Documento de Diseño Ecosistema

Lorenz Francisco Equihua Loeck

March 13, 2024

1 Introducción

Este proyecto es una simulación de un ecosistema, donde viven cuatro animales diferentes, conejos, zorros, venados y osos. Estos animales interactúan entre sí simulando comportamientos que tendrían en la vida real, por ejemplo, el zorro se come al conejo para sobrevivir y el conejo cuando ve al zorro se esconde. Cada uno de estos animales tiene variables como hambre, sed, edad, entre otros, que utiliza para tomar decisiones, por ejemplo, si un venado tiene sed buscará una fuente de agua para poder beber.

El mundo en el que viven estos animales es generado proceduralmente, esto significa que es diferente cada vez. El mundo tiene tres tipos de suelo: agua, pasto y tierra. Cada uno de estos tipos se pintan de colores diferentes para representarse de tal forma que simulen la realidad, igualmente estos tipos de suelo tienen funcionalidades diferentes, por ejemplo, el piso de agua no se puede usar para caminar por los animales, pero se utiliza como fuente de agua por los animales.

2 Especificaciones

2.1 Escenario

El escenario se genera de forma procedural utilizando un autómata celular 2D, este consiste en generar un mapa aleatorio, donde se le asignara al *tile* un estado, verdadero o falso. Una vez generado este mapa aleatorio se iterará las veces definidas por el diseñador, en estas iteraciones el autómata celular seguirá la regla especificada, si el *tile* actual tiene cinco o más se marcará como verdadero, en caso contrario mantendrá su estado actual. Terminando estas iteraciones se genera el escenario que ve el usuario, se itera por el arreglo y dependiendo si el tile está marcado como verdadero, se considera un piso en el que pueden caminar los animales, o falso, que se considera un piso de agua, en el cual no pueden caminar los animales, pero usaran para beber.

2.2 Agentes

2.2.1 Agente Base

La clase *BaseAgent* se encarga de guardar las características de los agentes como: velocidad máxima, edad máxima, rango de visión, género, entre otras. Estas características son usadas para controlar el comportamiento de cada agente.

2.2.2 Conejo

Los conejos son controlados por la clase *RabbitAgent*, la cual se encarga de manejar el comportamiento de los conejos tomando decisiones dependiendo de las necesidades que tenga, por ejemplo, si el conejo tiene hambre buscará arbustos para poder comer, estos tienen que estar en el rango de visión del conejo

para ser detectados, en caso de encontrar el arbusto, se moverá hacia él para comerlo, en caso contrario seguirá buscando.

2.2.3 Zorro

Los zorros son controlados por la clase *FoxAgent*, la cual se encarga de manejar el comportamiento de los zorros tomando decisiones dependiendo de las necesidades que tenga, por ejemplo, si el conejo tiene hambre buscará conejos para poder comer, estos tienen que estar en el rango de visión del conejo para ser detectados, en caso de encontrar el conejo, se moverá hacia él para comerlo, en caso contrario seguirá buscando.

2.2.4 Venado

Los venados son controlados por la clase *DeerAgent*, la cual se encarga de manejar el comportamiento de los venados tomando decisiones dependiendo de las necesidades que tenga, por ejemplo, si el venado tiene hambre buscará arbustos para poder comer, estos tienen que estar en el rango de visión del conejo para ser detectados, en caso de encontrar el arbusto, se moverá hacia él para comerlo, en caso contrario seguirá buscando.

2.2.5 Oso

Los osos son controlados por la clase *BearAgent*, la cual se encarga de manejar el comportamiento de los osos tomando decisiones dependiendo de las necesidades que tenga, por ejemplo, si el conejo tiene hambre buscará conejos o venados para poder comer, estos tienen que estar en el rango de visión del oso para ser detectados, en caso de encontrar el conejo o venado, se moverá hacia él para comerlo, en caso contrario seguirá buscando.

2.3 Algoritmo Genético

Cuando dos animales se aparean, la madre guarda la información del agente base de su pareja, después del tiempo de gestación nacerá un bebe que recibirá las características de su agente base de alguno de sus papás. Esto se hace en la clase *GeneticsManager* que tiene una función que recibe la información base del agente papá y la agente mamá, cada variable del nuevo agente se define aleatoriamente entre el gen del papá o del de la mamá, pero hay una probabilidad baja de que este gen mute, si este es el caso, se toma el gen del papá o del de la mamá y se multiplica por un valor aleatorio entre 0.1 y 2. Una vez definida esta información, el nuevo agente tiene su propia información base que puede evolucionar a la especie, por ejemplo, en un ecosistema sin depredadores en el que solo viven conejos, se observó que los conejos evolucionaron a tener una gran resistencia al hambre y a tener sed, por lo que solo se dedicaban a aparearse, pero si se agregaba a los zorros, los conejos evolucionaban a tener un gran rango de visión, para así poder estar siempre lejos de los zorros.