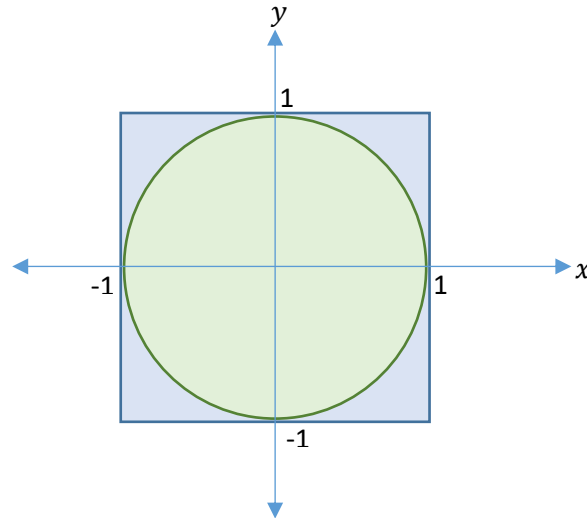


VAMOS A JUGAR

El método de Monte Carlos es un método ingenioso para encontrar aproximaciones de soluciones a problemas que es difícil encontrar su solución exacta. Un ejemplo muy común que utiliza esta técnica es aproximar el valor de π a través de la siguiente simulación.

Simulemos que estamos jugando en un tablero de dardos de radio 1, el cual está ubicado dentro de un cuadrado. Es fácil generar coordenadas con números aleatorios x y y con valores entre -1 y 1.



Si el punto generado cae dentro del círculo lo contamos como un acierto. Eso sucede cuando $x^2 + y^2 \leq 1$. Debido a que nuestros tiros son completamente al azar, esperamos que la tasa aciertos/intentos se aproxime a la relación que hay entre el área del círculo y del cuadrado la cual es $\pi/4$. Por lo tanto, para aproximar π debe ser $4 * \text{aciertos} / \text{intentos}$. Este método aproxima π usando simple aritmética.

Para generar un valor aleatorio decimal entre -1 y 1 puedes hacerlo de la siguiente manera:

```
double r = Math.random( ); //Genera un número aleatorio r tal que 0 <= r < 1
double x = -1 + 2 * r; // Hace que x esté en el rango -1 <= x < 1.
```

Genera la clase MonteCarlo, la cual tendrá el método estatico llamado aproximarPI, el cual regresa un double que representa la aproximación de π y recibe como parámetro un int, el cual representa el número de intentos que va a realizar. Aquellos intentos que “caigan” dentro del círculo se consideran aciertos.

Puedes probar tu programa corriendo varias veces el método aproximarPI con diferentes valores y luego imprimiendo el resultado y verás que el método regresa valores cercanos a π . Además. Mientras más grande sea el número de intentos el método tiende a ser más preciso. En mi caso al correrlo con $n = 100$, ya daba número cercanos pero al correrlo con $n = 10'000,000$ una vez me dio 3.1416104.