

Informe Civilization World

Diamis Alfonso Pérez C311
José Luis Leiva Fleitas C312

Descripción general:

Nuestro proyecto consiste en una simulación de un entorno donde interactúan tres civilizaciones con el objetivo de desarrollarse. Para desarrollarse una civilización debe aumentar su población creando, mejorando y conquistando ciudades. Con este fin es necesario entrenar tropas y desarrollar habilidades.

Habilidades:

Existen tres grupos de habilidades, el conocimiento de una permite el conocimiento de las otras. La segunda y la tercera habilidad de cada grupo solo se puede desbloquear si ya se tiene la primera y la segunda habilidad respectivamente.

Grupo 1:

- * Pesca: Permite recolectar peces.
- * Náutica: Permite hacer puertos.
- * Navegación: Permite la navegación por todo el mapa.

Grupo 2:

- * Escalada: Permite subir a las montañas.
- * Minería: Permite construir minas en las montañas.
- * Herrería: Permite el entrenamiento de soldados espadachines.

Grupo 3:

- * Organización: Permite recolectar frutos.
- * Agricultura: Permite la creación de granjas.
- * Milicias: Permite el desarrollo de soldados defensores.

Entorno

Un mapa es un tablero con casillas de diferentes tipos. Tenemos principalmente las casillas de tierra y de agua. Los terrenos terrestres están formados por llanuras y montañas, en las llanuras también encontramos casillas que

contienen frutos. Las aguas están formadas por playas y océanos, en las playas existen casillas con peces. La generación de los mapas se hizo de dos formas diferentes. Tenemos una generación de mapa realizada con CSP y otra generación del mapa realizada con una simulación. Para esta última generamos un mapa que solo contiene tierra y océano con un autómata celular de Moore. Después se le van incluyendo al mapa las ciudades y poblados con una probabilidad que depende de su distancia a las otras ciudades y poblados existentes. Luego los diferentes terrenos y recursos con cierta probabilidad.

Elementos de la simulación

Existen varios agentes. Tenemos a los líderes de las civilizaciones que toman las decisiones generales y los soldados que conforman las tropas, que realizan las acciones que ordena el líder y su resultado depende de las características propias de cada individuo.

Los líderes pueden tomar diferentes desiciones, que dividimos en tres grupos:

- * Acciones con las tropas:
 - * Entrenar soldados de tipo guerrero, espadachin o defensores.
 - * Ordenar a pelear a los soldados.
 - * Ordenar a conquistar una ciudad o un poblado.
- * Acciones que aumentan la población:
 - * Construcción de puertos.
 - * Construcción de minas.
 - * Construcción de granjas.
 - * Recolección de frutos.
 - * Pesca
 - * Desarrollo de habilidades

El jefe le ordena a los soldados las acciones que deben realizar, estos pueden hacerlas o no dependiendo de las características de cada individuos. Cada soldado tiene dos factores que intervienen en la ejecución de sus acciones.

* Pánico: es el miedo que tiene el soldado en el momento de realizar la acción. Depende de si se encuentra en territorio enemigo, rodeado de enemigos o en un terreno en el que no se siente cómodo.

* Inspiración: es la valentía del soldado en el momento de ejecutar la acción. Depende de si se encuentra en su territorio, rodeado de soldados que pertenecen a su tropa o en un terreno que domina.

Elementos de IA

Tenemos tres tipos de líderes. Uno random que escoge de forma aleatoria la acción a realizar. El Monte Carlo y el Monte Carlo Tree Search que explicaremos a continuación.

Monte Carlo Tree Search

La idea general de ir haciendo jugadas al azar desde la posición actual para muestrear cómo se comportan las distintas opciones que podemos tomar no es nueva. Sin embargo MCTS muestrea el árbol de jugadas para que los resultados obtenidos representen con cierta fiabilidad la bondad de las opciones existentes. Esencialmente, podemos ver cada uno de los movimientos posibles como una variable aleatoria cuyo muestreo nos informa acerca de lo bueno o malo que es dicho movimiento. El problema de que se ramifique tanto el juego que no podamos estar seguros de que el muestreo puramente al azar realmente sea capaz de captar la estructura intrínseca del árbol, se resuelve buscando un equilibrio entre la explotación y la exploración mediante la UCT.

