CUDA Project: Gamified Simulation of Interactive Particles

This project consists of creating an interactive particle simulation using CUDA for parallel calculations and Raylib for display. It must be possible to interact with the simulation and to observe dynamic and visually attractive behaviors in real time. The project introduces concepts of parallel computing, GPU memory management, and integration with a graphics library.

1. Project objectives :

- Discover and apply the basics of CUDA for massively parallel computing.
- Learn how to use Raylib to display objects in real time.
- Develop an interactive application with gamified user interactions.
- Understand the basic principles of physics simulation (forces, collisions, etc.).

2. Development Milestones:

1. Initialization:
   - Create a particle grid with initial positions, velocities, and colors (no need to display the grid).
   - Initialize this data on the GPU.

2. Particle Calculation (CUDA):
   - Implement a CUDA kernel to update particle positions and velocities.
   - Manage interactions between nearby particles (attraction/repulsion).

3. Display (Raylib):
   - Display the particles with colored circles on the screen.
   - Update positions in real-time.

4. User Interaction / Gamification (Take your pick):
   - Add mouse-controlled attraction/repulsive fields.
   - Allow the user to add obstacles or change settings with the keyboard.
   - Add a goal (for example, guide certain particles to a target).
   - Time performance or add an action-based score.
   - …

3. Technical constraints and rendering

You will need to use visual studio community to code your program, which will need to be in C++ (you'll have to download the cuda toolkit in order to be able to compile your kernels : https://developer.nvidia.com/cuda-downloads
You'll also need to work with Git to handle merging each other's codes.
You will present your work using a powerpoint to illustrate your work. You will have 10 minutes for presentation, followed by 5 minutes of questions.

4. Evaluation scale:

1. Presentation (3 points)
   - Clarity of presentation
   - Communication and balance within the team
   - Visualization of results

2. Git and project management (3 points)
   - Organization with Git (git Workflow)
   - Collaborative work

3. Cuda and C++ (7 points)
   - Code quality
   - CUDA Initialization and Calculations, Kernel Usage, Memory Management
   - Correct grid and initial data creation
   - Implementation of CUDA calculations for interactions and explanation of the concepts behind it (choice, thread management, synchronization...)
   - Optimization and dependency management
   -

4. Raylib display (4 points):
   - Correct representation of moving particles
   - Smooth, interactive particle editing

5. Interaction and Gameplay (3 points):
   - Implementing mouse/keyboard interactions
   - Gamification with objectives or challenges.

Conclusion:

This project will allow you to discover the basics of CUDA while having fun with an interactive simulation. This playful approach will teach you how to combine parallel computing and real-time graphical display.