Vanier College

Department of Computer Science

Celestial Simulator

PCA Report B

Ethan Tran

Integrative Project in Computer Science
420-204-RE

Yi Wang

May 27, 2024

Description du projet

Le *Celestial Simulator* est une application de bureau construite avec JavaFX qui vise à créer une simulation précise du ***n-body*** *problem*, qui implique de prédire les mouvements des objets célestes sous l'influence de la gravité. Le ***n-body*** *problem* étant un problème non résolu dans le domaine de la physique classique, la simulation tente de recréer les interactions gravitationnelles entre les corps en calculant les forces gravitationnelles entre ces objets à chaque image et en mettant à jour les positions de ces objets célestes. La simulation, qui s'inspire de la loi de la gravitation universelle de Newton, adopte deux approches pour créer le mouvement céleste. La première approche est l'algorithme de la somme directe, qui calcule la force gravitationnelle entre chaque paire de corps célestes individuellement. La seconde approche est l'algorithme de *Barnes-Hut*, qui améliore l'efficacité en estimant que les corps suffisamment éloignés ne font qu'un. Dans ce contexte, chaque corps dans l'espace génère un champ gravitationnel autour de lui, connu en physique sous le nom de champ de vecteurs, qui détermine la force gravitationnelle exercée sur un autre corps à n'importe quel endroit du champ. La simulation visualise ce champ gravitationnel à l'aide de flèches indiquant la direction et la magnitude. Les flèches pointent vers la direction de l'attraction gravitationnelle et leur couleur représente l'intensité de la force. La simulation est hautement interactive et offre une variété de contrôles à l'utilisateur. Les utilisateurs peuvent régler la caméra en effectuant des panoramiques, des zooms et en sélectionnant des planètes spécifiques. Des ajustements temporels sont également disponibles, y compris le contrôle de la vitesse, les fonctions de pause et de lecture. En outre, la simulation permet la manipulation d'objets, tels que la création et la suppression d'objets célestes. Les utilisateurs peuvent également modifier les paramètres de simulation, notamment le critère de Barnes-Hut et la constante gravitationnelle G. En outre, les utilisateurs peuvent sauvegarder et charger un état de simulation, ce qui permet de poursuivre et de partager facilement leurs simulations.

Approach taken

The development of the Celestial Simulator involved a structured and collaborative approach to ensure efficient task management, project coordination, and effective communication among team members. To ensure that the project progressed smoothly, we adopted an agile methodology, breaking down the project into tasks. Each sprint had clearly defined goals and deliverables, allowing the team to focus on specific aspects of the simulator sequentially. We utilized project management tools such as Jira to create, assign, and track tasks. This enabled us to maintain an organized workflow, identify bottlenecks early, and adjust priorities as needed. Project coordination was essential to align the efforts of team members and integrate various components seamlessly. We designated roles and responsibilities based on individual strengths and expertise. Regular meetings (in-class) were held to review progress, discuss challenges, and plan the next steps. We established multiple channels to facilitate clear and consistent communication. Discord was our primary platform for day-to-day interactions, allowing for instant messaging, file sharing, and real-time collaboration. Additionally, we kept track of meeting data on Google Docs to keep everyone up to date. Additionally, extensive research was conducted on the n-body problem, Newton's law of universal gravitation, and existing algorithms such as the direct sum and Barnes-Hut algorithms. Brainstorming sessions were held to generate ideas and innovative features that could enhance the simulator's functionality and user experience. The development process was iterative, with continuous integration and testing. Code was regularly committed to a shared repository on GitHub, allowing for version control and collaborative development. We implemented unit tests to ensure the reliability and accuracy of the simulation.

Contributions of each discipline

The discipline of **Mechanics** played a role in the development of the Celestial Simulator, as it provided the foundational principles necessary for accurately simulating celestial bodies. More precisely, the simulation used many of the concepts taught in classical physics, which is the main focus of the discipline of Mechanics (203-NYA-05). At its core lies Newton's law of universal gravitation, which dictates the gravitational force between any two objects is directly proportional to the product of their masses and inversely proportional to the square of the distance between their centers. By adhering to this law, the simulator calculates the gravitational forces acting on each celestial body, updating their positions accordingly. Additionally, classical mechanics concepts such as linear momentum conservation are employed for collision detection, ensuring realistic interactions between bodies. Furthermore, in implementing the Barnes-Hut algorithm, which optimizes computational efficiency by approximating distant bodies as a single mass, the simulation relies on the concept of the center of mass. This notion allows the simulation to effectively track the center of mass of quadrants, enabling efficient grouping of distant bodies for gravitational calculations.

