

Sesión 2 · Martes 27 de octubre de 2026

Programación de videojuegos

Organiza:

IRIO Incubadora de Realidades Inmersivas de Almendralejo by FUNDECYTPCTEX un espacio para innovar

Colaboran:

VIRAL STUDIOS

Ayuntamiento Almendralejo

Financian:

Cofinanciado por la Unión Europea

Ministerio de Hacienda

Fondos Europeos

Junta de Extremadura
Consejería de Industria, Energía, Ciencia y Territorio



Info

Seis partes, una pieza nueva en cada una.
Al final, un juego que se juega.


CRISTALES 0
VIDAS 3
El juego que vas a construir, con pulido.

El mapa de la sesión

Seis partes. En cada una, el juego gana una pieza nueva.

Parte 1
El motor por dentro

Parte 2
Tu primer script

Parte 3
Física y colisiones

Parte 4
Prefabs y aparición

Parte 5
Un juego de verdad

Parte 6
Programar en una jam

Una pieza, Play y comprobar. Seis veces.

IRIO by FUNDECYTPCTEX
Cofinanciado por la Unión Europea



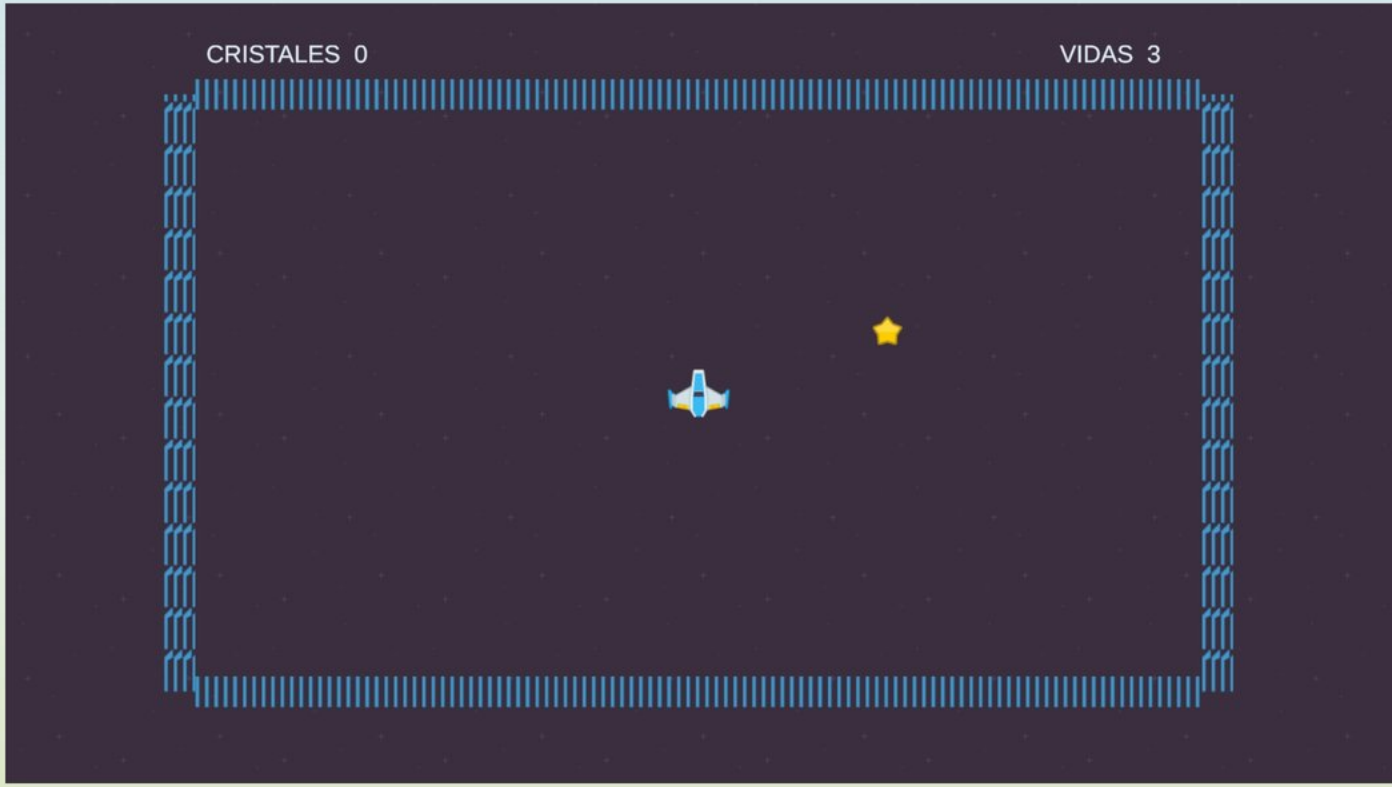
Info

Un motor es todo lo que no quieres programar tú.

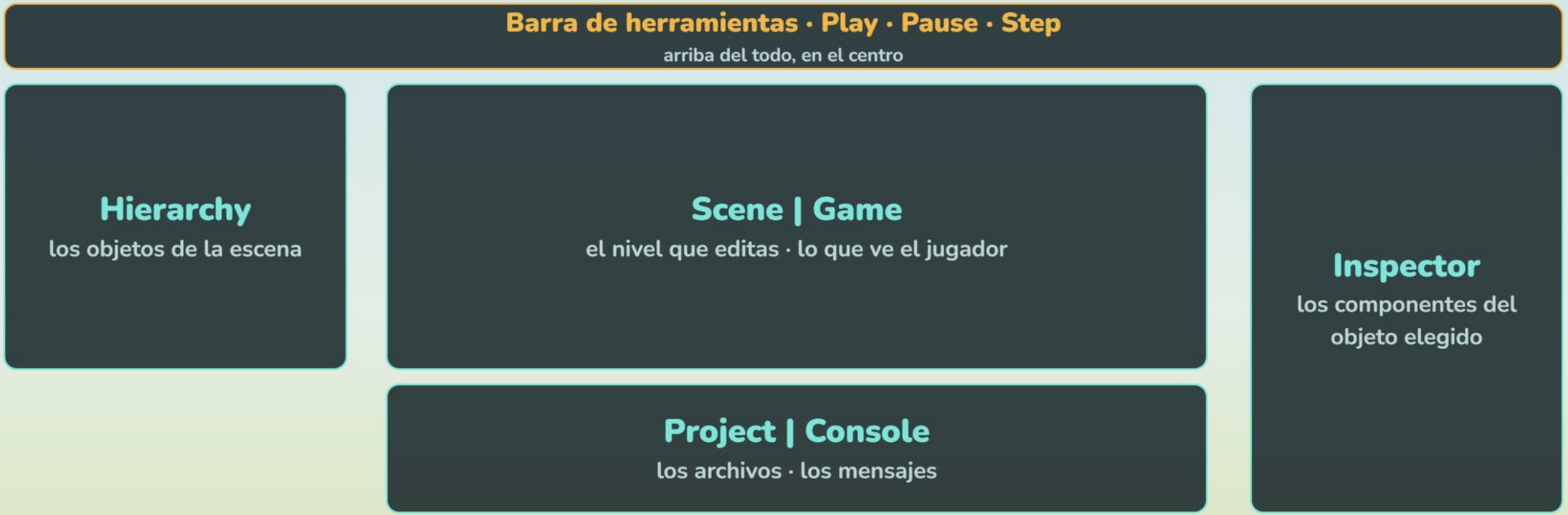
IRIO by FUNDECYTPCTEX
Cofinanciado por la Unión Europea

Lo que hace el motor por ti

Dibujar · Física y choques · Teclado y mando · Sonido · Exportar a Windows, móvil o consola



El editor de Unity, de un vistazo



Esquema de la distribución por defecto (Window › Layouts › Default). No es una captura.

Ver el juego: Scene y Game

Scene · escena

Tu mesa de trabajo. Aquí colocas, mueves, giras y escalas los objetos del nivel. Sólo la ve quien hace el juego.

