

IA para el Desarrollo de Videojuegos

Proyecto final:

Real Time Wargame.

Autores:

Antonio López Martínez-Carrasco

José María Sánchez Salas

Profesores:

Francisco Javier Marín-Blazquez Gómez

Luis Daniel Hernández Molinero

Introducción

- Software y Hardware utilizado.

- Software y Hardware utilizado.
 - Librería LibGDX (implementada en Java).

- Software y Hardware utilizado.
 - Librería LibGDX (implementada en Java).
 - IDE Eclipse.

- Software y Hardware utilizado.
 - Librería LibGDX (implementada en Java).
 - IDE Eclipse.
 - Ningún hardware específico.

Modelo

- WorldObject.

- WorldObject.
- Obstacle.

- WorldObject.
- Obstacle.
- Character.

- WorldObject.
- Obstacle.
- Character.
- Formation:

- WorldObject.
- Obstacle.
- Character.
- Formation:
 - CircularFormation.

- WorldObject.
- Obstacle.
- Character.
- Formation:
 - CircularFormation.
 - LineFormation.

- WorldObject.
- Obstacle.
- Character.
- Formation:
 - CircularFormation.
 - LineFormation.
 - StarFormation.

Steerings

- Interfaz Steering.

- Interfaz Steering.
 - Para tratar a todos los tipos de Steerings de manera homogénea.

- Interfaz Steering.
 - Para tratar a todos los tipos de Steerings de manera homogénea.
- Steerings no acelerados.

- Interfaz Steering.
 - Para tratar a todos los tipos de Steerings de manera homogénea.
- Steerings no acelerados.
 - Velocidad lineal y velocidad angular.

- Interfaz Steering.
 - Para tratar a todos los tipos de Steerings de manera homogénea.
- Steerings no acelerados.
 - Velocidad lineal y velocidad angular.
- Steerings acelerados.

- Interfaz Steering.
 - Para tratar a todos los tipos de Steerings de manera homogénea.
- Steerings no acelerados.
 - Velocidad lineal y velocidad angular.
- Steerings acelerados.
 - Aceleración lineal y aceleración angular.

Comportamientos

- Interfaz Behaviour.

Comportamientos

- Interfaz Behaviour.
 - Para tratar a todos los tipos de comportamientos de manera homogénea.

Comportamientos

- Interfaz Behaviour.
 - Para tratar a todos los tipos de comportamientos de manera homogénea.
- Behaviours no acelerados.

Comportamientos

- Interfaz Behaviour.
 - Para tratar a todos los tipos de comportamientos de manera homogénea.
- Behaviours no acelerados.
 - Seek

Comportamientos

- Interfaz Behaviour.
 - Para tratar a todos los tipos de comportamientos de manera homogénea.
- Behaviours no acelerados.
 - Seek
 - Flee

Comportamientos

- Interfaz Behaviour.
 - Para tratar a todos los tipos de comportamientos de manera homogénea.
- Behaviours no acelerados.
 - Seek
 - Flee
 - Arrive

Comportamientos

- Interfaz Behaviour.
 - Para tratar a todos los tipos de comportamientos de manera homogénea.
- Behaviours no acelerados.
 - Seek
 - Flee
 - Arrive
 - Wander

Comportamientos

- Interfaz Behaviour.
 - Para tratar a todos los tipos de comportamientos de manera homogénea.
- Behaviours no acelerados.
 - Seek
 - Flee
 - Arrive
 - Wander
- Behaviours acelerados.

Comportamientos

- Interfaz Behaviour.
 - Para tratar a todos los tipos de comportamientos de manera homogénea.
- Behaviours no acelerados.
 - Seek
 - Flee
 - Arrive
 - Wander
- Behaviours acelerados.
 - Align

Comportamientos

- Interfaz Behaviour.
 - Para tratar a todos los tipos de comportamientos de manera homogénea.
- Behaviours no acelerados.
 - Seek
 - Flee
 - Arrive
 - Wander
- Behaviours acelerados.
 - Align
 - Anti-Align

Comportamientos

- Interfaz Behaviour.
 - Para tratar a todos los tipos de comportamientos de manera homogénea.
- Behaviours no acelerados.
 - Seek
 - Flee
 - Arrive
 - Wander
- Behaviours acelerados.
 - Align
 - Anti-Align
 - Arrive con un radio

