

FIUBA - 75.07

Algoritmos y programación III

Trabajo práctico 2



1er cuatrimestre, 2016

(trabajo grupal de 4 integrantes)

Alumnos:

| Nombre | Padrón | Mail |
|--------------------------|--------|-----------------------------|
| Capurro, Juan Pablo | 98124 | juanpcapurro@gmail.com |
| González Pérez, Ailén Y. | 97043 | ailengonzalezp@yahoo.com.ar |
| Prado, María Florencia | 96626 | mflorprado@gmail.com |
| Talavera, Carlos | 97872 | talaveracarlosd@gmail.com |

Fecha de entrega final: Jueves 07/07/2016

Tutor: Pablo Suárez

Comentarios:

1. Introduction

1.1. 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).

1.2. 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.

2. Descripción de la aplicación

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.

2.1. 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 |

2. 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 |

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

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

4. 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 |

2.2. Decepticons

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

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

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

| Bonecrusher | Humanoide | Alternativo |
|---------------------|-----------|-------------|
| Ptos de Vida | 200 | |
| Ataque | 30 | 30 |
| Distancia de ataque | 3 | 3 |
| Velocidad | 1 | 8 |

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

| Frenzy | Humanoide | Alternativo |
|---------------------|-----------|-------------|
| Ptos de Vida | 400 | |
| Ataque | 10 | 25 |
| Distancia de ataque | 5 | 2 |
| Velocidad | 2 | 6 |

- 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 |

2.3. 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 | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |

2.4. Superficies

Tierra

- Rocosa: Todas las unidades pueden atravesarla.
- 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.

2.5. 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.

2.5.1. 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.

2.6. 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.

2.7. 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 | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |

2.8. 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 superior y menor también pueden atrapar la chispa suprema. Si un jugador logra destruir a todos los algoformers del jugador contrario, también gana en ese caso.

2.9. [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.

2.10. 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.

3. Entregables

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

3.1. Formas de entrega

Habr  4 entregas formales. Las mismas tendr n una calificaci n de APROBADO o NO APROBADO en el momento de la entrega. Aqu l grupo que acumule 3 no aprobados, quedar  autom ticamente desaprobado con la consiguiente p rdida de regularidad en la materia. En cada entrega se debe traer el informe actualizado.

3.2. Evaluaci n

El d a del vencimiento de cada entrega, cada ayudante convocar  a los integrantes de su grupo, solicitar  el informe correspondiente e iniciar  la correcci n mediante una entrevista grupal. Es imprescindible la presencia de todos los integrantes del grupo el d a de cada correcci n. Se evaluar  el trabajo grupal y a cada integrante en forma individual. El objetivo de esto es comprender la din mica de trabajo del equipo y los roles que ha desempe ado cada integrante del grupo. Para que el alumno apruebe el trabajo pr ctico debe estar aprobado en los dos aspectos: grupal e individual. Dentro de los  tems a chequear el ayudante evaluar  aspectos formales (como ser la forma de presentaci n del informe), aspectos funcionales: que se resuelva el problema planteado y aspectos operativos: que el TP funcione integrado.

3.3. Casos de prueba para cada entrega

3.3.1. 1er Entrega

Turnos, Jugadores, Unidades, Tablero, casillero

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

3.3.2. 2da Entrega

Turnos, Jugadores, Unidades, Tablero, casillero

1ra entrega + Interacci n con superficies

Llenar una zona rocosa, verificar que todos los algoformers en todos sus modos la atraviesen sin problemas Llenar una zona pantano, verificar que en modo humanoide no se pueda atravesar. Llenar una zona pantano, verificar que en modo alterno las unidades terrestres tardan el doble que rocoso Llenar una zona pantano, verificar que las unidades a reas las atraviesan sin problemas Llenar una zona de espinas verificar que todas las unidades terrestres pierden un 5% de sus vida por cada casillero de estos que atraviesen Llenar una zona de espinas, verificar que unidades a reas no tienen problemas al atravesarlas. Llenar una zona con nubes, verificar que las unidades a reas las atraviesan sin problemas Llenar una zona de nebulosa de andr meda, pasar una unidad a rea, corroborar que quede 3 turnos atrapada, sin moverse Llenar una zona de tormenta psi nica, pasar

un algoformer alterno aéreo, ver que baje su capacidad de ataque test 9 + volver a pasar y ver que no bajó su capacidad de ataque.