W mover · E rotar · R escalar · F centra lo seleccionado · botón 2D

Game · juego

Lo que ve el jugador: la imagen de la cámara. Aquí se juega al pulsar Play.

Arriba: proporción de pantalla y zoom; puede maximizarse al jugar

TO DO

Captura de Carlos en el ensayo: ventanas Scene y Game, una al lado de la otra

Elegir y ajustar: Hierarchy e Inspector

Hierarchy · jerarquía

La lista de objetos de la escena abierta. Un objeto dentro de otro es su hijo: se mueve, gira y escala con él.

Clic para elegir · arrastrar para ordenar · clic derecho para crear

Inspector · inspector

Todo sobre lo que tengas elegido: sus componentes y sus valores. Aquí se cambian números sin programar.

También enseña cómo se importa una imagen o un sonido

TO DO

Captura de Carlos en el ensayo: Hierarchy con Main Camera elegida e Inspector con su Transform

Archivos y mensajes: Project y Console

Project · proyecto

Los archivos del juego: imágenes, sonidos, scripts, escenas y prefabs. Es la carpeta Assets de tu disco.

Arrastrar un archivo a la escena crea un objeto con él

Console · consola

Los mensajes del juego y del motor: información, avisos en amarillo y errores en rojo.

Doble clic en un error lleva a su línea · Clear · Collapse · Error Pause

TO DO

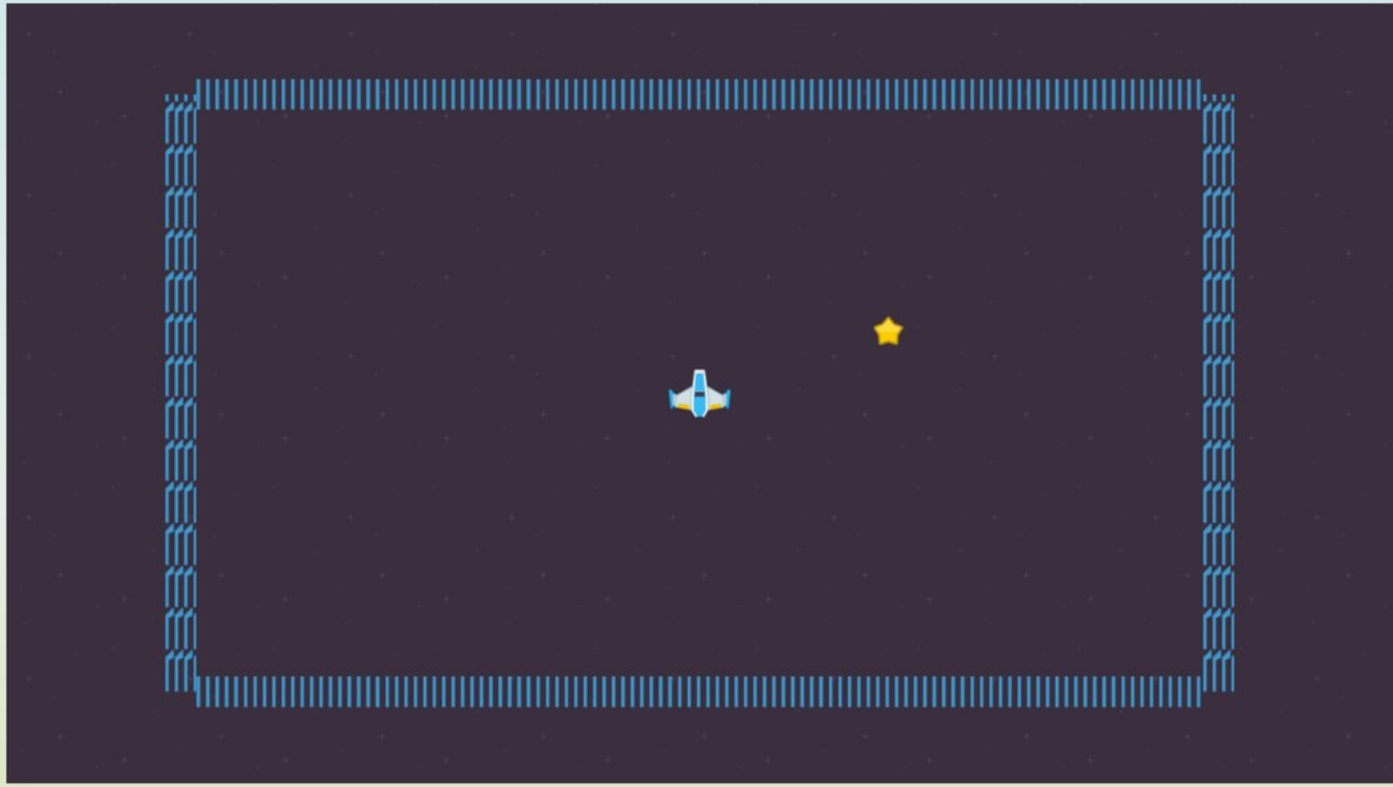
Captura de Carlos en el ensayo: Project con la carpeta Assets y Console con un mensaje