Comportamientos

- Interfaz Behaviour.
 - Para tratar a todos los tipos de comportamientos de manera homogénea.
- Behaviours no acelerados.
 - Seek
 - Flee
 - Arrive
 - Wander
- Behaviours acelerados.
 - Align
 - Anti-Align
 - Arrive con un radio
 - Arrive

Comportamientos

- Interfaz Behaviour.
 - Para tratar a todos los tipos de comportamientos de manera homogénea.
- Behaviours no acelerados.
 - Seek
 - Flee
 - Arrive
 - Wander
- Behaviours acelerados.
 - Align
 - Anti-Align
 - Arrive con un radio
 - Arrive
 - Flee

Comportamientos

- Interfaz Behaviour.
 - Para tratar a todos los tipos de comportamientos de manera homogénea.
- Behaviours no acelerados.
 - Seek
 - Flee
 - Arrive
 - Wander
- Behaviours acelerados.
 - Align
 - Anti-Align
 - Arrive con un radio
 - Arrive
 - Flee
 - Seek

Comportamientos

- Interfaz Behaviour.
 - Para tratar a todos los tipos de comportamientos de manera homogénea.
- Behaviours no acelerados.
 - Seek
 - Flee
 - Arrive
 - Wander
- Behaviours acelerados.
 - Align
 - Anti-Align
 - Arrive con un radio
 - Arrive
 - Flee
 - Seek
 - Velocity Matching

- Behaviours delegados.

- Behaviours delegados.
 - CollisionAvoidance

- Behaviours delegados.
 - CollisionAvoidance
 - Evade

- Behaviours delegados.
 - CollisionAvoidance
 - Evade
 - Face

- Behaviours delegados.
 - CollisionAvoidance
 - Evade
 - Face
 - Looking Where You Going

- Behaviours delegados.
 - CollisionAvoidance
 - Evade
 - Face
 - Looking Where You Going
 - PathFollowing con Arrive

- Behaviours delegados.
 - CollisionAvoidance
 - Evade
 - Face
 - Looking Where You Going
 - PathFollowing con Arrive
 - PathFollowing con Seek

- Behaviours delegados.
 - CollisionAvoidance
 - Evade
 - Face
 - Looking Where You Going
 - PathFollowing con Arrive
 - PathFollowing con Seek
 - Persue

- Behaviours delegados.
 - CollisionAvoidance
 - Evade
 - Face
 - Looking Where You Going
 - PathFollowing con Arrive
 - PathFollowing con Seek
 - Persue
 - WallAvoidance

- Behaviours delegados.
 - CollisionAvoidance
 - Evade
 - Face
 - Looking Where You Going
 - PathFollowing con Arrive
 - PathFollowing con Seek
 - Persue
 - WallAvoidance
 - Wander

- Behaviours delegados.
 - CollisionAvoidance
 - Evade
 - Face
 - Looking Where You Going
 - PathFollowing con Arrive
 - PathFollowing con Seek
 - Persue
 - WallAvoidance
 - Wander
- Behaviours en grupo.

- Behaviours delegados.
 - CollisionAvoidance
 - Evade
 - Face
 - Looking Where You Going
 - PathFollowing con Arrive
 - PathFollowing con Seek
 - Persue
 - WallAvoidance
 - Wander
- Behaviours en grupo.
 - Cohesion

- Behaviours delegados.
 - CollisionAvoidance
 - Evade
 - Face
 - Looking Where You Going
 - PathFollowing con Arrive
 - PathFollowing con Seek
 - Persue
 - WallAvoidance
 - Wander
- Behaviours en grupo.
 - Cohesion
 - Separation

Árbitros

- Interfaz Arbitrator.

- Interfaz Arbitrator.
- Lista de comportamientos de los personajes.

- Interfaz Arbitrator.
- Lista de comportamientos de los personajes.
 - Mapa con clave Float y valor Behaviour.

- Interfaz Arbitrator.
- Lista de comportamientos de los personajes.
 - Mapa con clave Float y valor Behaviour.
- Árbitro por prioridad.

- Interfaz Arbitrator.
- Lista de comportamientos de los personajes.
 - Mapa con clave Float y valor Behaviour.
- Árbitro por prioridad.
 - Para comportamientos acelerados y no acelerados.