$UCT(s) = \frac{Q(s)}{N(s)} + 2C\sqrt{\frac{\ln N(s_0)}{N(s)}}$ donde $N(s_0)$ es el número de veces que el nodo s_0 (el padre de s) ha sido visitado, $N(s)$ es el número de veces que el nodo hijo, s , ha sido visitado, $Q(s)$ es la recompensa total de todas las jugadas que pasan a través del nodo s , y $C>0$ es una constante para darle más peso a la explotación o a la exploración.

En vez de hacer muchas simulaciones puramente aleatorias, esta variante hace muchas iteraciones de un proceso que consta de varias fases y que tiene como objetivo mejorar nuestra información del sistema (exploración) a la vez que potencia aquellas opciones más prometedoras (explotación):

* La primera fase, Selección, se realiza mientras tengamos las estadísticas necesarias para tratar cada nodo/estado alcanzado como un problema de tragaperras múltiples. Comenzando por el nodo raíz, que representa el estado actual del juego/problema, seleccionamos recursivamente el nodo más urgente de acuerdo a una función de utilidad, hasta que se alcanza un nodo que, o bien representa un estado terminal, o bien no está completamente extendido (es decir, un nodo en el que hay posibles movimientos o acciones que no han sido consideradas).

* La segunda fase, Expansión, ocurre cuando ya no se puede aplicar la fase anterior. Para ello, se elige aleatoriamente una posición sucesora no visitada del nodo que no estaba completamente extendido y se añade un nuevo nodo al árbol de estadísticas con el fin de completar dicha información.

* Tras la expansión, nos encontramos en la fase de Simulación, que sigue los mismos parámetros de una simulación típica de Montecarlo. Es decir, partiendo del estado recién añadido se simula una partida, ya sea al azar, con una simple heurística de ponderación, o utilizando heurísticas y evaluaciones computacionalmente costosas para una estrategia más elaborada. Junto a la partida se obtiene un valor (premio, recompensa, etc) que determina la utilidad de esa rama para el jugador.

* Finalmente, la cuarta fase es la de Actualización o Retropropagación. Con el estado final de juego alcanzado en la fase anterior se actualizan las estadísticas de todas las posiciones previas visitadas durante la simulación completa que se ejecutó a partir del nuevo nodo (incluyendo la cuenta de ganancias si el jugador ganó la simulación).

Monte Carlo Simple

El jugador de Monte Carlo toma de forma aleatoria una de las posibles acciones a realizar y calcula con los rates de metropolis la probabilidad de realizar esta acción para decidir si la hace o no.

Los rates de Metrópolis tienen esta forma:

$r_n = \min(1, e^{\beta(h_n - h_a - G)})$ donde h_n es el valor de la heurística luego de realizar la acción, h_a es el valor de la heurística en el estado actual, el parámetro $\beta \geq 0$ está asociado al ruido y el parámetro G es el aumento del valor de la heurística esperado(valor de aumento medio). Monte Carlo realiza la acción si el aumento de la heurística que ella provoca es mayor que el esperado y si no se realiza con una cierta probabilidad que es menor mientras menor sea ese "aumento".

Monte Carlo Deep

Este jugador hace lo mismo que el jugador de Monte Carlo simple pero analizando varias jugadas en profundidad.

Resultados y conclusiones

Se realizaron 169 simulaciones.

Las puntuaciones medias fueron:

- * Para el jugador de MCTS : 1254.053254.
- * Para el jugador de MC : 826.124260.
- * Para el jugador aleatorio : 231.005917.

Los juegos ganados por jugadores fueron:

- * Para el jugador de MCTS : 109.5.
- * Para el jugador de MC : 47.5.
- * Para el jugador aleatorio : 12.0.

Podemos concluir que el mejor jugador es el que implementa Monte Carlo Tree Search. Sin embargo, el jugador que implementa Monte Carlo Simple también obtiene resultados muy buenos.

Mientras más historias pueda comprobar, o sea, mientras mayor sea el árbol de búsqueda del MCTS, mejores resultados se obtienen pero aumenta considerablemente el tiempo de duración de la simulación.

Se realizó un primer estudio con 70 simulaciones en las cuales los valores medios eran semejantes y la cantidad de partidos ganados y perdidos por jugador eran proporcionales. Por tanto consideramos que ya no es necesario realizar más simulaciones debido a que tiende a los mismos resultados.