Escuela Superior de Cómputo
Instituto Politécnico Nacional
Ingeniería en Sistemas Computacionales

Evolutionary Computing

Practice 7: Particle Systems

Professor: Jorge Luis Rosas Trigueros
Student: Ayala Segoviano Donaldo Horacio
Creation date: October 23 2021
Delivery date: October 29 2021

1 Introduction

Swarm intelligence consists of two or more individuals independently, or at least partially independently, acquire information and these different packages of information are combined and processed through social interaction, which provides a solution to a cognitive problem in a way that cannot be implemented by isolated individuals.

The main elements of of swarm intelligence are the following:

- **Group-living agents:** each agent can independently gather information and also copy the behaviour of others.
- **Learning capability:** agents are able to store and recall information. This ability could lead to cooperative interaction networks and potentially enable collective solutions.
- **Signalling:** it is possible for agents to inform others without physically guiding them.
- **Division of labour:** every individual is capable of adopting any task or role.

1.1 Particle Systems

William Reeves (1983) defines **particle systems** as a collection of many many minute particles that together represent a fuzzy object. Over a period of time, particles are generated into a system, move and change from within the system, and die from the system.

Particle systems are widely used in computer graphics, since they allow to represent complex behaviors. Although this practice focuses on Particle Systems from the Swarm Intelligence point of view, there are some characteristics that are useful from the computer graphics definition. Reeves states that every time unit the following steps are performed:

1. New particles are generated into the system.
2. Each new particle is assigned its individual attributes,
3. any particles that have existed within the system past their prescribed lifetime are extinguished.
4. The remaining particles are moved and transformed according to their dynamic attributes and finally
5. An image of the living particles is rendered in a frame buffer.

From which we can take numbers 1, 2 and 4. So, in our case, it is enough to create new particles and add them into the system, each particle will have its own individual attributes and they will have to be initialized, and lastly, every unit of time, the particles will be moved and transformed according to their dynamic attributes. For this practice, step 5 will also be used, since we will visualize the particles and their behavior.

1.2 Neighbor influence

One of the most important things about a particle system is their attributes and behaviors. We learned from the particle swarm optimization algorithm that in order to find an optimal solution there is usually an influence from the best individual over the swarm, so all the other particles will be, at some point, influenced to follow the best individual. However, in this practice, it is going to be observed the behavior of a particle system that is influenced only by the nearby neighbors and by other elements such as physics.

1.3 Physics in particle systems

Particles can be influenced by several factors when taking decisions. We have seen so far only the influence of the fittest individual and the influence of the best solution found by the particle itself. However, there are several other factors that could influence on a particle, in this practice both the influence of nearby neighbors and the influence of physics over the particles will be explored to see how particle systems behaves.

2 Material and equipment

Following are the hardware and software used during the realization of this practice. ***Google Colaboratory*** was the tool used for the development of this practice and the next list shows the specifications of the hardware and software provided by Google Colab.

- **Hardware:**
  - **CPU:** Intel(R) Xeon(R) CPU @ 2.30GHz
  - **Memory:** 12GB
  - **Disk:** 108GB
- **Software:**
  - **Platform used:** *Google colaboratory* was used for this practice
  - **Programming language:** Python 3.7.11

3 Development

3.1 Mexican wave

The wave is an example of **metachronal rhythm** achieved in a packed stadium when successive groups of spectators briefly stand, yell, and raise their arms. Immediately upon stretching to full height, the spectator returns to the usual seated position, an example is shown in figure 3.1.

A **metachronal rhythm** or metachronal wave refers to wavy movements produced by the sequential action of structures such as cilia, segments of worms or legs. These movements produce the appearance of a travelling wave.

Figure 3.1: Example of a mexican wave at a stadium.

For this section, particle systems will be used to try to replicate the effect of a mexican wave, this will be achieved by setting each particle to be influenced by its neighbors.

3.1.1 Implementation and testing

The mexican wave will be visualized in 2D, each particle will have an initial radio, and each particle will be set to have an influence from the adjacent particles, this way, the particles will increase their radio as time goes on and based on their neighbors radios. In order for the particles not to grow indefinitely, once a particle hits the maximum radio, it will start decreasing until it returns to its initial position. To trigger a wave, it is necessary to initialize a particle with a bigger radio so it can influence other particles.

The behavior described above will be coded in Python programming language. Code implementation can be found by clicking **here**.

