Instituto Tecnológico de Costa Rica
Sede Regional San Carlos

Escuela de Computación

Lenguajes de programación

Proyecto 2

Responsables:
Crisly González Sánchez

19 de abril del 2016

Santa Clara, San Carlos

Introducción

La programación en la cual se realizan una serie de pasos,se define memoria, funcionalidad y además a través de este modelo podemos llegar al objetivo o resultado, a esto se le conoce como programación imperativa. Aunque aún es muy utilizada, en comparación de paradigmas con la programación funcional viene siendo más eficiente la segunda mencionada ya que, se compone de llamadas a funciones y retorno de funciones.

En la programación funcional hay menos gasto de memoria, tiene como característica fundamental un concepto de función diferente al que los programadores están acostumbrados, no hay variables por lo que las funciones solo trataran con sus valores de entrada y con constantes predefinidas.

Algunas características del lenguajes funcional son: transparencia referencial, se refiere a que al no haber variables, las funciones no dependen del entorno; altísima abstracción, implementa funciones de orden superior, funciones anónimas,operaciones que son totalmente funcionales, etc; legibilidad y flexibilidad, por lo general entre más abstracto sea un lenguaje de programación mayor será su legibilidad, además al ser totalmente compuesto de funciones se pueden probar individualmente sin preocuparnos por las demás funciones existentes.

Análisis del problema

Se requiere el desarrollo de un sistema de juego basado en el paradigma funcional el cual debe ser implementado en lenguaje de programación Python, se solicitó por el cliente que el frontend del juego “misioneros vrs caníbales” sea desarrollado en lenguajes de programación javascript para darle funcionalidad al juego, de manera que este sea interactiva. Es importante agregar que se debe tener una interfaz amigable y sencilla el cual puede realizar en HTML5 y CSS3 para agregar estilos.

El sistema requiere que se cumplan los siguientes requisitos funcionales:

El inicio del sistema debe realizarse en una secuencia de (3,3,1) (0, 0, 0) donde cada número indicará cantidades de caníbales y misioneros, asimismo tener un numero de representación de la balsa en la que serán transportados de ambos lados del mar.
El sistema deberá permitir que los caníbales y misioneros sean transportados en un bote de izquierda a derecha o revés, cumpliendo las mínimas funcionalidades previamente definidas.

El sistema tiene como requisito no funcional los siguientes requerimientos:

El sistema deberá ser desarrollado basado en el entorno funcional, aplicando el concepto orden superior.
El lenguaje de programación core a utilizar deberá ser Python.
El frontend de la aplicación deberá estar desarrollada con el lenguaje de programación javascript e interfaz y diseños html5 y CCS3

El sistema deberá incluir al menos los movimientos básicos, previamente estipulados por el cliente en la documentación oficial,el bote no lleva ningún misionero y lleva un caníbal,el bote lleva 2 misioneros y ningún caníbal, el bote lleva 2 caníbales y ningún misionero,el bote lleva 1 caníbal y un misionero,el bote no lleva

ningún caníbal y lleva un misionero.

El sistema deberá ser un juego automático, sin interacción alguna con el usuario, además deberá contar con tiempos de espera al dar la solución.

Solución del problema

El juego misioneros vrs caníbales fue desarrollado en lenguaje de programación python con lógica del paradigma funcional, cuenta con los movimientos básicos que debían tener los caníbales y misioneros, el bote no lleva ningún misionero y lleva un caníbal,el bote lleva 2 misioneros y ningún caníbal, el bote lleva 2 caníbales y ningún misionero,el bote lleva 1 caníbal y un misionero,el bote no lleva ningún caníbal y lleva un misionero. Además cuenta con un método que averigua los vecinos de los pares ordenados que generan [3,3,1] [0,0,0].

Luego se desarrolló un método extender que recibe una ruta verifica si está vacía y sino llama los vecinos de dicha ruta y se guardan en una lista, donde luego se analiza si cada elemento de la lista es miembro de la ruta, si se cumple agrega una lista vacía y si no guarda una lista de resultados con el elemento y la ruta en la posición [0], luego generada la lista de vecinos, se eliminan las listas vacías para quedar únicamente con las rutas absolutas **[[3,3,1] ,[0,0,0]]** [[3,2,1], [0,1,0]].

Al desarrollo le fue aplicado un método profundidad en el cual por medio de profundidad aux se utiliza el extender con lista [0] y va disminuyendo hasta que tenga un len de 0 y entonces le es agregado a la lista principal.

El método solución comprueba que el escenario recibido en la posición derecha es decir escenario[1][0] , escenario [1][1], escenario [1][2] tengan los valores iniciales escenario [3, 3,1] para retornar la solución.