3.3.3. 3er Entrega

2da entrega + Bonus, Interfaz gráfica inicial

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. Repetir para el modo alterno. Ubico un algoformer, ubico un bonus burbuja, ubico un otro algoformer 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. Realizar el mismo test en modo alterno Ubico un algoformer, ubico un bonus flash, verifico que se mueve 3 veces más rápido durante 3 turnos propios. Repetir en modo alterno Repetir en modo humanoide-alterno-humanoide Test boundary cases (Si ya tiene un bonus de un tipo que no pueda agarrar otro del mismo tipo, Atrapar 2 bonus distintos verificar ambos comportamientos, etc. . .)

3.3.4. 4ta Entrega

Trabajo Práctico completo funcionando, con interfaz gráfica final, sonidoPs e informe completo. Tiempo total de desarrollo del trabajo práctico: 5 semanas completas.

4. Informe

4.1. Cosas a tener en cuenta al jugar

- Con click izquierdo se selecciona el Algoformer; otro click izquierdo en una posición vacía, se mueve hacia allí. Si en vez de una posición vacía se selecciona una posición con otra unidad en ella, el Algoformer se acercara lo más posible a esta unidad.
- Si se hace click derecho en un Algoformer y luego click izquierdo en otro, la primer unidad atacará a la segunda, siempre y cuando esta se encuentre en su rango de ataque permitido.
- En los casilleros lluviosos (correspondientes a la nebulosa), las unidades aéreas quedan atrapadas por tres turnos
- En los casilleros tormentosos, los algoformers pierden capacidad de ataque en un 40% PARA SIEMPRE
- Las unidades alternas terrestres tardan el doble de tiempo en cruzar el pantano.
- Los algoformers humanoides no pueden cruzar el pantano.
- Por una razón de estética, las espinas se convirtieron en volcanes. Por lo tanto, si un algoformer pasa por el volcán se verá afectado como si fueran espinas. Es decir, perderá un 5% de vida en cualquiera de sus modos.
- Los turnos finalizan automáticamente luego de realizar un ataque o completar un movimiento, pero si tu unidad queda atrapada tenés que finalizar el turno por tu cuenta.
- No está permitido el robo de algoformers. Cada jugador puede mover sólo los propios.

4.2. Supuestos

Se indican a continuación las ideas que fueron supuestas para el desarrollo del trabajo, así como las decisiones que merecen ser aclaradas:

- Dos algoformers no pueden ubicarse en la misma posición (casillero), ni siquiera uno en aire y otro en tierra
- Los bonus que duran una x cantidad de turnos, no cuentan en ese x el turno de obtención. El bonus se pone en uso al momento de atraparlo, y el siguiente turno es el primero de los x disponibles.
- Por estética de la interfaz gráfica, se representa a la superficie ESPINAS con imagen de VOLCÁN.
- Se elige un tamaño de tablero de forma que no sea ni de tamaño muy pequeño, que los algoformers se encuentren muy cercanos desde un principio, y que no sea demasiado grande para que los algoformers no tarden demasiados turnos para acercarse.

4.3. Modelo de dominio

Analizando la estructura del problema, generamos una serie de clases que serán parte de la solución, pero que surgen del problema en sí mismo. Para mayor organización y división clara de los archivos, se ordenan en distintos conjuntos (paquetes) de acuerdo a su naturaleza.

En cuanto a responsabilidades que necesitamos asignar, podemos decir que son variadas y numerosas, y el cumplimiento de las mismas requiere estrechas relaciones de clases.

Por ejemplo, el tablero se encarga de contener todos los casilleros del juego; cada uno de estos casilleros, contiene superficies y (puede o no) contener un robot en cualquier modo.

Las superficies a su vez, saben qué daño (o no) generan a quien pasa por ellas, enviando la información necesaria para la aplicación del daño.

El jugador guarda sus robots y es quien decide los movimientos de los mismos; establece qué acción se realizará en cada jugada.

Turno y Partida (con su consiguiente ProxyPartida, utilizado para la parte gráfica), controlan la mecánica del juego y la interacción entre los dos jugadores, así como limita el accionar de un jugador de acuerdo al estado actual del juego y decisiones tomadas previamente.

Cada robot, funciona de cierta manera de acuerdo a su estado. Con su atributo de estado, responderá a los mensajes de distinta forma dependiendo su situación actual. Se generan clases de Estado, que serán las que respondan de una forma u otra a los mensajes. El objetivo es no preguntar por el modo de un algoformer, sino que estos actúen y respondan a mensajes teniendo en cuenta su estado interno.

