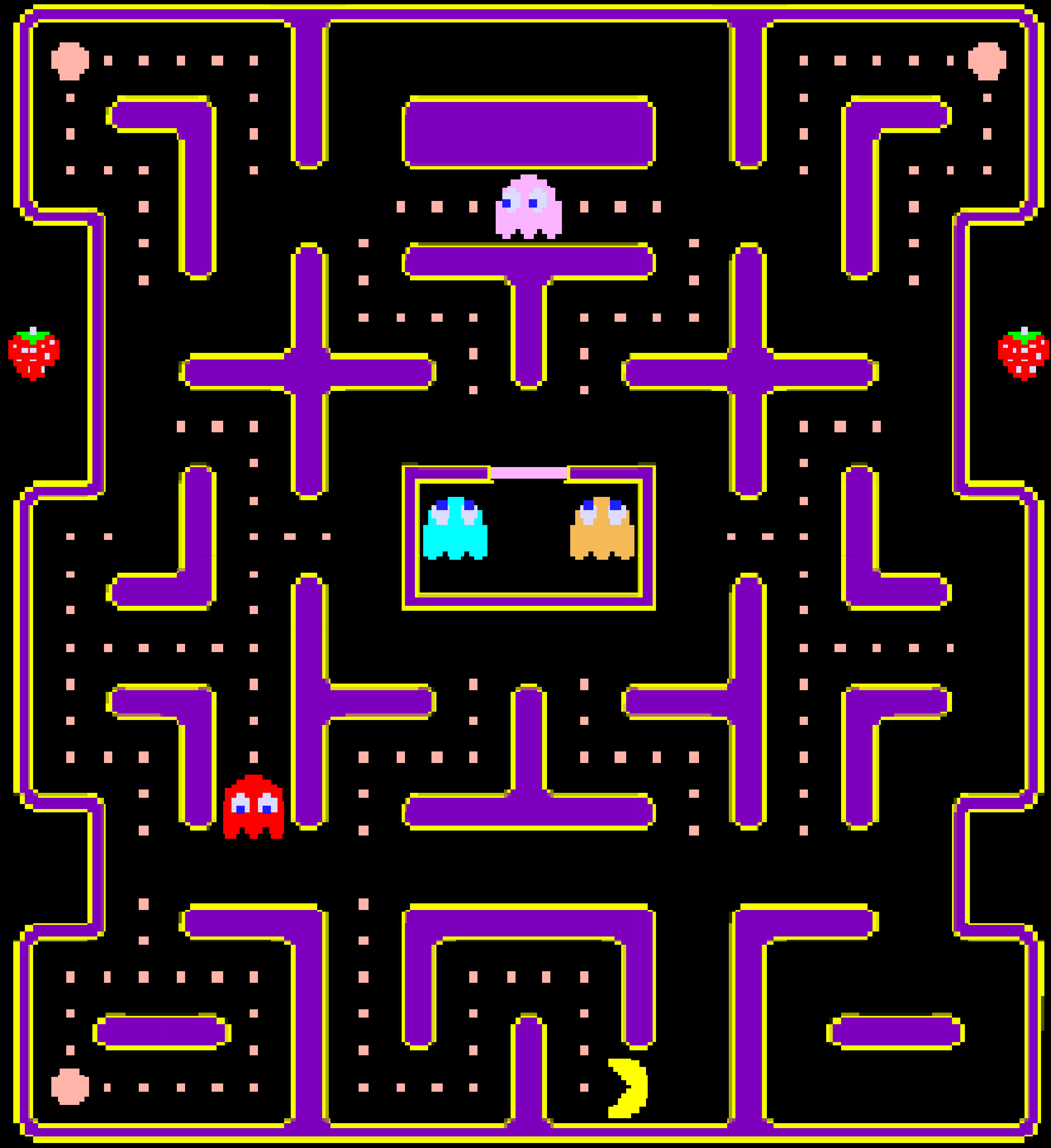
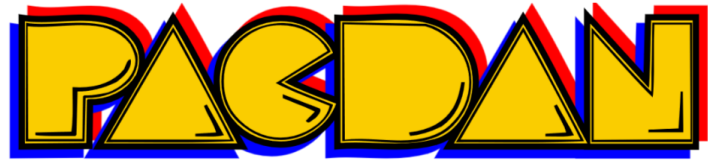


# RACON

BY: MIQUEL DURAN, ASHLEY CASTELLVI & GABRIEL SHARMAN

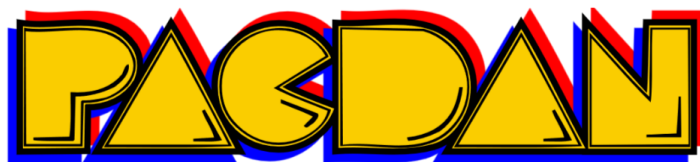






# ÍNDICE

<b>01</b>	<b>DESCRIPCIÓN/RESUMEN</b>
<b>02</b>	<b>REFERENCIAS/COMPETENCIAS</b>
<b>03</b>	<b>DAFO</b>
<b>04</b>	<b>GAMEPLAY</b>
<b>05</b>	<b>WORLDBUILDING</b>
<b>06</b>	<b>PERSONAJES E HISTORIA</b>
<b>07</b>	<b>LEVEL DESIGN</b>
<b>08</b>	<b>ASSETS Y ART STYLE</b>
<b>09</b>	<b>MÚSICA Y SONIDOS</b>
<b>10</b>	<b>TDD</b>
<b>11</b>	<b>PROPUESTA DE ESTUDIO</b>
<b>12</b>	<b>OTRAS IDEAS</b>
<b>13</b>	<b>DOCUMENTACIONES</b>