In addition, the material taught in the computer science disciplines, such as **Data Structures and Object-Oriented Programming** (420-202-RE) were used throughout the integrative project. Indeed, data structures were extensively utilized, particularly in the implementation of the Barnes-Hut algorithm. Quad trees, a fundamental data structure, were employed to efficiently partition space into hierarchical quadrants, facilitating the grouping of celestial bodies based on proximity.

Recursion, another concept learned in Data Structures, was instrumental in traversing these quad trees, allowing for the recursive approximation of distant bodies. Moreover, the time complexity analysis of algorithms covered in 420-202-RE provided crucial insights into optimizing the performance of the simulation. The direct sum algorithm computes the gravitational force between each pair of celestial bodies individually. With n bodies in the simulation, this results in a time complexity of $O(n^2)$. As the number of bodies increases, the computational cost grows quadratically, which can quickly become impractical for simulations with many celestial objects. Conversely, the Barnes-Hut algorithm offers a substantial improvement in efficiency. By recursively partitioning space into quadrants and approximating distant bodies as a single mass, the algorithm reduces the number of gravitational force calculations needed. This hierarchical approach leads to a time complexity of $O(n\ log\ n)$ in the average case, and the changing of the Barnes-Hut criterion allows to trade off accuracy for speed. By understanding the computational demands of various algorithms, including the Barnes-Hut algorithm, we were able to make informed decisions regarding algorithmic efficiency.

JavaFX and UI design principles covered in **Program Development in a Graphical Environment** (420-203-RE) played a pivotal role in shaping the user interface and overall user experience of the Celestial Simulator. Concepts like event handling, scene management, animations, timelines, taught in 420-203-RE, were instrumental in implementing user interactions within the simulation and ensuring the smoothness of simulation.

Challenges

Developing the Celestial Simulator presented several challenges that required innovative solutions. Firstly, since JavaFX 3D is relatively new, this posed a hurdle due to limited documentation and resources. To overcome this, our team engaged in extensive experimentation and collaborated closely. However, it took much more time than expected to learn how JavaFX 3D worked. Moreover, using the Perspective Camera in JavaFX 3D was essential for immersive user experiences, but required immense amounts of testing to get it working properly. Implementing the Barnes-Hut algorithm for efficiency required meticulous attention to detail and optimization techniques. Additionally, coordinating tasks and managing team dynamics presented its own set of obstacles, requiring clear communication channels and effective project management strategies to maintain momentum and meet deadlines. It was often hard to coordinate and keep track of tasks all the time because they kept changing as we developed the application.

Strengths and weaknesses

The Celestial Simulator showcases several strengths that make it a valuable tool for both enthusiasts and students interested in celestial mechanics. Its accurate representation of the n-body problem, utilizing both direct sum and Barnes-Hut algorithms, ensures that users can observe realistic celestial motion. The visualization of gravitational fields adds an educational dimension, aiding in understanding complex gravitational interactions. The application's interactivity empowers users to explore and manipulate celestial scenarios with ease, thanks to features like camera controls, time adjustments, and object manipulation. Additionally, the ability to modify simulation parameters and save/load states enhances versatility and convenience. However, despite

its strengths, the Celestial Simulator still faces limitations. Its computational demands, particularly when simulating large numbers of celestial bodies, could strain system resources, potentially leading to performance issues on less powerful hardware. Even with the Barnes-Hut algorithm, the simulation still lags out with large numbers of bodies. Especially when toggling visualization for vector fields and the visualization for the Barnes-Hut algorithm, the simulation slows down and lags significantly. Furthermore, while the application offers extensive user controls, the learning curve for understanding and effectively utilizing these features might be steep for new users.