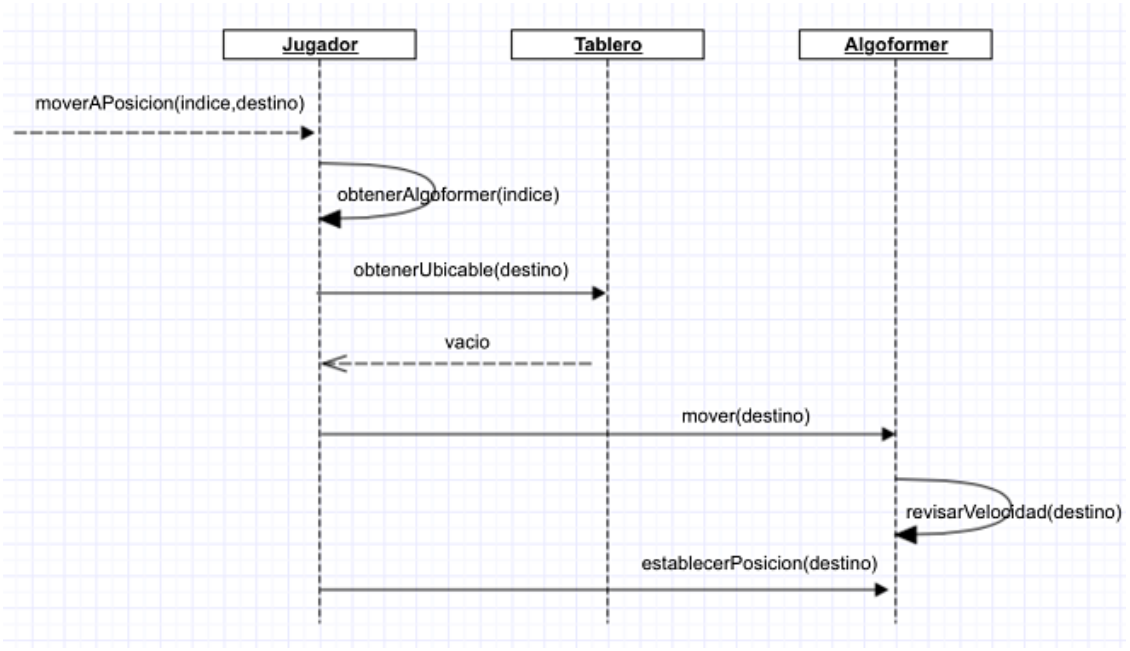

Figura 8: Diagrama de Secuencia del movimiento de un Algoformer en el tablero realizado por el jugador

Diagrama de paquetes

[A completar]