La barra de arriba y el modo Play

- **Play · Pause · Step:** arrancar, pausar y avanzar un solo fotograma (Ctrl+P)
- **Modo Play:** el juego funciona de verdad; lo que cambies se pierde al salir
- **Menú Window:** abre cualquier otra ventana (Package Manager, Animation, Lighting…)
- **Layouts:** guarda tu distribución de ventanas favorita

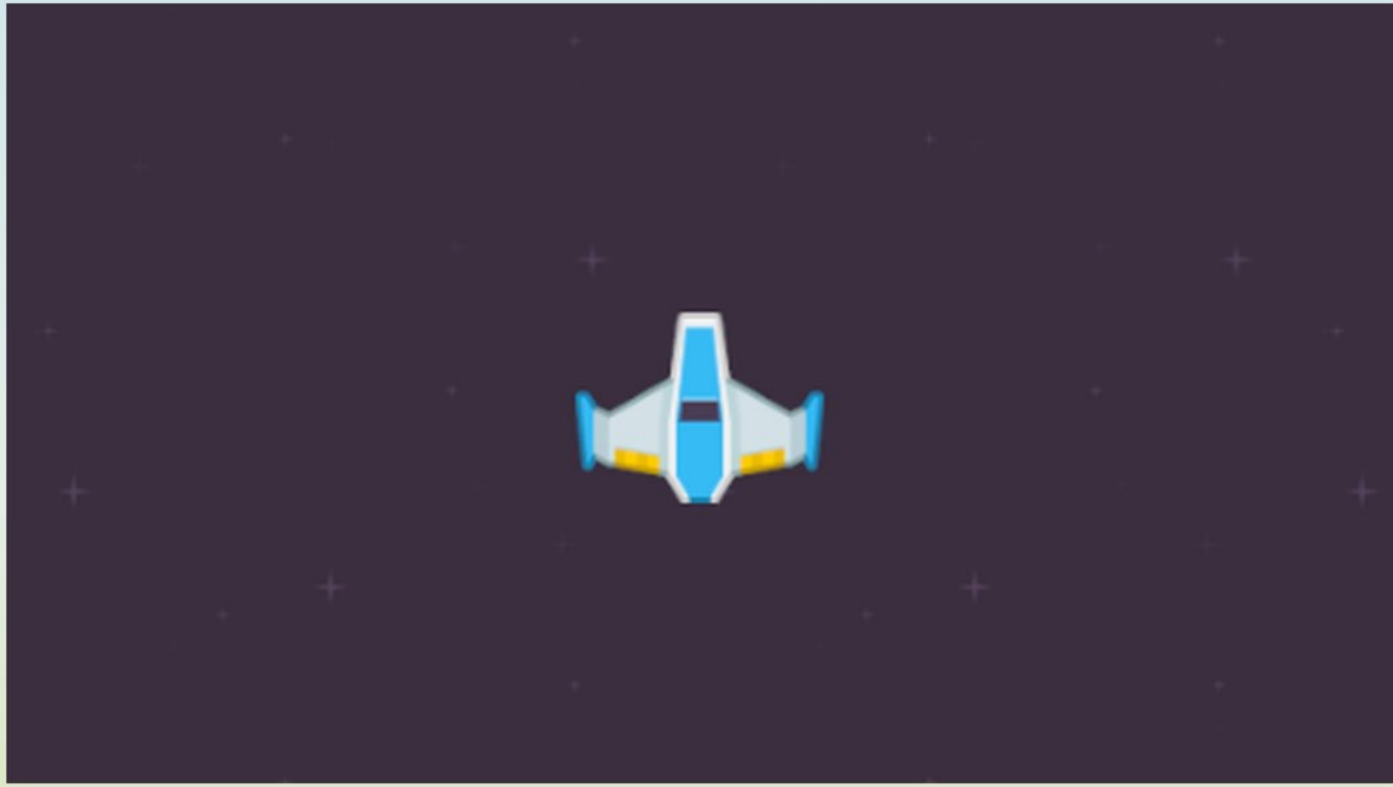
La escena: un nivel

Un archivo .unity con todos los objetos de un nivel o una pantalla: el menú, el nivel 1, la derrota. Nuestro juego cabe en una sola escena.



El GameObject: una cosa del juego

La nave, cada muro, el cristal, la cámara. Tiene nombre, etiqueta, capa y una casilla para activarlo. Vacío, no hace nada.



El componente: una capacidad

Transform: dónde está, cuánto está girado y su tamaño. Todos lo tienen.

SpriteRenderer: lo dibuja con una imagen

BoxCollider2D: le da una forma con la que chocar

Rigidbody2D: le da cuerpo físico (parte 3)

Script: una capacidad que escribes tú (parte 2)

Ajusta tu editor: Edit › Preferences

External Tools: elige VS Code como editor de scripts

Colors › Playmode tint: tiñe el editor al jugar para que no se te olvide que estás en Play

Son ajustes tuyos, de tu ordenador: valen para todos tus proyectos

Ajusta tu proyecto: Edit › Project Settings

Player: nombre del juego, empresa e icono

Player › Active Input Handling: el sistema de teclado y mando (hoy, Input System)

Physics 2D: la gravedad del mundo