# **01 DESCRIPCIÓN/RESUMEN**

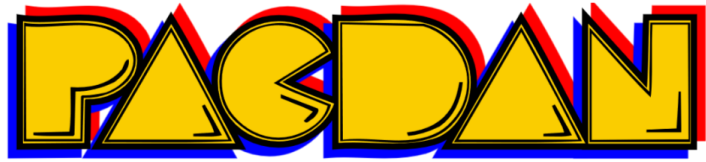
## **Pac-Dan: Un Clásico Reinventado**

**PAC-DAN es un juego de laberinto que moderniza la fórmula clásica del arcade. Los jugadores controlan a Pac-Dan mientras consume bolitas y power-ups, esquivando fantasmas en niveles cada vez más desafiantes. Esta edición conserva la esencia retro e incorpora nuevos modos de juego, niveles variados y pantallas llenas de acción.**

## **Tema y Género**

**Tema: Escape, Supervivencia**

**Género: Arcade, Puzzle de acción**



## **Público objetivo**

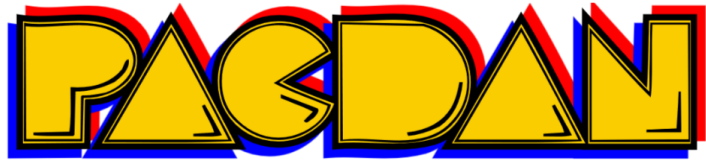
**Jugadores de entre 10 y 45 años, desde nostálgicos de los arcades clásicos hasta jugadores casuales y fanáticos de la acción rápida.**

## **Modos de juego**

- **Modo Práctica: Sin vidas ni fantasmas, apréndete los mapas y consigue puntuaciones elevadas.**
- **Modo Original: 3 vidas, 4 fantasmas, 4 power pellets, el modo de juego de toda la vida.**
- **Modo Difícil: 3 vidas, 4 fantasmas, 2 power pellets, para los jugadores mas experimentados.**
- **Modo Imposible: 3 vidas, 3 fantasmas, sin power pellets, el modo de juego para los que quieren un verdadero reto.**

## **Puntos clave (USP)**

- **Niveles con dificultad progresiva**



- **Música dinámica y envolvente**

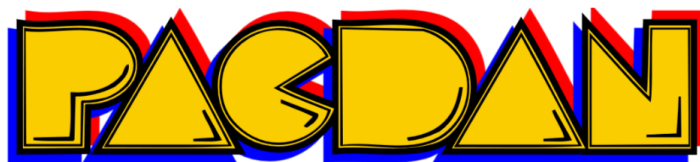
## **Pilares del juego**

**Gameplay:** Laberinto, evasión, recolección de objetos, uso de power-ups

**Dinámicas:** Huida, persecución, estrategia, exploración rápida

**Emociones:** Tensión, euforia, nostalgia

**Dirección artística:** Estética retro-futurista con neón, pixel art e influencias vaporwave



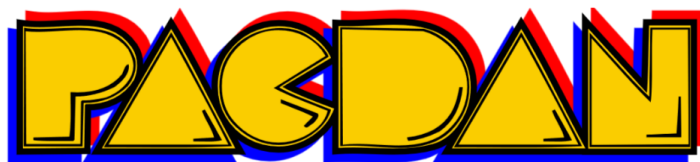
## **02 REFERENCIAS/COMPETENCIAS**

### **Pac-Man Championship Edition DX**

**Una versión moderna y acelerada del clásico Pac-Man. Introduce laberintos dinámicos, efectos visuales vibrantes y una jugabilidad basada en el tiempo. Mantiene la esencia original mientras añade velocidad, combos de fantasmas y power-ups estratégicos.**

### **Geometry Wars**

**Juego arcade de disparos con estética minimalista y efectos visuales intensos. El jugador controla una nave en un plano bidimensional, enfrentando oleadas de enemigos en partidas rápidas, caóticas y altamente adictivas.**



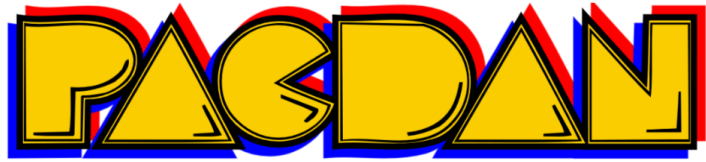
## **Enter the Dungeon**

**Roguelike de acción en el que los jugadores exploran un calabozo generado proceduralmente, combatiendo enemigos con un arsenal de armas excéntricas. Combina disparos, esquiva precisa y elementos de exploración con un tono humorístico y referencias a la cultura gamer.**

## **Super Mario Run**

**Plataforma de desplazamiento lateral optimizado para móviles. Mario corre automáticamente, mientras el jugador controla los saltos con un solo toque. Ofrece niveles clásicos reinterpretados para una experiencia accesible, rápida y rejugable.**





## **03 DAFO**

### **Debilidades**

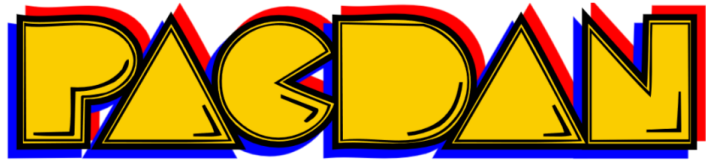
- **Limitada complejidad narrativa**
- **Juego simple que puede parecer repetitivo**

