

Documentación de la función `create()` en LibGDX

¿Qué es `create()`?

La función `create()` es lo primero que se ejecuta cuando arrancas el juego.

Se llama **una sola vez**, al inicio, y es donde preparas todo lo que necesitarás durante la partida.

Piensa en ello como encender la consola: antes de jugar, se cargan gráficos, música y controles.

Si algo no está preparado aquí, el juego no podrá usarlo y fallará.

¿Qué tengo que hacer en `create()`?

Debes inicializar tres tipos de elementos:

1 Gráficos (lo que se ve en pantalla)

- **SpriteBatch** → el “pincel” del juego, imprescindible para dibujar.
- **Texture** → las imágenes (serpiente, comida, fondo).
- **ShapeRenderer** → para dibujar formas básicas.
- **BitmapFont** → fuentes de texto para puntuaciones y mensajes.

2 Estado inicial del juego

- Posición inicial de la serpiente.
- Dirección de movimiento.
- Velocidad de desplazamiento.
- Puntuación en 0.
- Generar la primera comida en un lugar aleatorio.

3 Controles y utilidades

- **InputProcessor** → detecta teclas o toques en pantalla.
 - **AssetManager** → opcional, organiza la carga de recursos.
-

Ejemplo de `create()` para Snake

```
@Override
public void create() {
    // Objeto para dibujar
    batch = new SpriteBatch();

    // Cargo imágenes
    texturaSerpiente = new Texture("snake.png");
    texturaComida = new Texture("food.png");
    texturaFondo = new Texture("fondo.png");

    // Creo la serpiente con 3 segmentos
    cuerpo = new Array<>();
    cuerpo.add(new Vector2(5, 5)); // Cabeza
    cuerpo.add(new Vector2(4, 5)); // Medio
    cuerpo.add(new Vector2(3, 5)); // Cola

    // Dirección inicial
    direccion = Direccion.DERECHA;

    // Velocidad
    velocidad = 0.15f;
    tiempo = 0f;

    // Puntos iniciales
    puntos = 0;

    // Genero comida
    generarComida();

    // Activo controles
    Gdx.input.setInputProcessor(new ControlesTeclado());
}
```

Errores típicos y soluciones

- X El juego se cierra → No has creado `batch` o alguna textura.
- X No se ven imágenes → Ruta incorrecta, revisa carpeta `assets/`.
- X Teclas no responden → Falta registrar `InputProcessor`.
- X Serpiente fuera de pantalla → Coordenadas iniciales mal definidas.
- X El juego se ralentiza → Estás cargando texturas en `render()` en vez de `create()`.

Organización recomendada

@Override

```
public void create() {  
    inicializarGraficos();  
    inicializarJuego();  
    cargarImagenes();  
}
```

```
private void inicializarGraficos() {  
    batch = new SpriteBatch();  
}
```

```
private void inicializarJuego() {  
    cuerpo = new Array<>();  
    cuerpo.add(new Vector2(5, 5));  
    direccion = Direccion.DERECHA;  
    puntos = 0;  
}
```

```
private void cargarImagenes() {  
    texturaSerpiente = new Texture("snake.png");  
    texturaComida = new Texture("food.png");  
}
```

Liberar memoria en **dispose()**

@Override

```
public void dispose() {  
    batch.dispose();  
    texturaSerpiente.dispose();  
    texturaComida.dispose();  
    texturaFondo.dispose();  
}
```

Ciclo de vida en LibGDX

1. **create()** → Preparar recursos (una vez).
2. **resize()** → Ajustar ventana.
3. **render()** → Actualizar y dibujar (60 fps).
4. **pause()** → Juego minimizado.

5. **resume()** → Volver al juego.
 6. **dispose()** → Liberar memoria (una vez).
-

Checklist antes de terminar **create()**

- [] SpriteBatch creado
 - [] Imágenes cargadas
 - [] Serpiente con al menos 3 segmentos
 - [] Dirección inicial definida
 - [] Velocidad configurada
 - [] Primera comida generada
 - [] Puntuación en 0
 - [] Controles registrados
 - [] Archivos en carpeta **assets/**
 - [] Plan de liberación en **dispose()**
-

Resumen rápido

create() es el **punto de partida** del juego:

- Cargas imágenes
- Creas objetos gráficos
- Defines estado inicial
- Generas comida
- Activas controles

Si algo falta aquí, el juego no funciona.

Todo lo que se crea aquí debe liberarse en **dispose()**.