Tags and Layers: las etiquetas, como Crystal y Meteor

File › Build Profiles: para qué plataforma se exporta el juego

Al terminar la parte 1

Reto extra: dos obstáculos fijos dentro de la arena

Punto de control 1: extrejam.com/unity

Hasta la parte 3 sólo se verán: aún no paran la nave



Info

Un script es una pieza más que le encajas a un objeto.

IRIO by FUNDECYTPCTEX
Cofinanciado por la Unión Europea

Update: en cada fotograma

MonoBehaviour: lo que convierte un script en componente

Update(): se ejecuta en cada fotograma, decenas de veces por segundo

Start(): una sola vez, al empezar

Tres palabras y ya

Variable: un dato con nombre · float speed = 5

Tipo: qué clase de dato · float, Vector2

Método: instrucciones agrupadas con nombre · Move()

[SerializeField]: el valor aparece en el Inspector y se cambia sin tocar código

Moverse es sumar un poco en cada fotograma

PlayerController.cs · PC2 — C# · Unity 6.3

```
1  public void Move(Vector2 direction)
2  {
3      // deltaTime mantiene la velocidad al cambiar los fotogramas por segundo.
4      transform.position += (Vector3)direction.normalized * speed * Time.deltaTime;
5  }
```

Línea 4 · Time.deltaTime = segundos desde el fotograma anterior

Al terminar la parte 2

Reto extra: velocidad máxima ajustable en el Inspector

Punto de control 2: extrejam.com/unity

¿Por qué la nave atraviesa los muros?



Info

Antes colocábamos la nave. Ahora vamos a empujarla.