Diagramas de estado

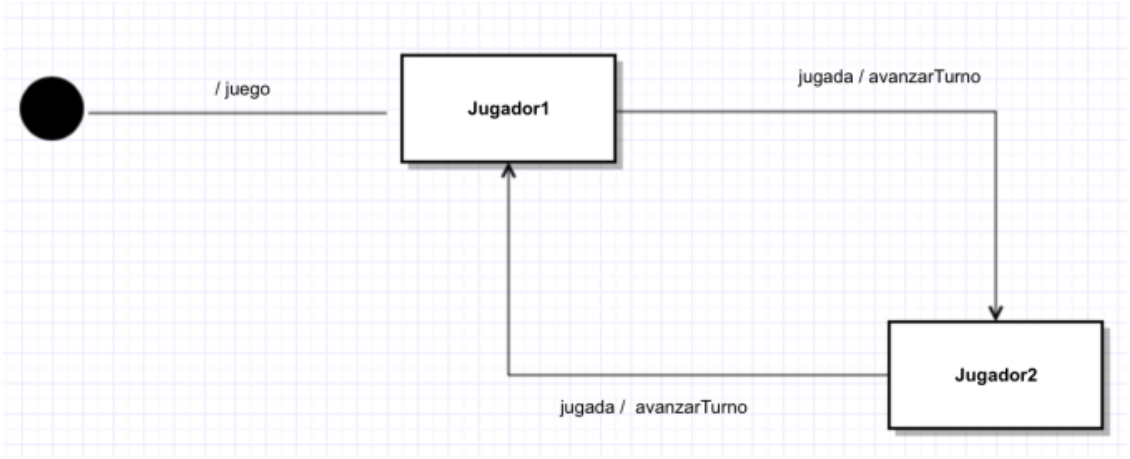

Figura 9: Diagrama de Estado del Turno en el Juego

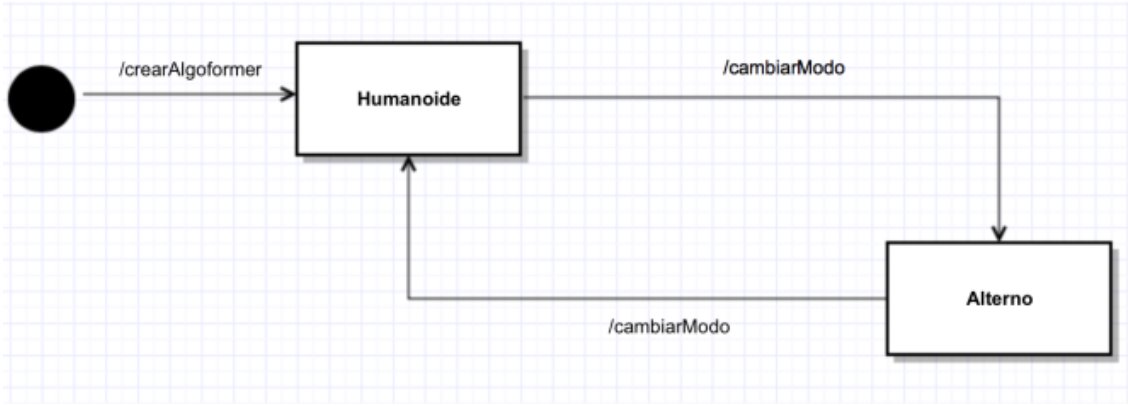

Figura 10: Diagrama de Estado de un Algoformer Generico

Detalles de implementación

Para implementar a los algoformers, se optó por crear una clase abstracta Algoformer que contenga como atributo un objeto perteneciente a la interfaz ModoAlgoformer, esto permite usar el patron strategy y que el comportamiento de la clase se base en el estado actual de ese objeto. De esta forma se evita tener que crear dos clases más que hereden de Algoformer y a su vez que estas hereden en Autobot y Decepticon. Algoformer se desarrollo de tal manera que recibe por parametro todos los atributos que

componen a cada tipo de Algoformer en particular y ahorrando así tener que crear una clase para cada tipo de Algoformer (Optimus, Megatron, etc). Como se mencionó en el apartado "Modelo de Dominio", se creó una clase FabricaAlgoformers que crea cada Algoformer pasandole sus atributos propios.

Por otra parte implementamos una interfaz Ubicable, para poder establecer una Posicion para todos los objetos en el Tablero con mas facilidad y no tener que crear un metodo de posicion para cada uno de los objetos del tablero. De esta manera un Algoformer, un Bonus o un “Vacio” son todos objetos que implementan la interfaz Ubicable por la cual podemos establecer sus posiciones dentro del tablero.

En cuanto a la resolución del manejo de las superficies, se realizó una clase abstracta Superficie, de la cual heredan las clases Tierra y Aire que tambien son abstractas, y a su vez, de cada una de estas heredan los distintos tipos de superficies (Rocosa, Pantano, Espinas y Nube, Nebulosa, Tormenta, respectivamente). Se decidió que las superficies sean contenidas por la posición, de esta manera hay por ejemplo una Posición(1,1,Aire) y una Posición(1,1,Tierra). Se podría pensar que Aire o Tierra son una tercer coordenada de la posición. A la hora de realizar el tablero, para cada par x,y se coloca una posición en la Tierra y otra en el Aire, cada una con un tipo de superficie especifica, por ejemplo, Posición(1,1,Rocosa) y Posición(1,1,Nube). Estas dos se hashean a posiciones distintas del HashMap usado en la implementación. Sin embargo, cuando el Jugador quiera colocar Algoformer en Posición(1,1,Rocosa), intentará acceder a Posición(1,1,Tierra), ya que no es responsabilidad del jugador conocer que tipo de superficie hay en cada posición. Esto se logró realizando algunas modificaciones en los métodos HashCode() y Equals() de Posición.