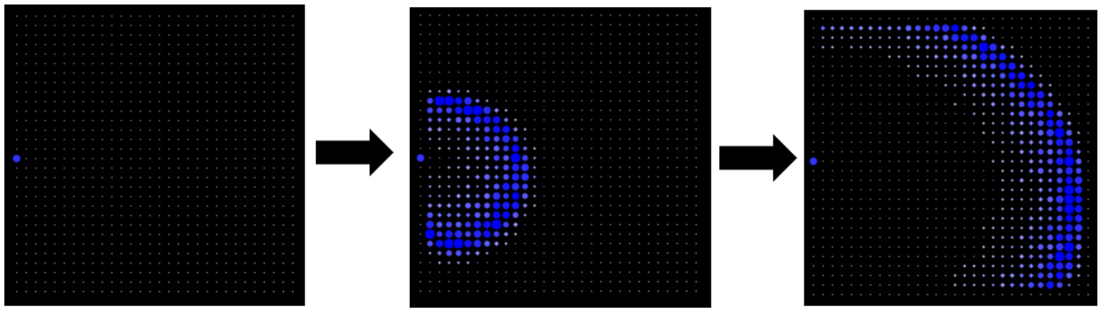

Figure 3.2: Execution of the mexican wave program.

Figure 3.2 shows different phases in the execution of the program of the mexican wave particle system. There are three phases, the initial one, shows the initial state and the only particle that starts influencing its neighbors, second and third screenshot, show the particles continuing influencing their neighbors and it can be observed how the particles that have already reached the maximum, start decreasing their radio until they return to its initial position

As a whole, it can be seen how the particles, all together form a wave. And this is a good example of why particle systems are used in computer graphics, they can have very complex behaviors that help to represent real life behaviors that otherwise would be really hard to obtain, some examples are the ones shown in figure 3.3.

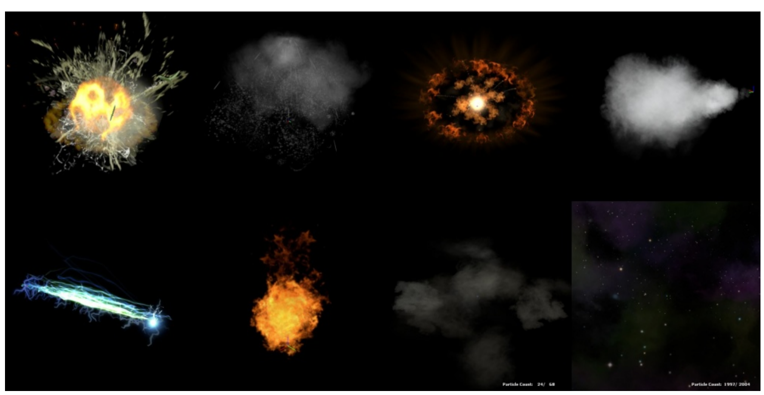

Figure 3.3: Examples of applications of particle systems in computer graphics.

3.2 Physics in particle systems

In this section, physics will be added to a particle system to simulate a crowd trying to go through a small exit. The physics added to the particle systems are the ones of the three Newton's laws:

1. An object stays in its current state unless it is acted upon by a force.
2. The acceleration of an object generates a force in the same direction.
3. Applying force to an object results in an equal and opposite force.

Each particle will have an initial position, a velocity and to achieve the effect of a collision system, a force or acceleration will be used every time the particles have a collision. Initially each particle will start at a random position in the left area of the room, they will have a velocity of between 0 and 1 to the right and a random velocity between -0.5 and 0.5 in the vertical axis. This way all of the particles will start going towards the exit, which is located at the right.

In addition to the initial force going to the right, a force will be applied to all of the particles to pull them closer to the exit and accelerate the process of evacuation of the particles.

In this practice the theory of placing an obstacle near an exit to improve the evacuation time will be tested, so, first, the program will be run with no obstacles, and then placing a small obstacle in front of the exit. The results will be compared to verify if this technique works for particle systems.

3.2.1 Implementation and testing

After implementing the particle system applying physics, in Python programming language, it is now going to be executed to see the behavior of the particles when there is no obstacle near the exit. The implementation can be found by clicking **here**.

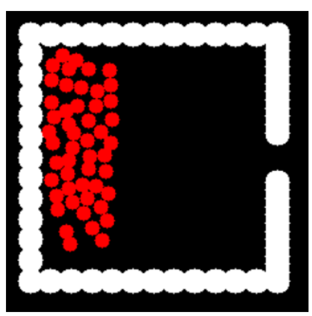

Figure 3.4: Initial state of the particles in the room with no obstacles.

Figure 3.4 shows the initial state of the particles inside the "room", all of them start at some random point in the left side of the room and with an initial velocity going to the right and additionally a drag force is applied to all the particles so that they are pulled to the exit. 50 particles were used in the simulation.

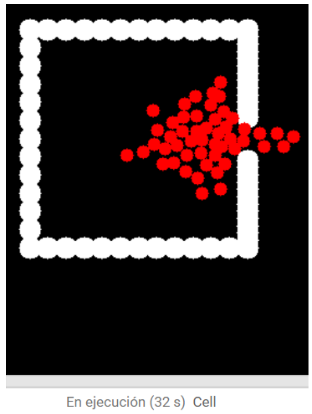

Figure 3.5: State of the particles in the room with no obstacles after 32s.