IRIO by FUNDECYTPCTEX
Cofinanciado por la Unión Europea

Colocar frente a empujar

Mover el Transform

Teletransporte en pasos pequeños
La física no se entera
Atraviesa muros

Rigidbody2D + velocidad

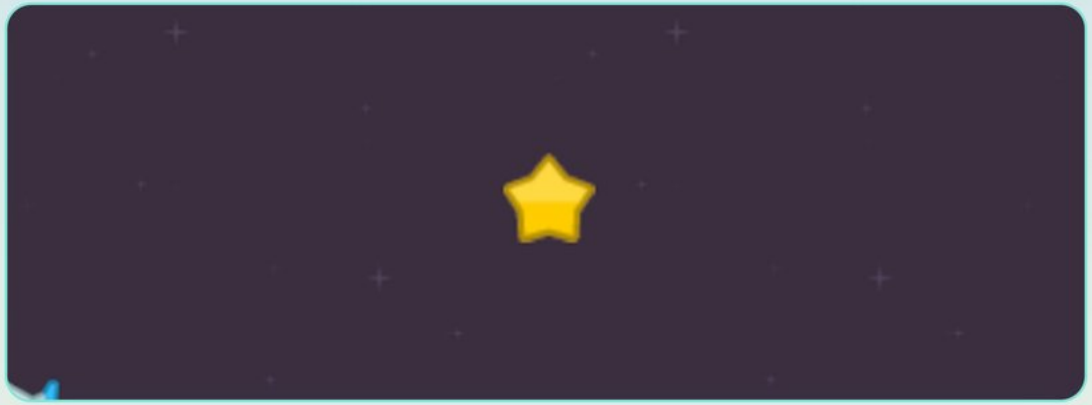
La física mueve la nave
El collider da la forma
Los muros la paran

El muro y el cristal



Collider normal

Detiene



Collider trigger

Avisa y deja pasar

Pedir una velocidad, no una posición

PlayerController.cs · PC3 — C# · Unity 6.3

```
1  public void Move(Vector2 direction)
2  {
3      body.linearVelocity = direction.normalized * speed;
4  }
```

Sin deltaTime: linearVelocity ya está en unidades por segundo

Recoger el cristal

PlayerController.cs · PC3 — C# · Unity 6.3

```
1  private void OnTriggerEnter2D(Collider2D other)
2  {
3      if (other.CompareTag("Crystal"))
4      {
5          score++;
6          Debug.Log("Cristales: " + score);
7          other.GetComponent<Crystal>().Relocate();
8      }
9  }
```