- Interfaz Arbitrator.
- Lista de comportamientos de los personajes.
 - Mapa con clave Float y valor Behaviour.
- Árbitro por prioridad.
 - Para comportamientos acelerados y no acelerados.
- Árbitro por mezcla. Comportamientos acelerados.

- Interfaz Arbitrator.
- Lista de comportamientos de los personajes.
 - Mapa con clave Float y valor Behaviour.
- Árbitro por prioridad.
 - Para comportamientos acelerados y no acelerados.
- Árbitro por mezcla. Comportamientos acelerados.
- Árbitro por mezcla. Comportamientos no acelerados.

Interacción parte Reactiva

- Método `applyBehaviour()`.

- Método `applyBehaviour()`.
- Método `applySteering(steering)`.

- Método `applyBehaviour()`.
- Método `applySteering(steering)`.
- Método `update(steering, time)`.

PathFinding

- PathFinding Continuo.

- PathFinding Continuo.
 - Clase PathFinding.

- PathFinding Continuo.
 - Clase PathFinding.
 - Algoritmo LRTA*.

- PathFinding Continuo.
 - Clase PathFinding.
 - Algoritmo LRTA*.
- PathFinding Punto-A-Punto.

- PathFinding Continuo.
 - Clase PathFinding.
 - Algoritmo LRTA*.
- PathFinding Punto-A-Punto.
 - Clase PathFinding.

- PathFinding Continuo.
 - Clase PathFinding.
 - Algoritmo LRTA*.
- PathFinding Punto-A-Punto.
 - Clase PathFinding.
 - Algoritmo LRTA*.

Modificaciones del modelo para la parte Táctica

- Nuevos atributos.

- Nuevos atributos.
- Nuevos métodos.

- Nuevos atributos.
- Nuevos métodos.
 - `initializeTacticalRole(role).`

- Nuevos atributos.
- Nuevos métodos.
 - `initializeTacticalRole(role).`
 - `updateTacticalRole().`

- Equipos del juego.

- Equipos del juego.
 - FJAVIER.

- Equipos del juego.
 - FJAVIER.
 - LDANIEL.

- Equipos del juego.
 - FJAVIER.
 - LDANIEL.
 - NEUTRAL.

- Equipos del juego.
 - FJAVIER.
 - LDANIEL.
 - NEUTRAL.
- Método `getEnemyTeam()`.

Roles Tácticos

- Interfaz TacticalRole.

- Interfaz TacticalRole.
- Clases abstractas Archer y Soldier.

- Interfaz TacticalRole.
- Clases abstractas Archer y Soldier.
- Roles defensivos.

- Interfaz TacticalRole.
- Clases abstractas Archer y Soldier.
- Roles defensivos.
 - Soldado y arquero.

- Interfaz TacticalRole.
- Clases abstractas Archer y Soldier.
- Roles defensivos.
 - Soldado y arquero.
- Roles ofensivos.

- Interfaz TacticalRole.
- Clases abstractas Archer y Soldier.
- Roles defensivos.
 - Soldado y arquero.
- Roles ofensivos.
 - Soldado y arquero.

- Métodos.

- Métodos.
 - `getVelocityFactor(ground)` (es usado en la clase `Character`).

- Métodos.
 - `getVelocityFactor(ground)` (es usado en la clase Character).
 - `getTacticalCost(ground)`.

- Métodos.
 - `getVelocityFactor(ground)` (es usado en la clase `Character`).
 - `getTacticalCost(ground)`.
 - `getMaxDistanceOfAttack()`.

- Métodos.
 - `getVelocityFactor(ground)` (es usado en la clase `Character`).
 - `getTacticalCost(ground)`.
 - `getMaxDistanceOfAttack()`.
 - `getDamageToDone()`.

- Métodos.
 - `getVelocityFactor(ground)` (es usado en la clase `Character`).
 - `getTacticalCost(ground)`.
 - `getMaxDistanceOfAttack()`.
 - `getDamageToDone()`.
 - `getMaxSpeed()`.

- Métodos.
 - `getVelocityFactor(ground)` (es usado en la clase `Character`).
 - `getTacticalCost(ground)`.
 - `getMaxDistanceOfAttack()`.
 - `getDamageToDone()`.
 - `getMaxSpeed()`.
 - `initialize(character)`.