Figure 3.5 shows an intermediate state of the particles inside the room that has no obstacles near the exit. It can be seen that particles start exiting the room. and that many of them are starting to accumulate at the exit.

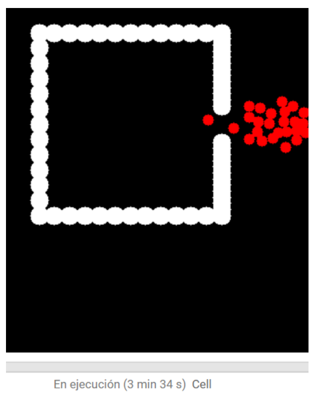

Figure 3.6: Final state of the particles in the room with no obstacles after 3 mins. 32 s.

Figure 3.6 shows the final state of the particles system where all of the particles are already outside the room (only one is in its way out). In the inferior part can be seen the execution time, which took 3 minutes and 32 seconds for all of the particles to go out.

Now, the program will be run again but this time an obstacle will be placed in front of the exit, to see if there is any improvement compared with the execution without an obstacle.

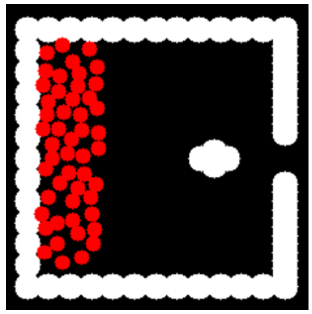

Figure 3.7: Initial state of the particles in the room with obstacles.

Figure 3.7 shows the initial state of the particles inside the room, the initial position is the same as the one with no obstacles.

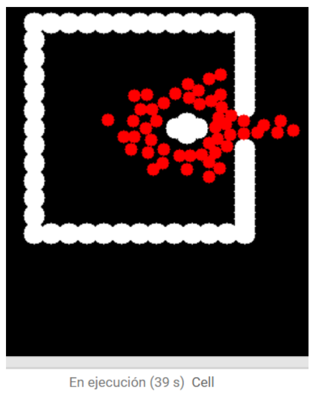

Figure 3.8: State of the particles in the room with no obstacles after 39s.

In figure 3.8 we can see the particle system after 39 seconds, we can observer how the first particles are starting to exit the room, however, it took a little bit longer for the first particles to start exiting.

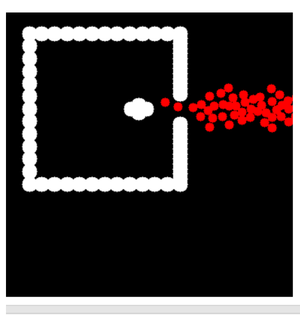

Figure 3.9: Final state of the particles in the room with no obstacles after 2 mins. 27 s.

Figure 3.9 shows the final state of the particles, where all of them are either already out of the room or in its way out. The difference in time compared to the simulation with no wall is quite clear, there is actually more than a minute of difference.

So, it was proved, that the theory of placing an obstacle in front of an exit to speed up evacuation is true for particle systems. There was more than a minute of difference between both ways, being the one with the obstacle in front of the exit, the fastest.

During the simulation, it was observed that even when the particles in the simulation with no obstacle started to exit faster, the accumulation of particles in the exit started to slow down the process since they started to block each other. In the other hand, in the simulation with the obstacle, although it took more time for the first particles to exit the room, the obstacle reduced the accumulation of particles in the exit, which helped to speed up in the evacuation of the particles.

4 Conclusions

During the development of this practice resulted in a better comprehension of particle systems and how the ways of communication affect the performance of the particles.

It was concluded that particle systems are very helpful to model real life behaviors, such as a ”mexican wave” or simulate a crowd trying to exit a room. It was observed that with simple behaviors defined for each particle, one can create complex behaviors of the particles as a whole.

It was also concluded that there are many ways of communication and influence of particles and they can completely change the way particle swarms behave. In this case, changing the way the particles are influenced changed totally the way they behaved. Also, it was observed why particle systems are so used in computer graphics, they are helpful to create very attractive visual effects.

5 Bibliography

- **Kennedy J., C. Eberhart R. (2001) Swarm Intelligence. San Diego, Ca, USA. Academic Press, Morgan Kaufmann.**
- **Krause J., Ruxton G., Krause S. (2010). Swarm intelligence in animals and humans. Trends in Ecology & Evolution.**
- **William Reeves. (1983) Particle Systems—A Technique for Modeling a Class of Fuzzy Objects. ACM Transactions on Graphics.**
- **Finlo Rohrer. (2010) Who invented the Mexican Wave?. BBC News.**
- **Gueron, Shay, Gurevich, Liron, Blum (1997) Cilia internal mechanism and metachronal coordination as the result of hydrodynamical coupling. Proc. Natl. Acad. Sci. USA.**
- **A. Garcimartín (2018) Redefining the role of obstacles in pedestrian evacuation. New J. Phys.**