Para realizar la mantención del estado interno de los colocables (algoFormers) proponemos el uso de modificadores; el turno avisa su finalización a los jugadores, los cuales avisan a sus robots que actualicen su estado conforme a la finalización del turno. A su vez, al final de cada turno, los modificadores saben si deben influir en el estado del algoformer o no, así como si deben ser existiendo o “autodestruirse” porque su tiempo de vida ha expirado.

4.4. Diagramas de clases

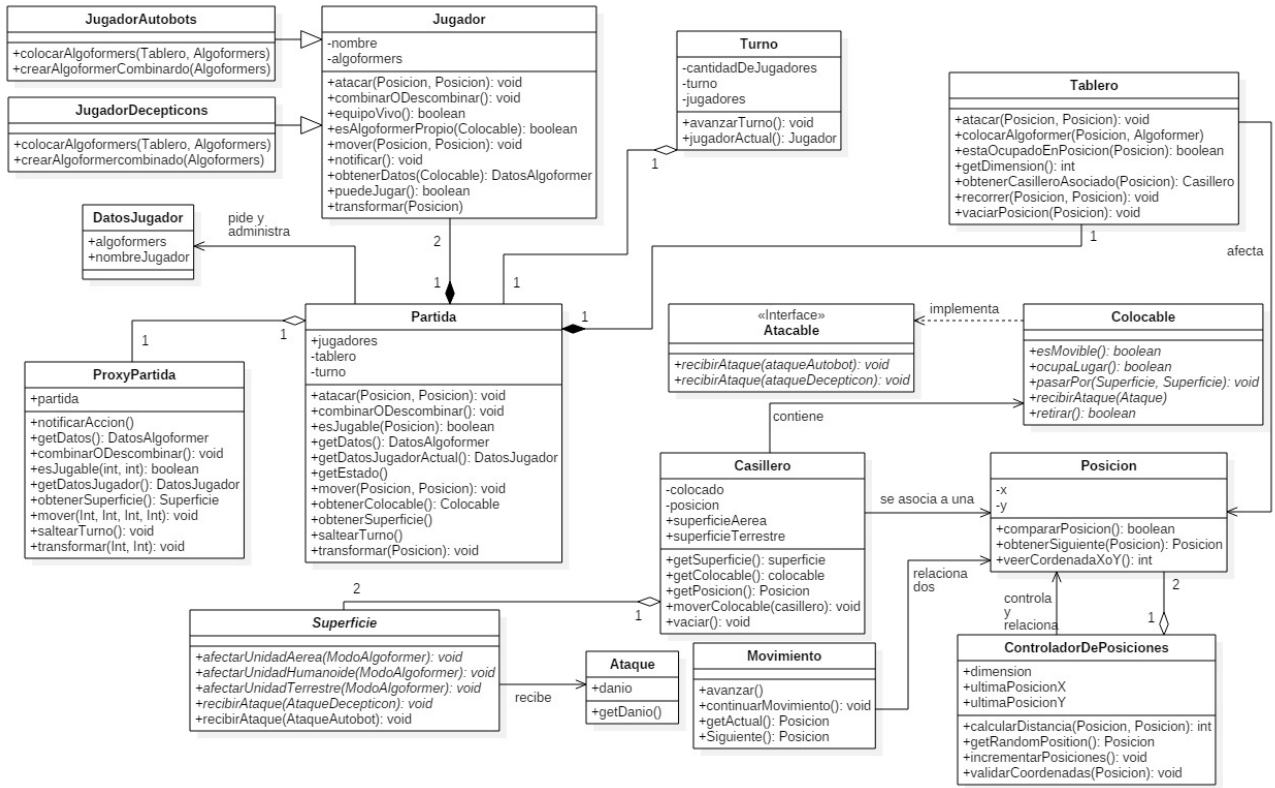


Figura 1: Diagrama de clases principal

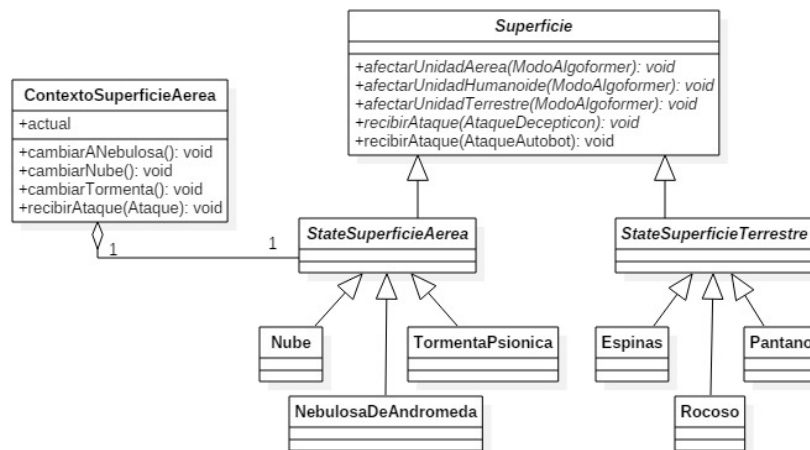