- Métodos.
 - `getVelocityFactor(ground)` (es usado en la clase `Character`).
 - `getTacticalCost(ground)`.
 - `getMaxDistanceOfAttack()`.
 - `getDamageToDone()`.
 - `getMaxSpeed()`.
 - `initialize(character)`.
 - `update(character)`.

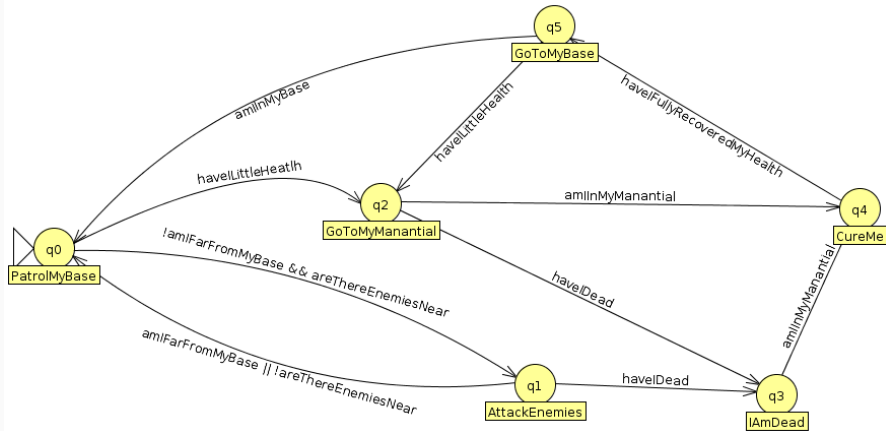
- Métodos.
 - `getVelocityFactor(ground)` (es usado en la clase `Character`).
 - `getTacticalCost(ground)`.
 - `getMaxDistanceOfAttack()`.
 - `getDamageToDone()`.
 - `getMaxSpeed()`.
 - `initialize(character)`.
 - `update(character)`.
 - Estos métodos son implementados en las clases abstractas `Archer` y `Soldier`.

- Métodos.
 - `getVelocityFactor(ground)` (es usado en la clase `Character`).
 - `getTacticalCost(ground)`.
 - `getMaxDistanceOfAttack()`.
 - `getDamageToDone()`.
 - `getMaxSpeed()`.
 - `initialize(character)`.
 - `update(character)`.
 - Estos métodos son implementados en las clases abstractas `Archer` y `Soldier`.
- Atributo `health_cure`.

- Métodos.
 - `getVelocityFactor(ground)` (es usado en la clase `Character`).
 - `getTacticalCost(ground)`.
 - `getMaxDistanceOfAttack()`.
 - `getDamageToDone()`.
 - `getMaxSpeed()`.
 - `initialize(character)`.
 - `update(character)`.
 - Estos métodos son implementados en las clases abstractas `Archer` y `Soldier`.
- Atributo `health_cure`.
 - Todos se curan al mismo ritmo, independientemente de su rol.

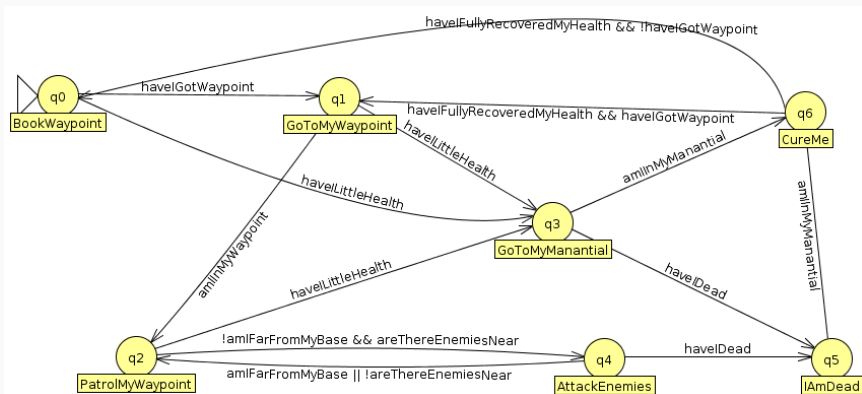
Roles Defensivos

- Soldado.



