Using Cellular Automata To Simulate Liquid, Gas, and Powder Physics

Graham, Erick (School: InTech Collegiate High School)

Right now, most game engines only use a form of simulating physics called rigidbody physics. Rigidbody physics engines work well but are very limited as they only support simulating solid and basic objects. Using a method called “cellular automata” we can simulate physics by applying a set of behavioral rules related to surrounding pixels to each pixel on the screen and either move or remove that pixel based on that, allowing for far more advanced physics simulation. This project’s purpose is to integrate that principle of simulating physics as a tool for the Unity game engine, providing a way to simulate liquids, powders, and gases - along with some helper functions to make modifying the behavior rules for each pixel easy. I developed the program within the Unity game engine, and I used the C# programming language. I started the development by creating some common actions that each of the pixels could take. I then went to work on a way to update the pixels, checking whether some specific surrounding pixels were empty or not and then moving the pixel based on its predefined rules. After adding all the behavioral rules for the states of matter and making a couple of performance optimizations across 3 different iterations, I began working on adding velocity, acceleration, and force to the update loop, making it so that the velocity of the pixels decided how it would move itself or other pixels. The physics engine ended meeting my goals (which was to create a simple form of simulating particles that moves based on the basics of cellular automata). It has liquids, gases, and powders built in, and simple and easily modifiable behaviors for each of those.

Awards Won: