1-Identificación del documento:

Nombre: Pablo Gutiérrez Hidalgo

Curso: CC3501-1 “Modelación y Computación Gráfica para Ingenieros”

Fecha: 26-07-2020

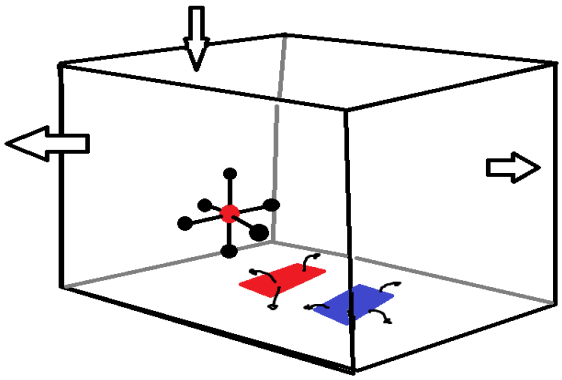
2-Solución propuesta:

Aquarium-solver.py es el archivo encargado de resolver la ecuación de Laplace para el acuario, los valores determinantes del problema (largo, ancho, alto, y temperaturas en los casos de borde) vienen determinadas en un archivo .json.

Para analizar los resultados generamos grillas, las cuales serán evaluadas en cada punto, tomando un stencil de 7 puntos que irá recorriendo todo el espacio tridimensional con un paso de h=0.2 (lo cual entregó buenos resultados) considerando casos bordes y evaluándolos con el respectivo método visto en el curso. Finalmente obtenemos un sistema de ecuaciones y poder obtener los valores de la temperatura para cada punto del problema, estos valores se muestras por medio de matplotlib y se guardan utilizando np.save() para utilizarlos en el segundo archivo de nuestra tarea.

Aquarium-view.py es el archivo encargado de simular el acuario con los peces más las funciones de poder ver los voxeles que cumplen los requerimientos para cada pez, esto lo hacemos primeramente leyendo los datos de Aquarium-solver.py y de un nuevo archivo .json que establece la cantidad de peces y las temperaturas aceptadas por cada uno.

La metodología para ver el acuario consta en hacer 3 tipos de peces utilizando grafos de escena (agregándole el movimiento de la cola) para luego ponerlo en las zonas donde podrían ir.



Modelo stencil 7 puntos

Para la creación de la pecera se tomaron las medidas del arreglo entregado por np.load y se construyó una pecera escalándola con un parámetro s. Para la creación de los voxeles se recorrieron todos los voxeles para ver cuales cumplían las condiciones de temperatura optima para a, b y c respectivamente, las cuales se convertían en una gpu que contuviera a todos los posibles voxeles. Las coordenadas se guardan en arreglos para un futuro uso.

Finalmente, escalamos cada una de estas 3 gpu’s de voxeles para que se adecuen al tamaño de la pecera, siendo controladas por parámetros en el controller que indiquen cuando se mostraría cada una.

Para la ubicación de los peces, del arreglo con todas las posibles coordenadas para cada tipo de pez, escogemos aleatoriamente tantas coordenadas optimas como peces necesitemos ubicar, las trasladamos (junto con un escalamiento para que se adecuén al acuario y finalmente podemos mostrar el acuario, las zonas y los peces.

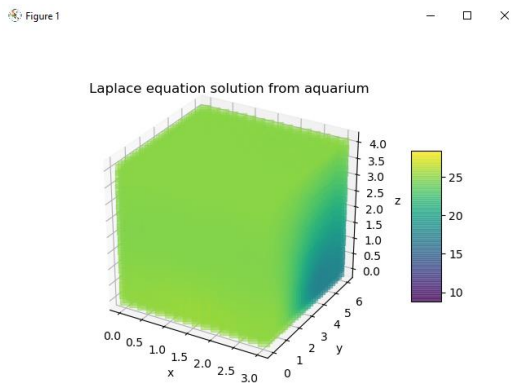
3-Instrucciones de ejecución:

Aquarium-solver.py se ejecuta escribiendo “python aquarium-solver.py problem-setup.json” donde “problem-setup.json” es un archivo con las indicaciones para iniciar el problema. En este archivo se generará un documento útil para el segundo programa (resolviendo el sistema de ecuaciones), a su vez, se muestra en matplotlib un espacio 3D donde podemos ver la distribución de las temperaturas.