Figura 2: Diagrama de clases de superficies

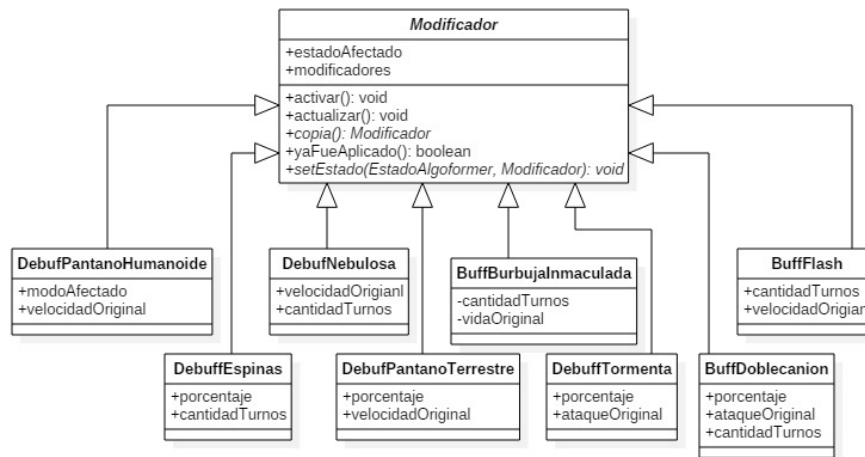


Figura 3: Diagrama de clases de modificadores

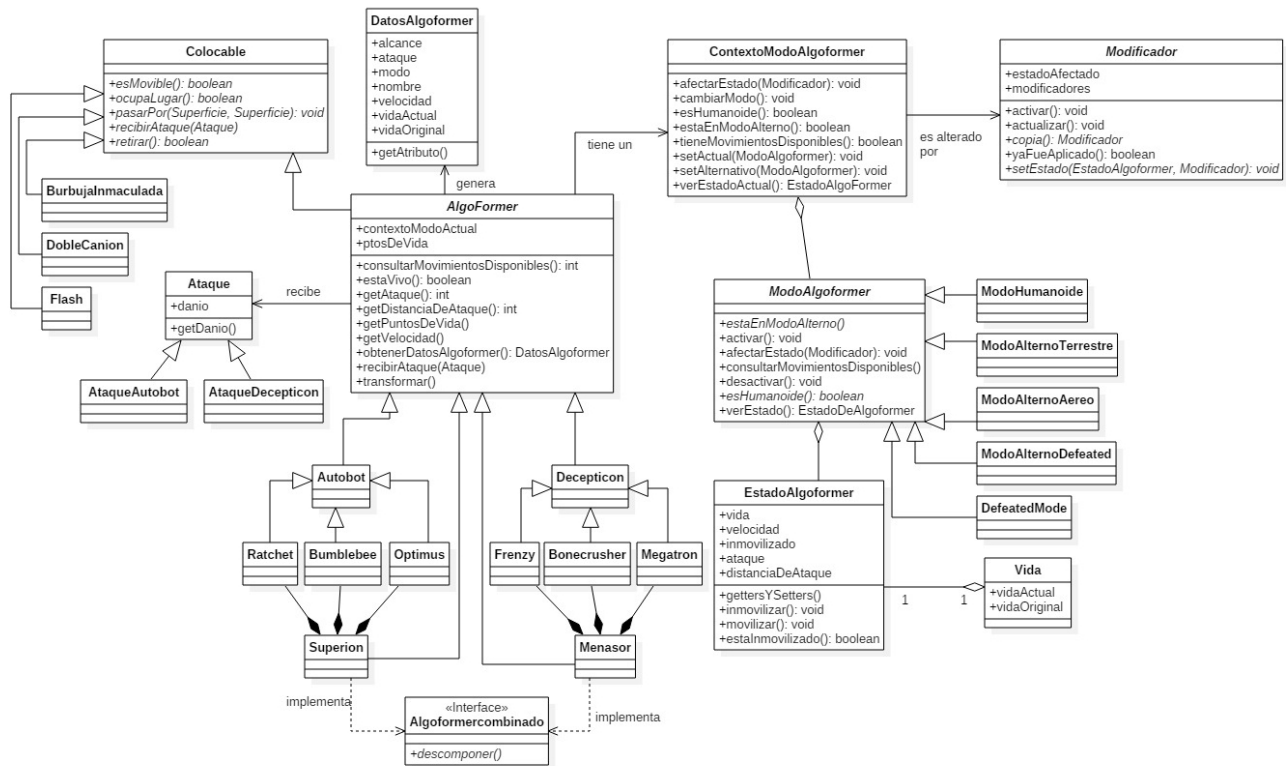


Figura 4: Diagrama de clases de robots

4.5. Diagramas de estado

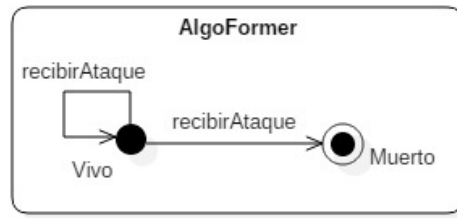


Figura 5: Estados de vida de un algoformer

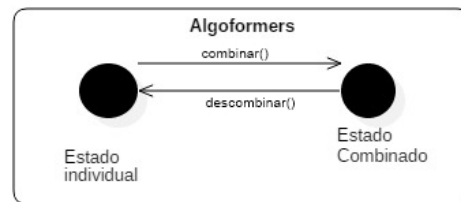
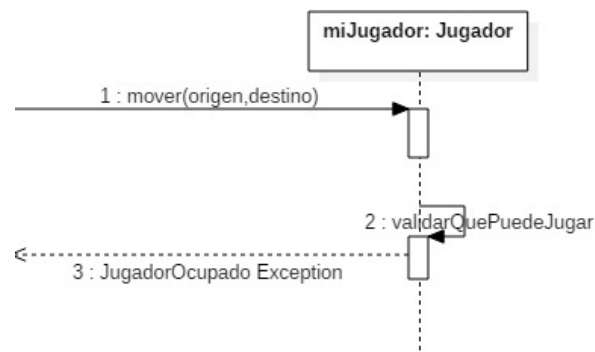


Figura 6: Estados de forma de un algoformer

4.6. Diagramas de secuencia

Se presenta a continuación el diagrama de secuencia correspondiente al método MOVER de la clase JUGADOR. Comienza con el llamado de este método, desde algún "lugar invocante" se lleva a cabo la primer llamada a otro método. Esta llamada, comprueba si el jugador está o no en condiciones de jugar o se encuentra ocupado. Si la respuesta fuera que se encuentra ocupado, se lanza una excepción y encontramos que el diagrama de secuencia es el siguiente.



Si la respuesta es negativa, se procede a preguntar si el algoformer de la posición de origen es propio, o se está cometiendo un hurto de robots. A esta pregunta, puede responderse con una excepción si el delito es un hecho.