### **Amenazas**

- **Saturación del mercado de retro-remakes**
- **Desinterés de las nuevas generaciones**

### **Fortalezas**

- **IP reconocida mundialmente**
- **Jugabilidad inmediata y adictiva**



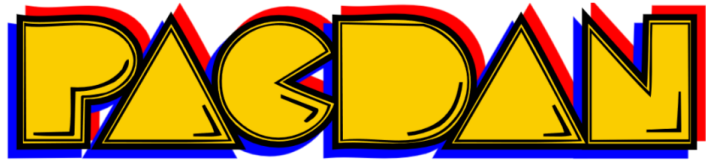
## **Oportunidades**

- **Expansión con integración del multijugador**
- **Contenido generado por usuarios**

# **04 GAMEPLAY**

## **Objetivos**

- **Comer todos los puntos del mapa**
- **Sobrevivir sin ser atrapado por los fantasmas**
- **Completar niveles y hacer nuevos récords**



## **Skills del usuario**

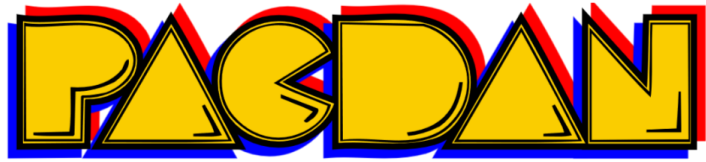
- **Reflejos**
- **Toma de decisiones rápida**
- **Memorización de rutas**

## **Mecánicas del juego**

- **Movimiento en 4 direcciones**
- **Power-up temporal para comerse a fantasmas**
- **Recolección de puntos**

## **Ítems y power-ups**

- **Superpíldora: permite atacar**



## **Recompensas y castigos**

- **Castigo: pérdida de vida al ser tocado por un fantasma**
- **Recompensa: más puntos al comer todos los fantasmas**

## **Economía del juego**

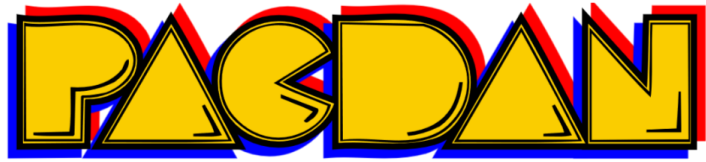
- **Puntos acumulables para conseguir nuevos récords**

## **Progresión y retos**

- **Niveles distintos con mayor dificultad**
- **Más fantasmas y menos vidas**

## **Gameplay loop**

- 1. Entra al nivel**
- 2. Recoge puntos**
- 3. Evade fantasmas**



**4. Usa power-ups estratégicamente**

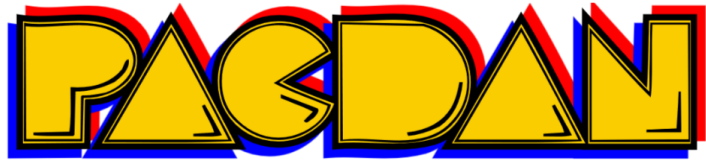
**5. Completa el nivel**

## **Enemigos y obstáculos**

- **Blinky (veloz)**
- **Pinky (predice movimiento)**
- **Inky (errático)**
- **Clyde (impredecible)**

## **Estado de derrota y vidas**

- **Muerte al contacto: 1 golpe = pérdida de vida**
- **Número limitado de vidas**



## **Rejugabilidad**

- **Variedad de niveles y modos**
- **Rutas alternativas**
- **Ranking online**

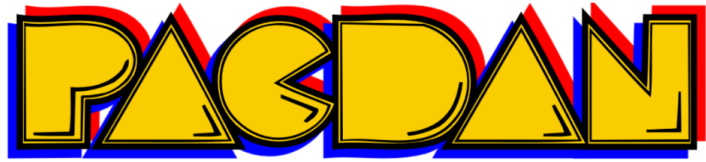
## **Guardado del juego**

- **Auto-save al final de cada nivel**

# **05 WORLDBUILDING**

## **Mapas**

- **Miki Maze: Un laberinto moderno con diseño fluido, ideal para niveles intermedios. Añade caminos estrechos y zonas de riesgo controlado.**



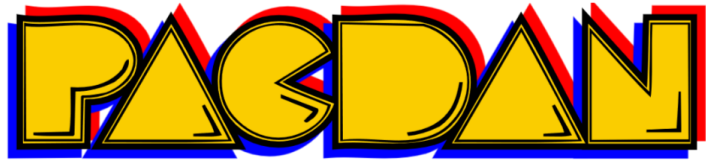
- **Original: El clásico laberinto de Pac-Man, equilibrado en dificultad, perfecto para el modo original y jugadores nostálgicos.**
- **Ms. Pacman 1: Inspirado en el primer nivel de Ms. Pac-Man, con rutas más curvas y una distribución más abierta.**
- **Ms. Pacman 2: Mapa con mayor complejidad estructural y múltiples rutas, ideal para jugadores avanzados que buscan variedad y desafío.**

