

# Práctica 2: Sistemas de Partículas



## Actividad 1

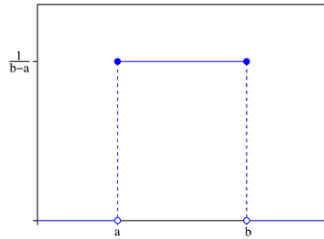
Los sistemas de partículas resultan muy útiles para representar muchos fenómenos dentro de un juego. Casi todos funcionan de la misma manera y solo cambian los parámetros que los configuran. Con ellos podemos conseguir efectos muy distintos.

Para familiarizarnos con ellos vamos a **hacer un sistema de partículas simple que tenga emisores de partículas que generan partículas desde un punto cada *frame***. En otras palabras: vamos a simular una especie de fuente de partículas. Dicha fuente de partículas la implementaremos como una clase que tendrá las siguientes características:

- El generador de partículas “fuente” emite las partículas de media con una determinada velocidad y dirección en un lugar determinado (la posición de la fuente no tiene apenas variabilidad).
- El sistema de partículas almacenará cada partícula que se produzca y se encargará de eliminarlas a su debido tiempo. Veremos dos condiciones:
  - Tiempo de vida.
  - La partícula abandona el espacio de acción.
- No obstante, la velocidad precisa de cada partícula seguirá una determinada distribución aleatoria.
  - Como solución simple podemos usar una distribución uniforme de velocidades de salida. Una primera aproximación podría ser una distribución tridimensional uniforme, donde deberemos especificar los valores máximos y mínimos en cada dirección (véase la Figura 1a).
  - No obstante, la distribución más común es la campana Gaussiana, también llamada **distribución normal**. Esta distribución viene determinada por su media y su varianza. Cuanto mayor sea la varianza, menos representativa es la media y la distribución se aplana (ver Figura 1b). Para utilizar una distribución de este tipo podemos usar las cabeceras estándar de C++. (véase ejemplo en <https://en.cppreference.com/w/cpp/numeric/random>). Como primera aproximación podemos usar una distribución normal independiente para cada componente de la velocidad, sus varianzas pueden ser distintas.
  - De forma general, la distribución sería una gaussiana multivariable tridimensional, que está determinada por un vector como media y una matriz de covarianzas. Las covarianzas indicarían que los valores de las componentes no son independientes, si no que están ligados. Esta distribución está fuera del alcance de la asignatura, pero el alumno interesado puede consultarnos en tutoría. Una posible implementación: <https://github.com/beniz/eigenmvm>
- **Nuestro sistema de partículas se verá afectado por la gravedad**, que será la misma para todas las partículas. Haremos que dicha gravedad aporte una aceleración de valor -10.0 en la dirección Y.

La función de densidad uniforme tiene como función de densidad de probabilidad:

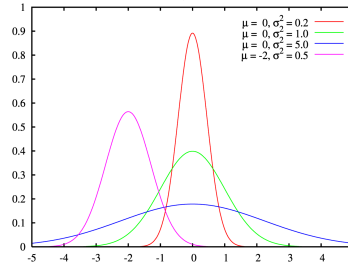
$$f(x) = \begin{cases} \frac{1}{b-a}, & a < x < b \\ 0, & \text{e. o. c.} \end{cases}$$



**Figura 1a.** Función de densidad de probabilidad uniforme

La función de densidad Gaussiana tiene como función de densidad de probabilidad:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



**Figura 1b.** Función de densidad de probabilidad gaussiana.  $\mu$  es la media.  $\sigma$  es la desviación típica.

Mediante emisores de partículas, podemos simular muchos efectos distintos. Hasta ahora, hemos hecho un emisor que parte de un punto determinado (la posición no tendría varianza alguna) con una velocidad aleatoria, con la distribución seleccionada. Esto modelaría el comportamiento una manguera, etc. Por otro lado, si quisiéramos emular el efecto de niebla, tendríamos partículas que parten de posiciones muy variables con velocidades pequeñas y también aleatorias. Las partículas usadas deberían ser muy pequeñas y transparentes.

**Configura al menos 3 emisores de partículas que generen distintos efectos.**

## Actividad opcional

En esta actividad el objetivo es diseñar un sistema para crear unos fuegos artificiales. Un fuego artificial se caracteriza por ser un sistema jerárquico, una única partícula inicial, a su “muerte” genera más partículas, que a su vez al explotar (“morir”) generan más. Además, las partículas “hijas” han de heredar algunas características básicas de la partícula progenitora (como la posición en la que nacen y su velocidad base).

**Genera un sistema jerárquico para simular unos fuegos artificiales (o bien otro sistema jerárquico que se te ocurra a ti).**