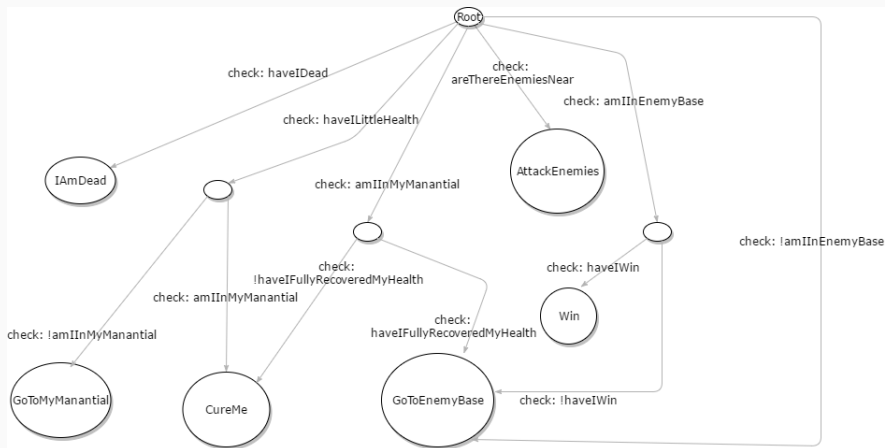
Roles Defensivos

- Arquero.



Roles Ofensivos

- Soldado y arquero.



Otros comportamientos

- Ataque.

- Ataque.
- Cura.

- Ataque.
- Cura.
- Se han implementado como Behaviours para tratarlos de manera homogénea.

- Ataque.
- Cura.
- Se han implementado como Behaviours para tratarlos de manera homogénea.
- Ataque y Cura en las formaciones.

Acciones y comprobaciones

- Actions.

- Actions.
- Checks.

- Actions.
- Checks.
- Útiles para la abstracción de la parte reactiva.

- Actions.
- Checks.
- Útiles para la abstracción de la parte reactiva.
- Sirven de puente entre la parte reactiva y la parte táctica.

Waypoints

- Waypoints de las bases.

Waypoints

- Waypoints de las bases.
- Waypoints de los puentes.

Waypoints

- Waypoints de las bases.
- Waypoints de los puentes.
 - Cada equipo tiene 6 waypoints.

Waypoints

- Waypoints de las bases.
- Waypoints de los puentes.
 - Cada equipo tiene 6 waypoints.
 - En cada lado de los puentes, hay 2 waypoints.

- Waypoints de las bases.
- Waypoints de los puentes.
 - Cada equipo tiene 6 waypoints.
 - En cada lado de los puentes, hay 2 waypoints.
 - Sistema de reserva y liberación de waypoints.

Puntos de Moral

- Cada base tiene una puntuación de moral.

- Cada base tiene una puntuación de moral.
- La moral de una base sube cuando hay personajes de un equipo en su base y no hay personajes del equipo contrario.

- Cada base tiene una puntuación de moral.
- La moral de una base sube cuando hay personajes de un equipo en su base y no hay personajes del equipo contrario.
- La moral de una base baja cuando la base no está protegida y hay personajes del equipo contrario.

- Cada base tiene una puntuación de moral.
- La moral de una base sube cuando hay personajes de un equipo en su base y no hay personajes del equipo contrario.
- La moral de una base baja cuando la base no está protegida y hay personajes del equipo contrario.
- El equipo ganador es el que consigue reducir los puntos de moral de la base contraria a 0.

Mapas de Influencia

- El mapa de influencia refleja la presencia/poder de un equipo en una zona.

Mapas de Influencia

- El mapa de influencia refleja la presencia/poder de un equipo en una zona.
- Cada equipo tiene su propio mapa/matriz de influencia.

Mapas de Influencia

- El mapa de influencia refleja la presencia/poder de un equipo en una zona.
- Cada equipo tiene su propio mapa/matriz de influencia.
- El equipo con mayor influencia en una casilla, domina dicha casilla.

Mapas de Influencia

- El mapa de influencia refleja la presencia/poder de un equipo en una zona.
- Cada equipo tiene su propio mapa/matriz de influencia.
- El equipo con mayor influencia en una casilla, domina dicha casilla.
- La influencia se va reduciendo con la distancia.