## **06 PERSONAJES E HISTORIA**

### **Pac-Man**

#### **Body:**

- **Cuerpo esférico, amarillo brillante**



**Mind:**

- **Determinado**
- **Instinto de supervivencia**

**Objetivos:**

- **Obtener la mayor puntuación posible**

**Antagonistas:**

- **Fantasmas**

**Historia**

**Tema:**

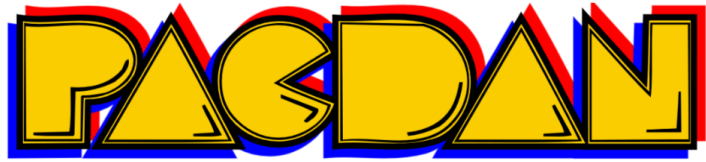
- **Lucha por la libertad digital**

**Género:**

- **Ciencia ficción ligera**

**Tono:**





- **Misterioso pero alegre**

### **Setting:**

- **Mundo virtual que se fragmenta**

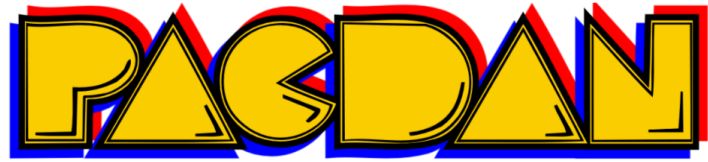
## **07 LEVEL DESIGN**

### **Metas**

- **Distintas dificultades según las habilidades del jugador**

### **Mapas**

- **Iniciales: Simples**
- **Avanzados: Con rutas múltiples y más complejos**



# **08 ASSETS Y ART STYLE**

## **Estilo artístico**

### **Art Bible**

- **Neón, pixel 2D**

### **Moodboard**

- **Inspiraciones: Tron, Vaporwave, Synthwave**

### **Paleta**

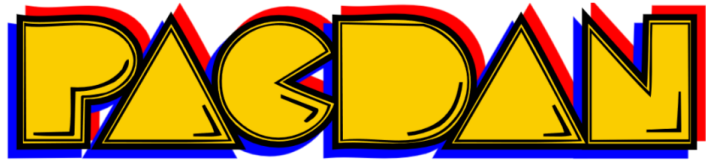
- **Colores brillantes sobre fondo oscuro**

### **Texturas**

- **Digitalizadas, con efectos glitch**

### **UI/UX**

- **Interfaces retro-modernas**



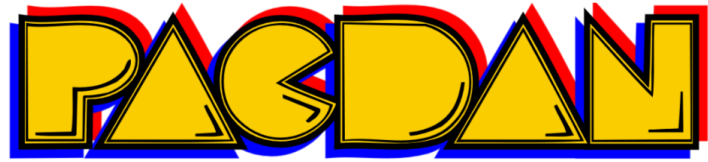
## **HUD**

- **Elementos visibles: puntuación, vidas**

# **09 MÚSICA Y SONIDOS**

## **BSO**

- **Música electrónica retro-futurista**
- **Temas dinámicos que cambian según la dificultad**



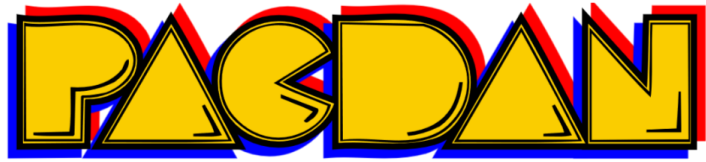
# **10 TECHNICAL DESIGN DOCUMENT**

## **Fase 1: Preproducción**

- **Definición del concepto y diseño general**
- **Creación de documentación (game design document)**
- **Bocetos y moodboard de arte y sonido**

## **Fase 2: Prototipo inicial**

- **Desarrollo básico de gameplay: movimiento, enemigos, power-ups**
- **Implementación de uno o dos mapas básicos**
- **Testeo interno y ajustes iniciales**

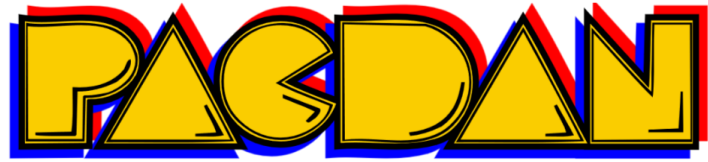


## **Fase 3: Producción**

- **Desarrollo completo de niveles y modos de juego**
- **Integración de arte final y animaciones**
- **Programación de UI/HUD y mecánicas avanzadas (power-ups, IA fantasmas)**
- **Implementación de música dinámica y efectos de sonido**

## **Fase 4: Pulido y testeo**

- **Optimización para PCs de bajo nivel**
- **Corrección de bugs y balance de dificultad**
- **Pruebas de jugabilidad y experiencia usuario**

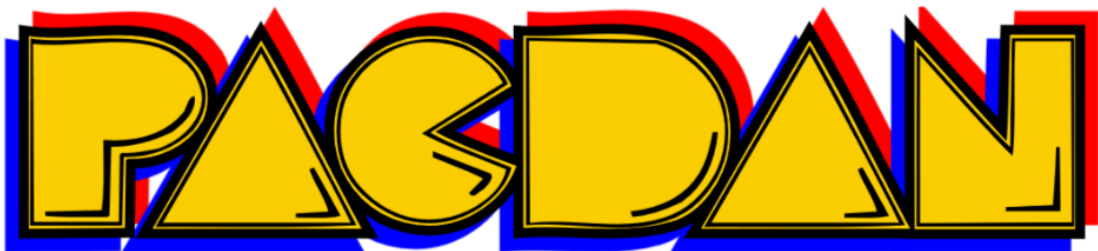


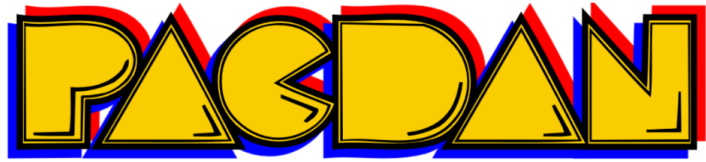
## **Fase 5: Lanzamiento y soporte post-lanzamiento**