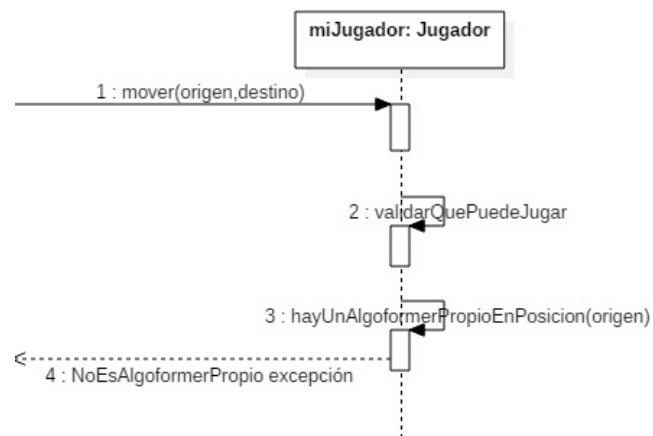
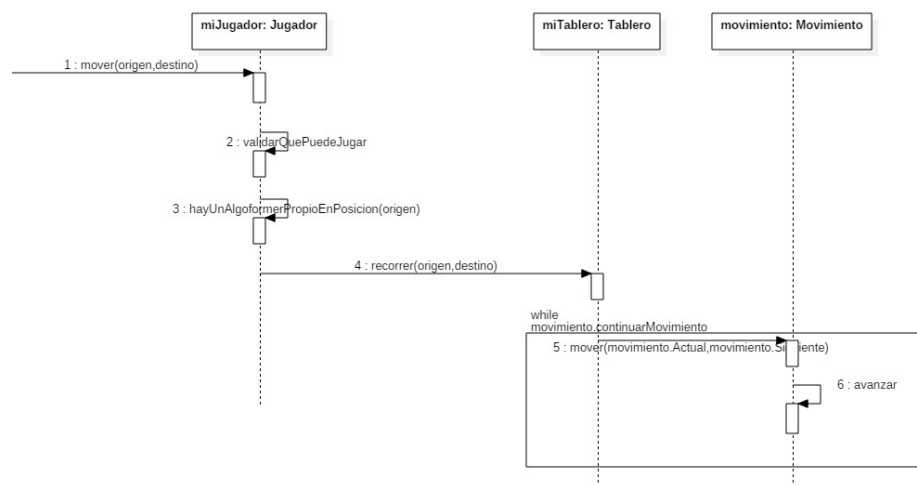
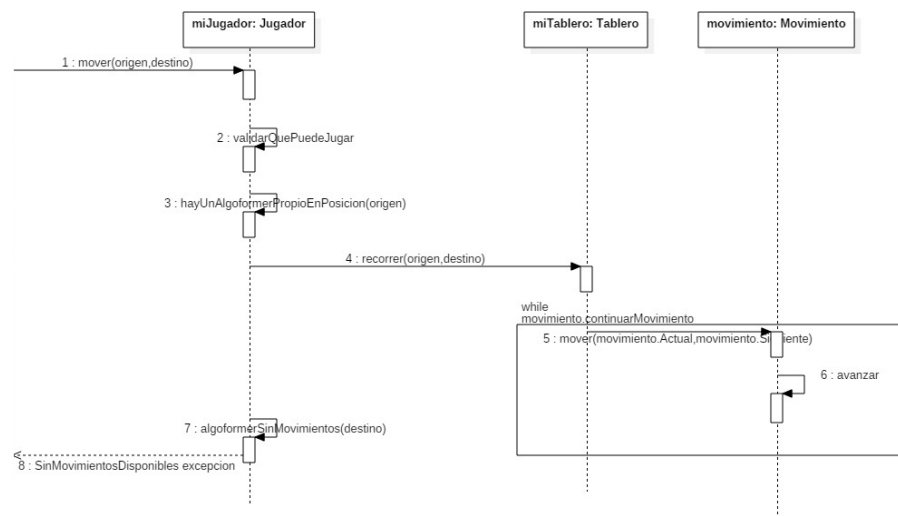


Figura 7:

Si el algoformer elegido es propio, se procede a crear un movimiento y comenzar el recorrido a partir de él, hasta llegar a la posición de origen (idealmente).



Al final, se va a chequear si el jugador llegó a destino o se quedó sin movimientos en el camino. Si se da esta segunda situación, se lanza una excepción.