Una condición elige qué hacer según lo que tocas

Al terminar la parte 3

Reto extra: que el cristal no reaparezca pegado a la nave

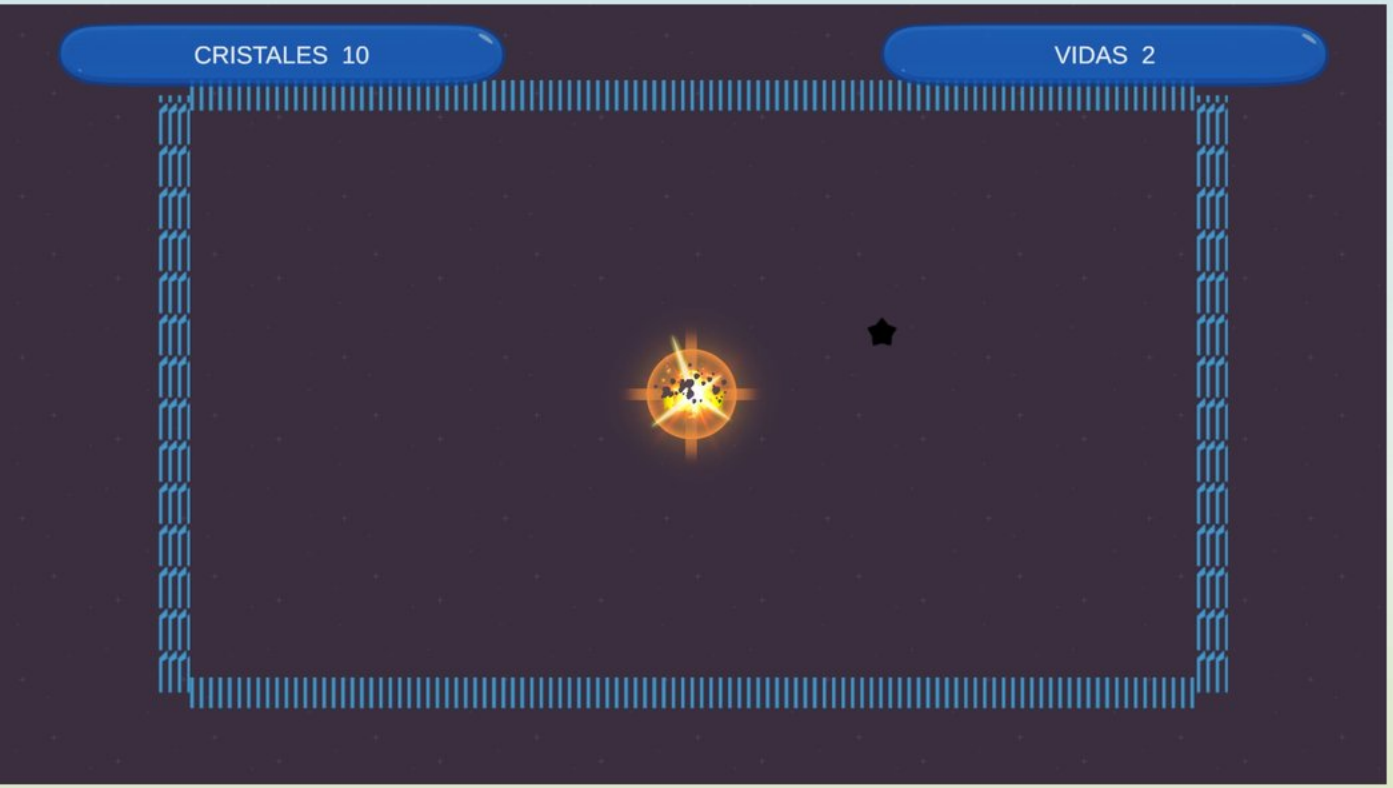
Punto de control 3: extrejam.com/unity



Info

Un meteorito está bien. Doscientos iguales, sólo si tienes un molde.

INIA by FUNDECYTPCTEX
Cofinanciado por la Unión Europea



Prefab = molde · Instancia = copia

Instantiate crea una copia nueva. Destroy la elimina cuando sale de la arena o choca.

El meteorito sólo sabe lanzarse

Meteor.cs · PC4 — C# · Unity 6.3

```
1  public void Launch(Vector2 direction)
2  {
3      // La dirección se fija al nacer: el meteorito no persigue después.
4      body.linearVelocity = direction * speed;
5  }
```

direction es un parámetro: el dato llega desde fuera

Crear la copia y apuntar

MeteorSpawner.cs · PC4 — C# · Unity 6.3

```
1  GameObject meteor = Instantiate(meteorPrefab, position, Quaternion.identity);
2  // El prefab no puede guardar una referencia a la nave de esta escena.
3  meteor.GetComponent<Meteor>().Launch(
4      ((Vector2)player.position - position).normalized);
```

Destino menos origen = dirección hacia la nave

Por qué el molde no apunta a la nave

El prefab vive en Project, no en una escena

El spawner sí está en la escena: conoce la nave y le pasa la dirección

Al terminar la parte 4

Reto extra: tres tipos de meteorito: tamaño, forma y rapidez distintos

Punto de control 4: extrejam.com/unity

PC4 trae además la interfaz preparada para la parte 5



Info

Tenemos piezas. Falta alguien que lleve la cuenta.

IRIO by FUNDECYTPCTEX
Cofinanciado por la Unión Europea

El GameManager es el árbitro

- Puntos
- Vidas
- **Dificultad:** cada cuánto salen meteoritos
- Fin de partida

Toda la dificultad, en una línea

GameManager.cs · PC5 — C# · Unity 6.3

```
1  public float GetSpawnInterval()
2  {
3      return Mathf.Max(0.4f, 1.5f - score * 0.05f);
4  }
```

0 cristales: 1,5 s · 10: 1 s · 22 o más: 0,4 s

Canvas + TextMeshPro

Textos de puntos y vidas, título y derrota.