- **Preparación y publicación en plataformas seleccionadas**
- **Actualizaciones, parches y posible integración de contenido adicional (modos multijugador, mapas)**

# **11 PROPUESTA DE ESTUDIO**

## **Logo**





## **Marketing y Funding**

- **Kickstarter / Steam Early Access**
- **Redes sociales y TikTok con challenges retro**

## **Demographics**

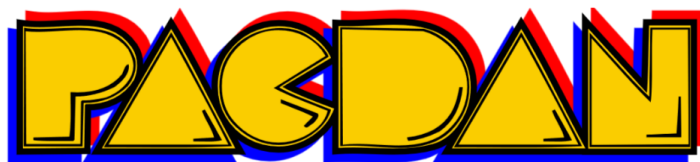
- **Nostálgicos (25-40 años)**
- **Casual gamers (10-30 años)**

## **Plataformas**

- **PC**

## **Monetización**

- **Compra única**



## **Localización**

- **Idiomas principales: Inglés, Español**

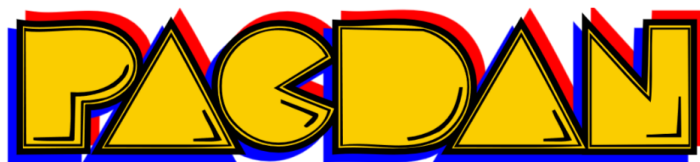
# **12 OTRAS IDEAS**

## **Modo editor de niveles**

**El juego incluirá un editor de niveles intuitivo que permitirá a los jugadores crear, personalizar y compartir sus propios laberintos.**

- **Herramientas sencillas para diseñar caminos, colocar fantasmas y objetos.**
- **Opciones para ajustar la dificultad y seleccionar música ambiental.**
- **Integración con la plataforma online para subir y descargar mapas creados por la comunidad.**





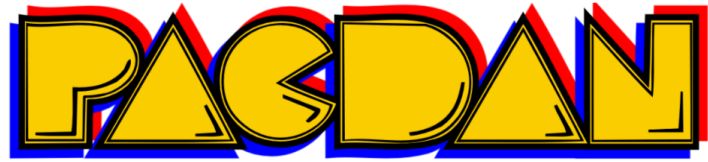
- **Posibilidad de votar y destacar los mejores niveles, fomentando la creatividad y la rejugabilidad.**

## **Eventos semanales**

**Para mantener la comunidad activa y motivada, se organizarán eventos semanales con diferentes desafíos y recompensas.**

- **Modos temporales con reglas especiales (por ejemplo, velocidad aumentada, fantasmas adicionales).**
- **Competencias de puntuación con tablas de líderes semanales.**
- **Recompensas exclusivas para participantes y ganadores, como skins o power-ups especiales.**
- **Eventos temáticos relacionados con fechas especiales o temporadas.**

## **13 DOCUMENTACIONES**



# Documentació: Digitalització

## RA3 – UI:

Utilitzant els tilesets de Unity, hem importat els sprites originals de Pac-Man per fer la primera paleta de colors, i hem creat una UI basant-nos en l'original, però modificada per fer-la més senzilla al nostre gust.

Aquí hem recreat el mapa original del Pac-Man (amb una lleugera modificació)



1. Comptador de puntuació.
2. Comptador de vides restants
3. Laberint "OG"
4. Boto HOME per retornar al menú principal

# PACDAN

Hem aprofitat la forma del mapa per crear els menús de selecció de nivell, així fa una experiència més seguida, ja que tot és visualment similar i saps on mirar tota l'estona, a més de fer que el jugador no es perdi.

També hem fet altres paletes de color seguint la spritesheet original, així es poden distingir altres segments del menú, i com el joc principal segueix el color de l'original, se sap que les coses interactuables són d'aquell color.

El Menú Principal:



1. Títol del Joc
2. Panell de Puntuacions globals
3. Espai de registrar el nom de jugador
4. Panell d'informació (referència a l'original)
5. Botons de Selecció de Mode
6. Botó del Menú de configuració
7. Boto per tancar el joc.

# PACDAN

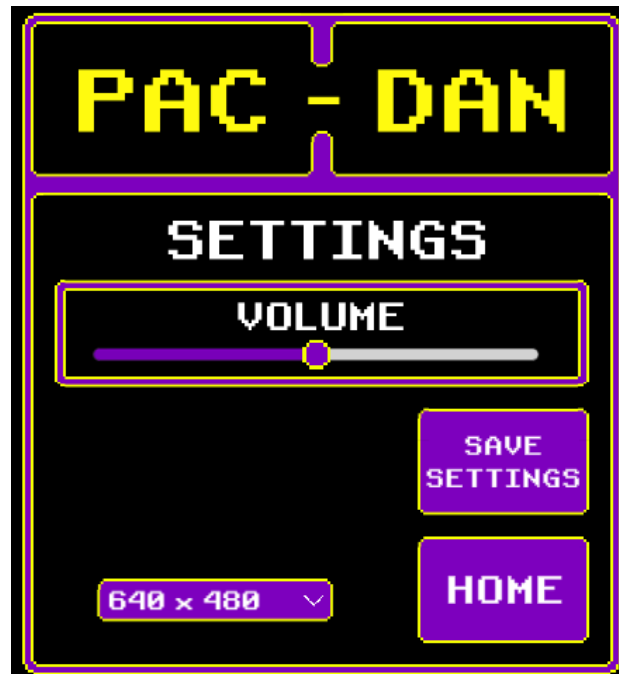
Els menús de selecció de mapa de cada mode segueixen la mateixa forma per mantenir la coherència entre menús, però cadascun té colors diferents per així diferenciar-los més entre ells i que se sàpiga fàcilment quin és quin.