Mapas de Influencia

- El mapa de influencia refleja la presencia/poder de un equipo en una zona.
- Cada equipo tiene su propio mapa/matriz de influencia.
- El equipo con mayor influencia en una casilla, domina dicha casilla.
- La influencia se va reduciendo con la distancia.
- Para el cálculo de la influencia se ha utilizado la distancia de Chebyshev (con algunas modificaciones).

- El mapa de influencia refleja la presencia/poder de un equipo en una zona.
- Cada equipo tiene su propio mapa/matriz de influencia.
- El equipo con mayor influencia en una casilla, domina dicha casilla.
- La influencia se va reduciendo con la distancia.
- Para el cálculo de la influencia se ha utilizado la distancia de Chebyshev (con algunas modificaciones).
- Al dibujar el mapa se tienen en cuenta ambas matrices de influencia.

Mapas de Influencia



PathFinding Táctico

- Hace uso de los mapas de influencia.

- Hace uso de los mapas de influencia.
- También puede hacer uso de la información táctica (`getTacticalCost(ground)`) de cada rol.

- Hace uso de los mapas de influencia.
- También puede hacer uso de la información táctica (`getTacticalCost(ground)`) de cada rol.
- Flag para que se pueda ir activando y desactivando sobre la marcha.

- Hace uso de los mapas de influencia.
- También puede hacer uso de la información táctica (`getTacticalCost(ground)`) de cada rol.
- Flag para que se pueda ir activando y desactivando sobre la marcha.
- Video.

Flocking

- Se utiliza un árbitro por mezcla ponderada con los siguientes comportamientos:

- Se utiliza un árbitro por mezcla ponderada con los siguientes comportamientos:
 - CollisionAvoidance.

- Se utiliza un árbitro por mezcla ponderada con los siguientes comportamientos:
 - CollisionAvoidance.
 - Separation.

- Se utiliza un árbitro por mezcla ponderada con los siguientes comportamientos:
 - CollisionAvoidance.
 - Separation.
 - Cohesion.

- Se utiliza un árbitro por mezcla ponderada con los siguientes comportamientos:
 - CollisionAvoidance.
 - Separation.
 - Cohesion.
 - VelocityMatching.

- Se utiliza un árbitro por mezcla ponderada con los siguientes comportamientos:
 - CollisionAvoidance.
 - Separation.
 - Cohesion.
 - VelocityMatching.
 - LookingWhereYouGoing.

- Se utiliza un árbitro por mezcla ponderada con los siguientes comportamientos:
 - CollisionAvoidance.
 - Separation.
 - Cohesion.
 - VelocityMatching.
 - LookingWhereYouGoing.
 - Wander.

- Se utiliza un árbitro por mezcla ponderada con los siguientes comportamientos:
 - CollisionAvoidance.
 - Separation.
 - Cohesion.
 - VelocityMatching.
 - LookingWhereYouGoing.
 - Wander.
- Video

Interacción con el usuario

- Selección de personajes del mismo equipo.

- Selección de personajes del mismo equipo.
- Realizar ciertos comportamientos.

- Selección de personajes del mismo equipo.
- Realizar ciertos comportamientos.
- Implementado siguiendo una máquina de estados.

- Selección de personajes del mismo equipo.
- Realizar ciertos comportamientos.
- Implementado siguiendo una máquina de estados.
- La terminal muestra la información sobre lo que se puede hacer (y cómo).

Elementos opcionales

- Todos los comportamientos implementados.

- Todos los comportamientos implementados.
- Modo debug (comportamientos, personajes, información táctica...).

- Todos los comportamientos implementados.
- Modo debug (comportamientos, personajes, información táctica...).
- Formación en línea y en estrella.

- Todos los comportamientos implementados.
- Modo debug (comportamientos, personajes, información táctica...).
- Formación en línea y en estrella.
- Formación de formaciones hasta el infinito y más allá.

- Todos los comportamientos implementados.
- Modo debug (comportamientos, personajes, información táctica...).
- Formación en línea y en estrella.
- Formación de formaciones hasta el infinito y más allá.
- PathFinding táctico: mapa de influencia y costes tácticos del terreno.

Demostración del juego completo

- Video.

Gracias por su atención.

Ruegos y preguntas.