Respecto a los efectos producidos por cada una de estas superficies, se realizó un Double Dispatch entre cada tipo de Superficie y el ModoAlgoformer. Esto se realizó debido a que el efecto producido por las superficies depende del modo de cada algoformer. Para ello el ModoAlgoformer se implementó como una clase abstracta de la cual heredan ModoAlterno y ModoHumanoide. Por un lado, ModoAlterno se la modeló como una clase abstracta de la cual heredan ModoAereo y ModoTerrestre. Se optó por la herencia en este caso, ya que la diferenciación entre aéreo y terrestre solo existe en ModoAlterno, por lo tanto los ModoAereo y ModoTerrestre son ModoAlterno. Por otro lado, el ModoHumanoide es una clase concreta, ya que no existe ninguna diferenciación entre los algoformers de este modo.

Por ultimo, existen algunas superficies en particular, cuyo efecto persiste durante más de un turno o tienen consecuencias permanentes en el Algoformer. Superficies con estas características son la NebulosDeAndromeda y la TormentaPsiónica. En estos casos se optó por implementar "Buffers" que guardan las posiciones por las que pasó el Algoformer (con sus superficies). De esta manera, el Algoformer puede ir verificando esos Buffers a cada turno.

Excepciones

- **AlgoformerNoExisteException:** si el jugador quiere acceder y utilizar un algoformer no existente de su lista de algoformers, es necesario informarle al jugador por lo tanto se lanza esta excepcion.

- **AtaqueInvalidoException**: si el jugador quiere atacar con su algoformer a otro algoformer de su mismo equipo, o a una posicion vacia, o a un “bonus”, es necesario avisarle que es imposible realizar esta tarea. Por lo que es necesario lanzar esta excepcion.

- **ObjetoMuyLejosException**: si un algoformer desea atacar a una posicion es necesario comprobar si la posicion se encuentra en el rango de ataque del algoformer o en el rango de movimiento del algoformer, de esta manera es necesario avisarle al jugador si la distancia del objeto a atacar es mayor que el rango de ataque del algoformer a utilizar o la distancia de la posicion a mover es mas mayor que el rango de movimiento del algoformer.

- **NoEsSuTurnoException**: cuando en el juego un jugador desea realizar algun movimiento, es necesario preguntar al juego si es el turno correspondiente del jugador. En caso de que no lo sea, se debe lanzar una excepcion.

- **NoSuperponibleException:** cuando un jugador quiere realizar un movimiento de un algoformer, es necesario verificar que en la posicion a donde quiero mover el algoformer no este ocupado por otro algoformer. En caso de que lo este, se debe lanzar una excepcion invalidando el movimiento.

- **FueraDelTableroException:** cuando un jugador quiere realizar un movimiento de un algoformer, es necesario verificar que en la posicion a donde quiero mover el algoformer se encuentre dentro de los limites del tablero. Si no lo esta, es necesario lanzar una excepcion

- **EncadenadoException:** cuando un algoformer se encuentra en la superficie nebulosa de andromeda, es necesario informar que el algoformer no puede moverse por 3 turnos. Mientras no terminen los tres turnos es necesario lanzar una excepcion invalidando el movimiento del algoformer

- **SuperficieNoAtravesableException:** cuando un algoformer quiere movilizarse en una superficie, es necesario informar si el algoformer no puede moverse por el tipo de superficie. Por ejemplo en el pantano mientras el algoformer no cambie su modo a alterno es necesario lanzar una excepcion invalidando el movimiento del algoformer

- **NoSePuedeTransformarException:** cuando un algoformer aereo en modo alterno se encuentra en una posicion Aire no puede transformarse a su modo humanoide, por eso se debe lanzar una excepcion informando que no se pudo lograr el cambio de modo

Checklist de corrección

Esta sección es para uso exclusivo de los docentes, por favor no modificar.

Carpeta

Generalidades

- ¿Son correctos los supuestos y extensiones?
- ¿Es prolija la presentación? (hojas del mismo tamaño, numeradas y con tipografía uniforme)

Modelo

- ¿Está completo?¿Contempla la totalidad del problema?
- ¿Respeta encapsulamiento?
- ¿Hace un buen uso de excepciones?
- ¿Utiliza polimorfismo en las situaciones esperadas?

Diagramas

Diagrama de clases

- ¿Está completo?
- ¿Está bien utilizada la notación?

Diagramas de secuencia

- ¿Está completo?
- ¿Es consistente con el diagrama de clases?
- ¿Está bien utilizada la notación?

Diagrama de estados

- ¿Está completo?
- ¿Está bien utilizada la notación?

Código

Generalidades

- ¿Respeta estándares de codificación?
- ¿Está correctamente documentado?