Inclouen els mateixos quatre mapes cadascun, a més d'un botó per tornar endarrere.

# PACDAN

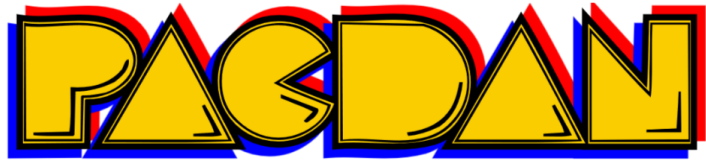
El Menú d'Ajustaments segueix la mateixa forma, però s'ha modificat per dins perquè la barra de volum hi càpiga, junt amb el seleccionador de resolució. També té un botó de HOME retocat.



Pantalla de Títol:



1. Títol de Joc (estilitzat)
2. Crèdits
3. Advertència de salut i seguretat (referència a les consoles de Nintendo)
4. Detalls per emplenar (Origen: Màquines Àrcade originals)

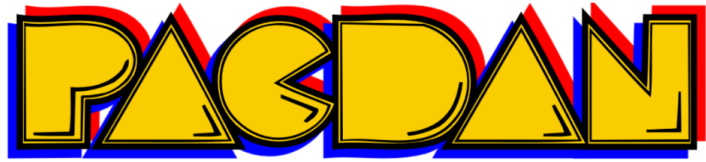


5. Boto per iniciar
6. Companyia totalment real

La pantalla de títol es completament original. Utilitzant la paleta original hem fet la pantalla, el separador del títol, i el boto de començar, que esta posat de forma que sigui ben visible, a mes, al estar amb la paleta original, se sap que es important.

A l'esquerra hem posat els noms de cadascun en una paleta que ens agradés.

I a la dreta hem posat l'avertencia que sempre surt a les consoles de Nintendo, mes que res com un petit gag per omplir l'espai que quedava. (hi ha una sorpresa si es clica l'enllaç al fons).



## RA4 i RA5 - Escena i Càmera:

El Pac-Man, en ser un joc 2D amb una paleta limitada, només necessita Sprites ben posats. Tenim quatre nivells diferents, cadascun amb quatre variacions, depenent de quin mode s'hagi escollit.

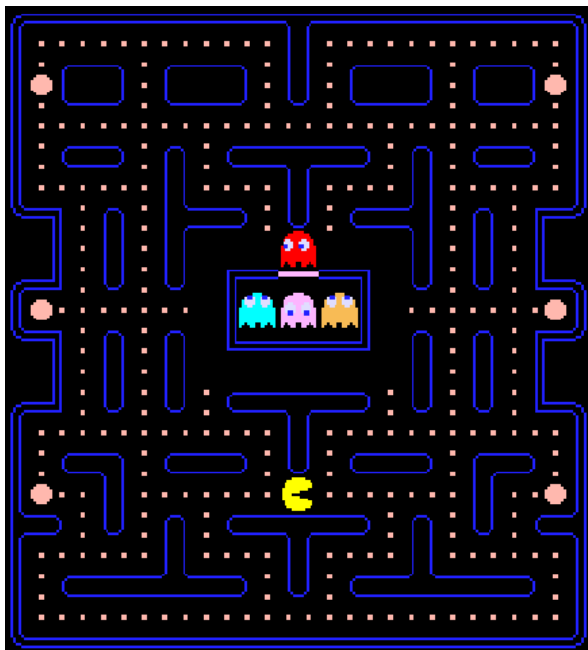
Cadascuna inclou diferents capes, les quals són els "Pellets", les parets que fan el laberint, i una invisible que són els nodes, posats a cada intersecció o gir perquè els fantasmes puguin saber on anar.

Totes inclouen en Pac-Man (sempre al mateix lloc d'inici) i els "Pellets", però depenent de quin mode sigui, pot incloure menys fantasmes, o menys "Power Pellets"

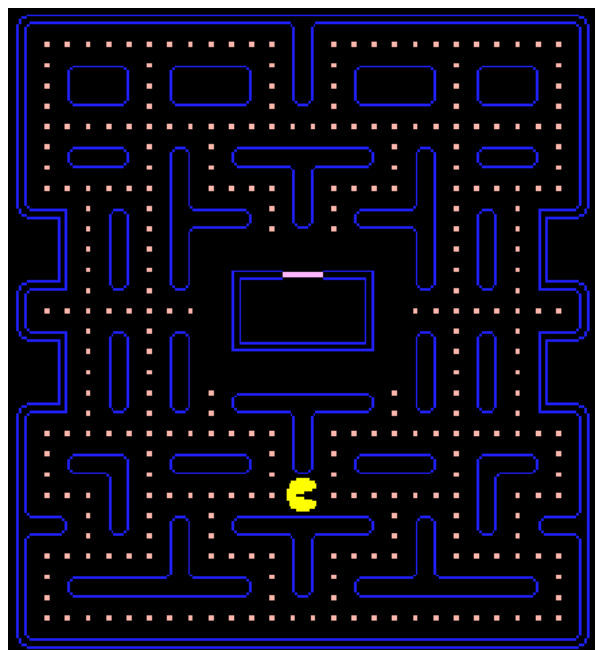
La càmera es manté estàtica constantment, ja que l'únic moviment és d'en Pac-man i els fantasmes al laberint.

