¡HOLA!

En principio, muchas gracias por tu tiempo.

La siguiente es una prueba de conocimientos que nos permitirá conocer un poco más sobre tus habilidades y que esperamos te resulte entretenida.

Lee detenidamente las siguientes indicaciones, por favor.

- Esta prueba tiene una duración de 01 día.
- No consideramos las horas, por lo que tienes hasta las 23:59 del día siguiente para resolver las preguntas.
- Queremos que estés cómodo: puedes usar cualquier lenguaje, incluyendo pseudocódigo, a menos que explícitamente se indique lo contrario.
- Si existe algún detalle que no te ha quedado claro, siéntete en la libertad de realizar las preguntas correspondientes a la dirección abarua@leapgs.com, o asume, por tu cuenta, cualquier cosa que creas conveniente para la resolución de la pregunta. Indícanos, eso sí, qué cosas has asumido para tenerlo en cuenta a la hora de revisar tu pregunta.

Muy bien. ¿Cómo nos entregas tus soluciones?

- Crea un repositorio público en Bitbucket/Github/GitLab. En cuanto lo hagas, envíanos un correo a la dirección abarua@leapgs.com, indicando el URL de tu repositorio. A través de él nosotros evaluaremos tu examen.
- Utiliza **GIT**.
- Sube continuamente el progreso de tu examen a este repositorio.

PREGUNTAS

1. ¿Qué hace el siguiente algoritmo?

```
bool xyz (int n)
{
    float i = math.sqrt(n);
    int  j  =  math.ceil(i);
    int k = 2;
    int x = k;
    while(x <= j)
    {
        if(!(n % x)) return false;
        else x++;
    }
    return true;
}
```

2. Dados 3 rectángulos:

   Rectángulo *A*, en posición central (xa, ya), wa: width y ha: height;

   Rectángulo *B*, en posición central (xb, yb), wb: width y hb: height;

   Rectángulo *C*, en posición central (xc, yc), wc: width y hc: height;

   Implementar un método que verifique colisiones entre los tres rectángulos.

3. Dado un punto P(x,y), escribir un algoritmo que muestre en pantalla 'n' puntos posicionados en una semicircunferencia de radio 'r' y cuyo centro es P.

4. [HTML5] Utilizando una librearía HTML5 de su preferencia (Que soporte WebGL con fallback a Canvas) desarrollar un pequeño juego top down en el que se deba llegar del punto A al punto B, en el escenario deberá haber objetos que te maten.

5. [Unity] Crea un proyecto en Unity e implementa un Animator Controller para un personaje que tiene los siguientes estados: Idle, Walking, Running, Jumping y Shooting. Incluye las transiciones y los parámetros.

6. [Caso] En un juego móvil desarrollado en Leap, tuvimos el siguiente problema:

   El juego instanciaba frecuentemente enemigos que el jugador derrotaba rápidamente. Al ser derrotados, los enemigos eran destruidos de la escena. Sin embargo, la creación y destrucción de objetos en tiempo de ejecución impactaba seriamente el rendimiento del juego.
   Explique según su criterio, la mejor solución para este problema.

Eso es todo. Y no olvides escribir a abarua@leapgs.com para indicarnos el URL de tu repositorio y para que resolvamos cualquier duda que tengas.

Mucha suerte.