

EvoSim

Integrantes

- Alejandro Yero Valdes C-311
- Leismael Sosa Hernandez C-312
- Eduardo Garcia Maleta C-311

Descripción del proyecto

El proyecto consistirá en una simulación de organismos simples que pueden interactuar entre ellos y con el medio. el comportamiento de estos estará definido por una cadena genética y otras características básicas, como pueden ser el tamaño de la especie, el tiempo de vida o el costo biológico (la cantidad de nutrientes que necesita para sobrevivir), estas últimas pueden ser modificadas por la cadena de genes. Los genes de un organismos podrán ser definidos al inicio de la simulación y redefinidos luego por mutaciones o recombinación de genes entre miembros de una misma especie dependiendo de características como el tipo de reproducción y se elidirán de un conjunto predefinido. La piscina de genes la modelaremos como un grafo dirigido para simular dependencia de funcionalidades, por ejemplo, para poder diferenciar colores primero el organismo debe haber desarrollado visión. También dividiremos los genes en distintos tipos, por ejemplo mientras algunos genes pueden aportar características físicas para que el organismo interactúe con el medio, como audición o visión, otros pueden permitirle procesar información obtenida mediante estos últimos o determinar como el organismo tomara decisiones a partir de información obtenida. El objetivo del proyecto es observar el comportamiento de este tipo de sistemas y ver como se desarrollan las especies a partir de la interacción entre los organismos y el medio.

Objetivo

El objetivo del proyecto consiste en observar cuales cadenas genéticas específicas (lo que se traduce en un grupo de características y comportamientos) garantizan una mejor adaptación a un medio con unas reglas definidas mediante un proceso de optimización.

Inteligencia Artificial

En el proyecto utilizaremos inteligencia artificial como parte de los comportamientos que pueden desarrollar los algoritmos y para intentar generar organismos que logren sobrevivir durante mas

tiempo. Para esto utilizaremos algoritmos genéticos para la evolución de los organismos, en los cuales se definirá la cadena genética de cada individuo y se definirá como se reproducen y mutan los genes. También utilizaremos algoritmos de búsqueda para determinar como los organismos interactúan con el medio y como toman decisiones. Por ejemplo, para determinar como un organismo se mueve en el medio, podemos utilizar un algoritmo de búsqueda para determinar el camino mas corto entre el organismo y el alimento.

Compilación

Arquitectura del Compilador

Dominio del DSL

El dominio de aplicación de nuestro DSL es la creación de nuevos organismos en entornos (mundos) personalizados donde estos se desarrollarán por medio de nuestro motor de simulación EvoSim. Para lograr esto, nuestro DSL tiene sintaxis especializada en la creación estas entidades. En concreto este permite la creación de genes, ADN (secuencia de genes), Behaviors, Organismos (tiene un ADN y un Behavior), Mundo y Simulación.

Sintaxis

Como el objetivo de nuestro DSL es simplificar el trabajo de construir una simulación en EvoSim se hizo necesario remover características de lenguajes de propósito general que no eran necesarios para nuestro problema como por ejemplo la creación de clases, pero se mantuvieron algunas características como la creación de funciones dentro de las cuales se pueden ejecutar instrucciones de control de flujo y se pueden crear variables. Vale aclarar que estas funciones solo pueden estar presentes en objetos especiales del DSL que se verán más adelante en el documento.

Al nuestra simulación de EvoSim tener un conjunto finito de objetos que se pueden crear, hace que en nuestro DSL sea posible tener sintaxis específica para estos. Las entidades que permite crear nuestro DSL son:

- Genes
- ADN
- Behaviors
- Organismos
- Mundo

- Simulación

Comenzaremos hablando de las entidades de nuestra simulación y como estas se crean con la sintaxis de nuestro lenguaje para luego explicar como se pueden crear funciones (y en donde) con controles de flujo y con la posibilidad de crear variables temporales.

Entidades

Genes

Los genes de nuestra simulación son de 3 tipos:

- Físicos
- Sensoriales
- De acción

Los genes son utilizados por otra estructura que se verá más adelante en el documento, el ADN. El objetivo fundamental de los genes es describir las características del organismo. Los genes físicos describen sus capacidades físicas del organismo (si tiene piernas podrá desplazarse, si tiene ojos podrá ver, etc.). Los genes sensoriales definen lo que puede percibir del mundo. Hay genes sensoriales que solo los puede tener un organismo si tiene un gen físico que lo permita (el sensor de la vista solo lo puede tener el organismo que tenga ojos, etc.). Los genes de acción son para definir cuales acciones puede realizar el organismo en el mundo donde se encuentre.

En nuestro DSL se distinguen 2 tipos de genes: los que tienen parámetros personalizables y los que no.

Si el gen es modificable se pudiera crear uno de la siguiente forma:

```
gene eyes aaa {  
  
  value 2 in {1 4}  
  
  mutation {  
  
    chance 0.75  
  
    step 2  
  
  },
```

```
}  
  
}
```

El fragmento de código anterior define un gen modificable de tipo `eye` con un identificador único `aaa`. Sus parámetros modificables son un `value` que se define con valor 2 pero que pudiera estar en un rango entre 1 y 4. También tiene un parámetro `mutation` que se inicializa con los valores 0.75 para `chance` y 2 para `step`.

Si el gen no es modificable se pudiera crear uno de la siguiente forma:

```
gene vision
```

Aquí se definió un gen de tipo `vision` que al no tener ningún parámetro hace que no sea necesario agregarle un identificador.

 **Note**

Los genes disponibles para usarse en el DSL dependen del motor de simulación EvoSim

ADN

El ADN es la información que necesita un organismo para existir en nuestra simulación y este es formado por una secuencia de genes

Un ADN se puede crear de la siguiente forma:

```
dna ddd {aaa vision}
```

o de la siguiente forma si un ADN quiere agregar los genes de otro adentro de si mismo:

```
dna eee {dna ddd bbb ccc}
```

Funciones

Instrucciones de Control

Variables