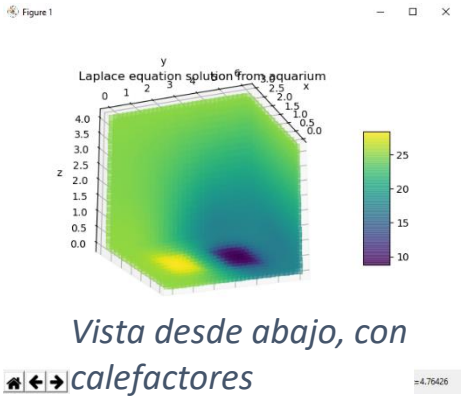
Aquarium-view.py se ejecuta escribiendo “python aquarium-view.py view-setup.json” donde “view-setup.json” contiene la información básica de los peces. Este programa muestra una pecera con la cantidad de peces indicada en el archivo .json, al igual que las limitaciones de temperatura. En este programa tenemos 7 interacciones con el usuario: la flechas horizontales permiten rotar la cámara, mientras que las flechas verticales permiten hacer un acercamiento o un alejamiento, las teclas “A”, “B” y “C” permiten mostrar los voxeles óptimos para cada tipo de pez.

4-Resultados:

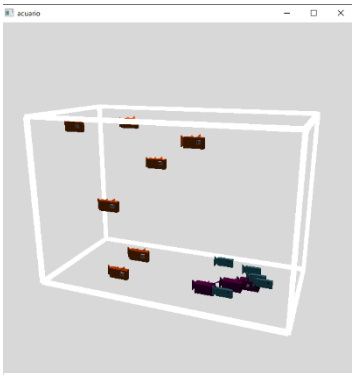
A continuación, se mostrarán algunas capturas de cada archivo.



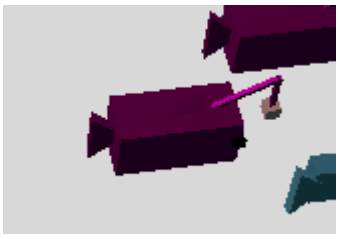
Pecera en matplotlib

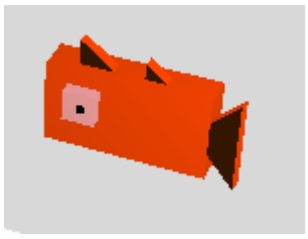


Vista desde abajo, con calefactores

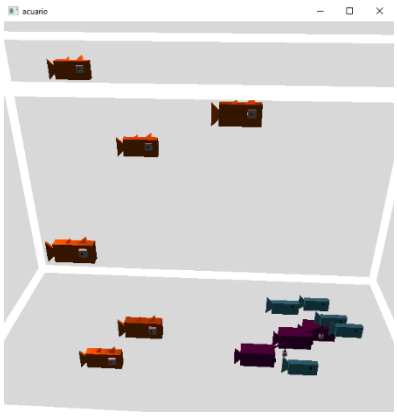


Vista de la pecera

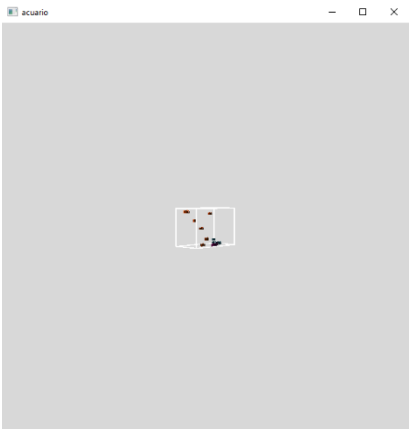




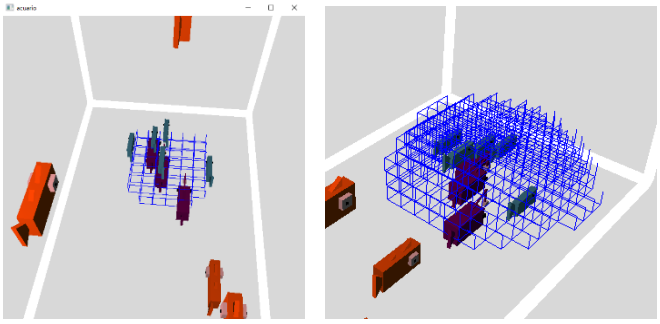
Los 3 tipos de peces



Pecera con un poco de zoom in



Mucho zoom out



Voxeles con las teclas A,B y C presionadas