4.7. Diagramas de paquete

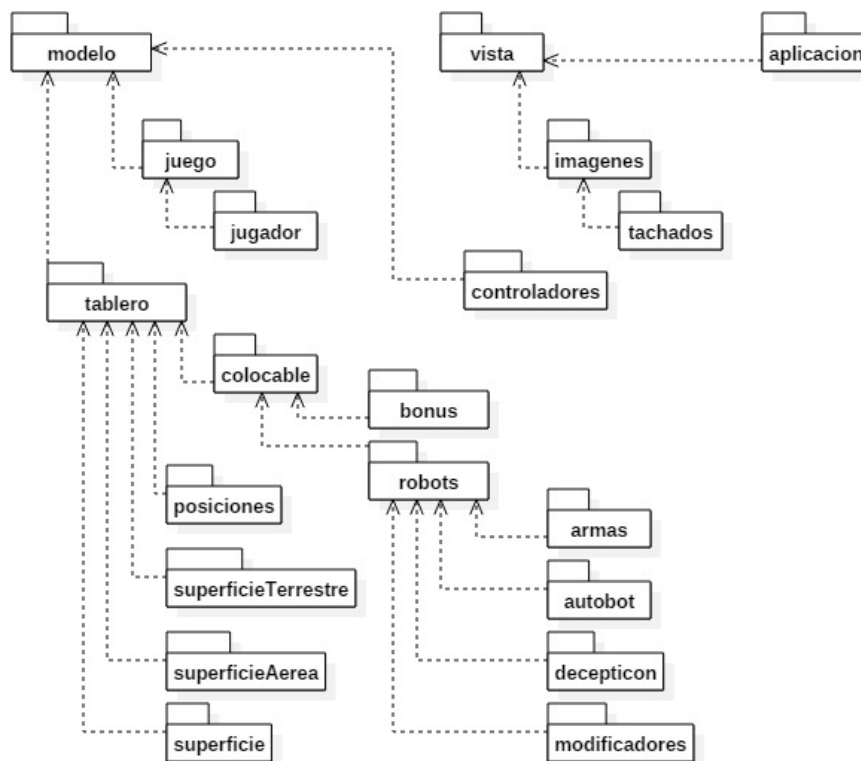


Figura 8: Diagrama de paquetes del proyecto

4.8. Excepciones

Cómo lo indica su nombre las excepciones se generan para tratar situaciones inesperadas. La mayoría de las excepciones, derivan de la clase de Java llamada `Exception`.

Se detalla a continuación en qué contexto se lanzan las excepciones.

- `CasilleroYaOcupado`: Se intenta mover un algoformer a un casillero que ya contiene un robot de forma previa. No depende de si el robot es o no del mismo equipo; no se permite nunca.
- `JuegoFinalizado`: ¡Ha ganado un equipo!
- `NoEsAlgoformerPropio`: Se lanza cuando un jugador intenta mover un robot ajeno.
- `ObjetivoFueraDeRango`: Se lanza cuando se intenta atacar un objetivo fuera del alcance de ataque.
- `SinMovimientosDisponibles`: No se llegó a la posición final porque estaba a mayor cantidad de casilleros que la permitida.
- `NoSePuedeCombinarPorTenerAlgoformersMuertos`: `Superion` y `Menasor` se forman combinando los 3 algoformers del equipo; si alguno está muerto, la combinación no se puede completar.
- `NoSePuedeTransformarPorSerCombinado`: `Superion` y `Menasor` no tienen modos alternos

4.9. Detalles de implementación

Para manejar los cambios que puede sufrir una superficie debido a la recepción de un ataque, implementamos un `State`. Nos permite responder de una manera u otra a los mensajes, de acuerdo al estado actual de la instancia. Las principales diferencias están en que una superficie puede ser aérea o terrestre, y dentro de esta división, se desprenden otras tantas de acuerdo al tipo de superficie. Cada casillero contendrá los dos tipos de superficie para simular cielo y tierra en cada posición.

La misma idea, se aplica al estado de los algoformers; responden de distinta manera si está en modo normal o alternativo, así como también cambia la respuesta si se han combinado los algoformers individuales en el algoformer “superior”. Las características del algoformer en un momento dado, las da su estado actual que es una clase con atributos definidos específicamente. Estos estados, alternativo y humanoide, se aplican sólo a los algoformers “básicos”, no a `Menasor` o `Superion`. A su vez, el estado alternativo, se diferencia entre terrestre y aéreo.

Los modificadores son los encargados de mantener actualizados los estados de los algoformers turno a turno. Cuando un turno termina, este notifica a los jugadores; de esta manera, los jugadores pueden avisar a sus algoformers de esta situación. Los modificadores son implementados como `buff/debuff` de acuerdo a si incrementan o decrementan un cierto valor del estado del algoformer.

Los bonus, que duran más de un turno, en sus modificadores controlan la cantidad de turnos que pasan y cuando se cumple la cantidad correspondiente, se autoelimina de la lista de modificadores. Además, puede distinguir si fue activado o no, de forma que no sean acumulables los efectos de dos bonus seguidos.

4.10. 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?
- ¿Respeto 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?
- ¿Está bien utilizada la notación?
- ¿Es consistente con el diagrama de clases?

Diagrama de estados

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

Código Generalidades

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