Un botón no hace nada: llama a un método.



Volver a jugar = volver a cargar el nivel

GameManager.cs · PC5 — C# · Unity 6.3

```
1  public void Restart()
2  {
3      // La pausa pertenece a la partida que termina, no al proceso de carga.
4      Time.timeScale = 1;
5      SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex);
6  }
```

timeScale = 0 pausa el juego; 1 lo devuelve a velocidad normal

Al terminar la parte 5

Reto extra: un escudo temporal que aparece de vez en cuando

Punto de control 5: extrejam.com/unity

Reserva de 30 a 60 minutos: es el más largo



Info

Cuarenta y ocho horas. Estos atajos son legales.

INRIA by FUNDECYTPCTEX
Cofinanciado por la Unión Europea



Info

Pide poco. Lee antes de aplicar. Compruébalo jugando.

El encargo a la IA

«Haz que las vidas iniciales se puedan configurar en el Inspector, entre 1 y 5.»

1. Leer la respuesta antes de aplicarla · 2. Rechazar lo que no has pedido · 3. Probar con 1, con 5 y reiniciando

Qué puede compartir un equipo



Kenney · CC0

Se usa, se modifica
y se comparte sin pedir permiso



Asset Store · licencia estándar

Se usa en tu juego terminado
No se reparte el proyecto con el paquete

Atajos legítimos en una jam

- Assets gratuitos, diciendo de dónde salen
- Tu propia plantilla de proyecto, preparada en casa
- Asset Store, respetando su licencia

Un script, una responsabilidad

`PlayerController`	`Crystal`	`Meteor`	`MeteorSpawner`	`GameManager`
mueve y avisa	se recoloca	vuela recto	lanza	arbitra

Por eso hemos podido añadir piezas sin romper lo anterior.

Para casa

- Cinco retos con pistas, separadas de la solución
- Elige uno, haz un cambio y cuéntanos qué has cambiado en Discord

Lo que tienes en extrejam.com/unity

- Cinco puntos de control
- El juego terminado con pulido (ejecutable)
- Guía con código comentado y cinco retos
- PDF de estas diapositivas



Info

Martes 3 de noviembre: arte para videojuegos, con Nuno.

Sesión completada

Gracias

Próxima sesión: martes 3 de noviembre · Arte, con Nuno

Puntos de control, guía y material en extrejam.com/unity





Organiza: IRIO Incubadora de Realidades Inmersivas de Almendralejo by FUNDECYTPCTEX un espacio para innovar

Colaboran: Viral Studios · Ayuntamiento de Almendralejo

Financian: Cofinanciado por la Unión Europea · Ministerio de Hacienda · Fondos Europeos · Junta de Extremadura, Consejería de Industria, Energía, Ciencia y Territorio

Anexo · Chuleta del editor de Unity

Seis ventanas y sus atajos

Scene

La escena que editas

W mover · E rotar · R escalar · F encuadrar el objeto · 2 vista 2D

Game

Lo que ve la cámara al jugar

Ctrl+P Play · Ctrl+Mayús+P pausa · Ctrl+Alt+P avanzar un fotograma

Hierarchy

Los objetos de la escena

Ctrl+D duplicar · F2 renombrar · Supr borrar · arrastrar para emparentar

Inspector

Los componentes del objeto

Add Component · ⋮ › Reset · el candado fija el objeto

Project

Los archivos del juego

Crear › C# Script · arrastrar a la escena · prefab = objeto arrastrado aquí

Console

Mensajes y errores

Clear · Collapse · Error Pause · doble clic abre la línea

Edit › Preferences: cómo funciona el editor en tu ordenador · Edit › Project Settings: cómo es el juego (física, entrada, calidad) · File › Build Profiles: exportar · Window › Layouts › Default: recuperar las ventanas perdidas.

INIA by FUNDECYTPCTEX · Cofinanciado por la Unión Europea