



# Requisitos para la entrega final

## Introducción a la Programación para Videojuegos

Para aprobar este trayecto de formación debe entregar un videojuego simple usando las herramientas dadas en clase.

A continuación se enumeran una serie de pautas mínimas que debe cumplir para proporcionar cierto estándar de calidad y atender a cuestiones propias de la experiencia del usuario.

## Pautas generales para la aprobación

1. **El juego no debe crashear ni presentar bugs mayores** (ejemplo: errores que bloqueen el avance del jugador).

✗ Error común: no testear lo suficiente o dejar el testeo para último momento.

✓ Pruebe el juego lo suficiente mientras lo va desarrollando y dedique un tiempo a testearlo una vez finalizado. Es preferible un juego muy simple pero pulido que uno complejo donde todo funciona a medias.

2. **El juego debe contar con un menú que permita volver a jugarlo o salir. Mientras se esté jugando se debe poder salir del gameplay a este menú.**

✗ Errores comunes:

- No proporcionar una forma de rejugar (el juego se termina la primera vez que se pierde o gana).
- No se puede salir (no hay botón o forma clara de cerrar el juego).
- El juego sale abruptamente desde el gameplay al tocar la tecla ESC (no vuelve al menú, ni pide confirmación de salir).

3. **El juego debe tener un nombre** (puede no tener logo y el nombre ser visible sólo como un texto. Incluso, este nombre puede no aparecer en el menú ni en las escenas del juego, pero sí debería ser el nombre del ejecutable y asignarse en las propiedades del proyecto)
4. **En algún lugar (ya sea en el menú principal o en otra pantalla) deben mostrarse los controles e instrucciones de juego** (no hace falta que los controles sean configurables).

✗ Errores comunes:

- No se le indica al usuario con qué controles maneja su personaje.
  - No se indica cómo salir de una pantalla para pasar a otra y el usuario debe probar tocar botones al azar para ver si logra salir.
  - Los controles son poco intuitivos (para comenzar a jugar se usa tecla P).
  - Los controles que determinan una misma acción cambian de escena en escena siendo inconsistentes (para volver al menú, se debe presionar ESC mientras se está en los créditos y M mientras se está en el gameplay).
5. **El juego debe tener una condición de derrota/victoria:** se debe poder perder y/o se debe poder ganar (aunque ganarlo signifique no perder).
  6. **En algún lugar (ya sea en el menú principal o en otra pantalla) deben incluirse los créditos.** Esto es:
    - a. Su nombre completo
    - b. Los nombres de quienes hayan creado los assets si no los hizo usted (seguir leyendo próximos puntos)
    - c. Opcional: un listado de todas las herramientas/software que utilizó
  7. **Los elementos gráficos y sonoros deben tener licencias válidas para su uso.** Pueden ser:
    - a. Creados por usted usando herramientas tradicionales (Paint, Photoshop, etc.) u otras simples como por ejemplo:
      - Arte 2D: <http://www.piskelapp.com/>
      - Fonts: <http://glyphrstudio.com/>, <https://fontforge.github.io/>

- Logos: <http://www4.flamingtext.com/>, <https://cooltext.com/>
- Música: <http://boscaoil.net/>
- Efectos de Sonido: <http://www.bfxr.net/>,  
<http://sfbgames.com/chiptone/>, [http://www.drpetter.se/project\\_sfxr.html](http://www.drpetter.se/project_sfxr.html)

b. Creados por terceros (algún compañero o amigo, o hay muchos que son gratuitos, pero es necesario cerciorarse de incluirlos en los créditos y de que la licencia sea válida para el uso):

- <http://kenney.nl/assets>
- <http://opengameart.org/>
- <https://www.freesound.org/>
- <https://www.gamedevmarket.net/>
- <https://mobilegamegraphics.com/>
- <http://indiegamemusic.com/>

✓ **Importante:** en caso de usar assets de terceros debe incluir en los créditos la siguiente información (por más que la licencia indique como opcional la mención):

- nombre del asset
- autor
- página web específica de donde lo obtuvo (no la general de la página que tenía todos los assets)

Ejemplos:

- UI Audio by Kenney (<https://kenney.nl/assets/ui-audio>)
- A blocky dungeon, creado por Buch, disponible en <https://opengameart.org/content/a-blocky-dungeon>

## 8. El juego debe contar con audio (al menos una música y tres sonidos)

## 9. Deben entregarse 3 archivos (se indicará cómo crearlos en clase)

- a. Archivo zip conteniendo proyecto (fuentes)
- b. Archivo zip conteniendo ejecutable (build)
- c. Video. El objetivo de este último entregable es que usted grabe un video del juego destacándolo lo mejor posible (por lo tanto, con la mejor calidad, mostrando sus mejores partes, sin elementos visuales de debug ni marcas del editor o programa que graba, etc.)

A estas pautas generales se pueden sumar pautas específicas para el tipo de juego que esté desarrollando, que se le irán comentando en clase y en las devoluciones que se le hagan.