Además se agregó un metodo archivo que es el encargado de crear la lista con rutas con una extensión .js la cual les pertenecen a los archivos javascript. Para que

se pueda complementar con la interfaz web se deberá adjuntar la carpeta disponible en tarea 2, copiarla en C (proyectoCrisly) y al finalizar el juego en python se agregará un archivo llamado soluciones.

El archivo soluciones.js tendrá todas las rutas solución que se crearon con el análisis en profundidad del juego misioneros vrs caníbales, este estará conectado a funciones.js. La carpeta está contenida un index html , 2 archivos javascript, 1 de diseño CSS, Bootstrap y Jquery.

El juego se puede ejecutar desde el index al abrirlo en un explorador, el index contiene la conexión con los archivos .js, .css, además de las rutas para acceder a los .css de bootstrap y a JQuery. La estructura está definida por una etiqueta <head> y < body> esta última contiene <header> <section> y <articule> donde está contenida una tabla con 3 <tr> uno para contener la imagen superior, las imágenes y entorno de juego y la imagen inferior de pasto y agua. Además de contener un botón que llama la función solución.

La función solución en javascript verifica que la lista creada en el archivo soluciones.js(desde python) no esté vacía de no cumplirse hay una variable global que cuenta la cantidad de soluciones y se envía a soluciónAux con el index igual a la variable solución index (variable global), soluciónAux recibe una ruta y mediante un timer, refresca y va pintando la ruta por el dibujar.

El método dibujar recibe un escenario, llama a dibujar misioneros y caníbales para ser inicializados, con las cantidades que trae escenario en cada posición, luego analiza el escenario en la posición [0][2] si el valor es 1 entonces pone la balsa a la izq si no lo pone a la derecha. Hay un método dibujar caníbales y misioneros que reciben la cantidad que deben pintar y la posición, por medio de los selectores en JQuery y el id colocado en la tabla del html , la imagen se sobreescribe para poder verlo en movimiento. Cabe destacar que soluciones Aux es el encargado de enviar cada solución a pintarse y avanzar hasta llegar al final, el juego está diseñado para que inicie con la primera solución una vez que se evalúan todas en la lista

soluciones. Y finalmente hay un método limpar que inicializa todo el entorno de juego.

Análisis de resultados

Tarea	Estado	Observaciones
Python		
Desarrollo de movimientos de caníbales vrs misioneros	100%	
Desarrollo método vecinos, extender,profundidad, remove null	100%	
Programación funcional del juego caníbales vrs misioneros	100%	
Creación de conexión con archivo javascript	100%	
Generación de las rutas posibles	100%	
Comprobación de la soluciones generadas	100%	
Función miembro de una lista	100%	
Interfaz Web		
Creación de entorno juego	100%	
Movimiento de los caníbales y misioneros	100%	
Implementación de un timer en el juego	100%	
Animación de todas las soluciones del juego	100%	
Lectura de soluciones generadas en python e animadas	100%	

Conclusión

La búsqueda en profundidad recursiva aunque consume más recursos que una solución iterativa es una potencial forma de recorrer todos los posibles escenarios dentro de una solución.

Python es un lenguaje multiparadigma, este aspecto facilitó el enfoque funcional que se le dio a la solución de los misioneros y caníbales.

Recomendaciones

1. El juego debe será implementado en un framework de desarrollo, para optimizar recursos y hacerlo más eficiente, además de que el modelo del proyecto podría ser mejor.

2. El juego deberá emplearse una interfaz más interactiva con mensajes al usuario, para estar enterado de cuando cambia de solución.
3. El desarrollo de la interfaz web requiere conocimientos previos en javascript, JQuery , HTML y CSS.

4. El desarrollo en el lenguaje de programación Python requiere conocer el lenguaje de programación y sus funcionalidad.

5. La mayor parte de las funciones implícitas en el lenguaje de programación scheme no funcionan en python.

Bibliografía

SR. (SR). programación-funcional. 18/04/2016, de genbetadev Sitio web: http://www.genbetadev.com/paradigmas-de-programacion/programacion-funcional-un-enfoque-diferente-a-los-problemas-de-siempre

SR. (2015). Este artículo se tradujo de forma manual. Mueva el puntero sobre las frases del artículo para ver el texto original. Más información. Traducción Original Diferencias entre programación funcional y programación imperativa. 18/04/2016, de microsoft Sitio web: https://msdn.microsoft.com/es-es/library/bb669144.aspx

w3school. (SR). SR. 18/04/2016, de w3school Sitio web: http://www.w3schools.com/js/