Mapa "OG" i les seves Variacions:

Mode: Original

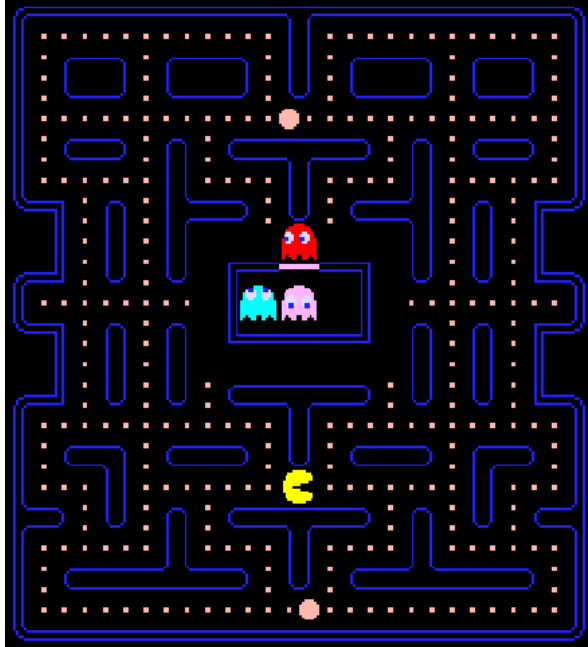


Mode: Practice

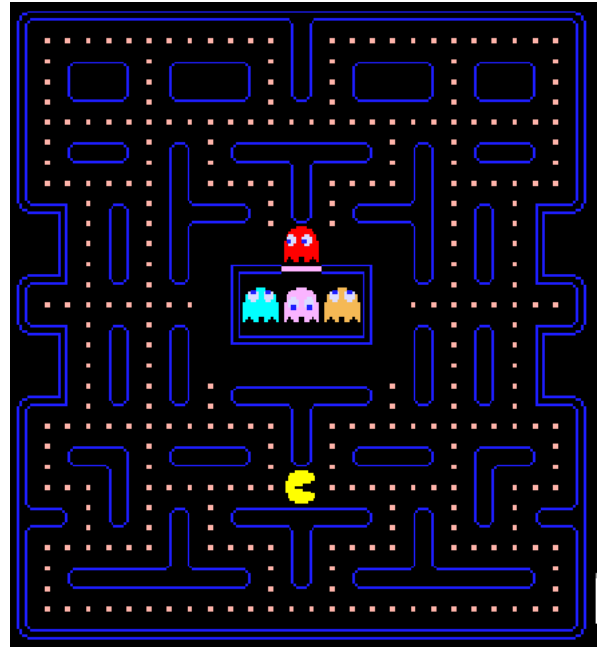


# PACDAN

Mode: Tough (Difícil)

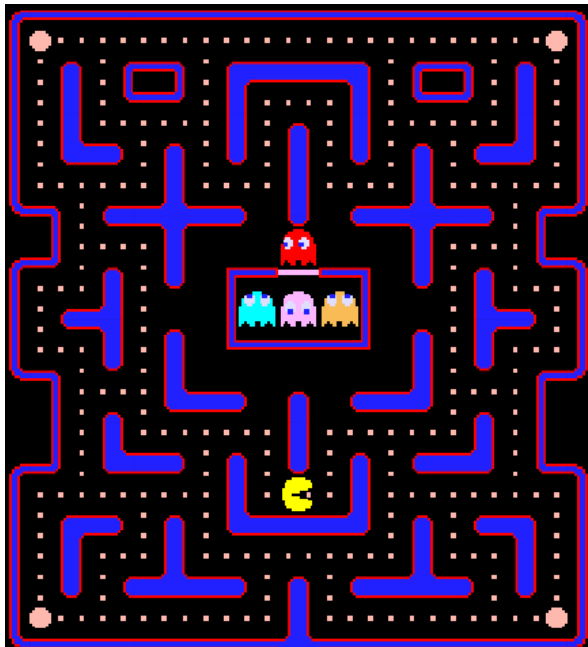


Mode: Impossible

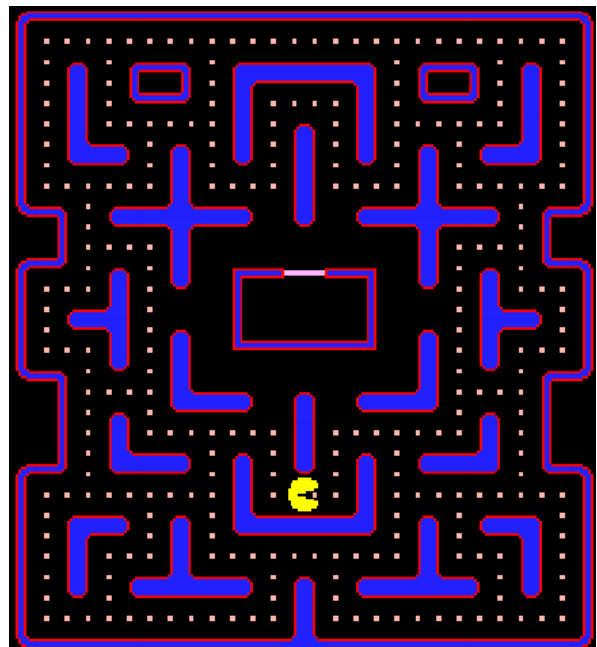


Mapa “Ms. Pac-Man” 1 i les seves variacions:

Mode: Original



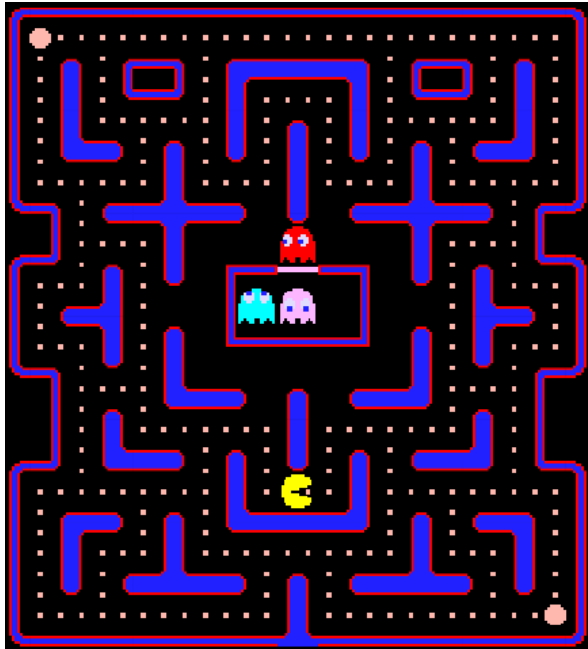
Mode: Practice



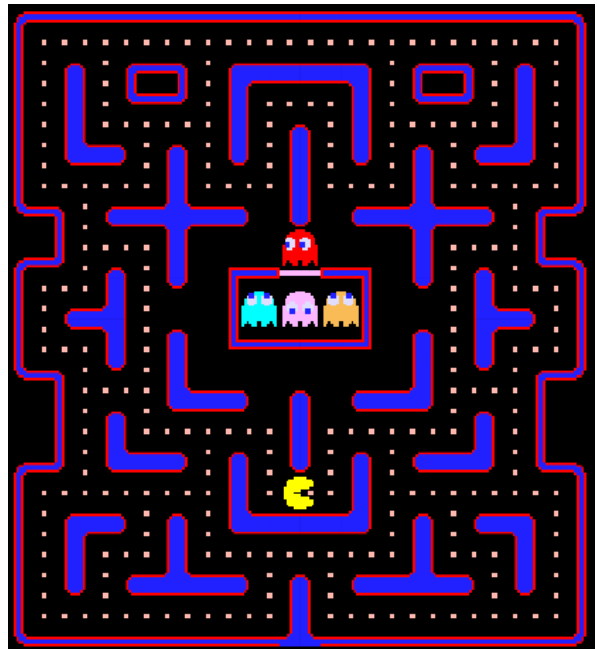


# PACMAN

Mode: Tough (Difícil)

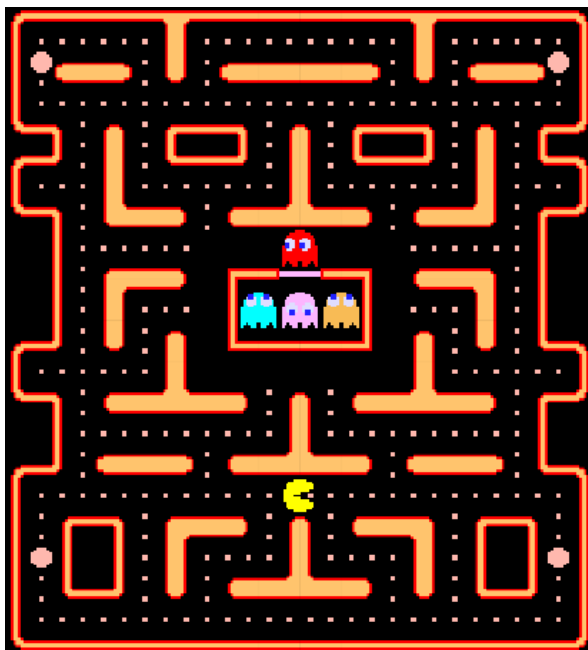


Mode: Impossible

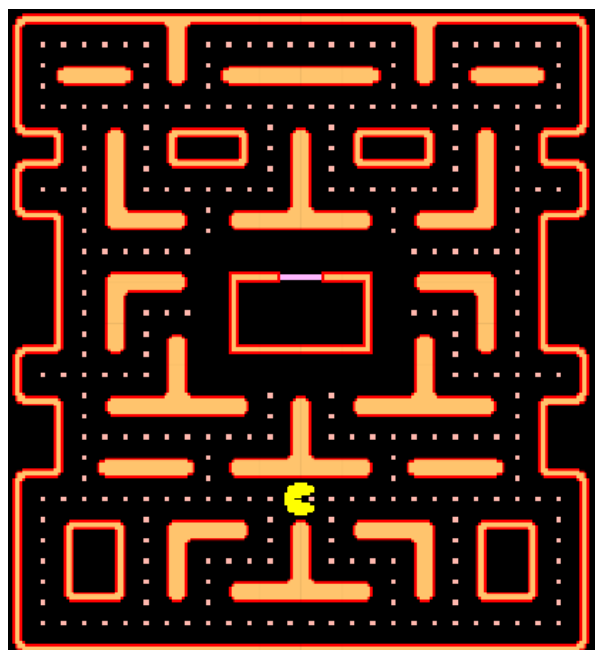


Mapa “Ms. Pac-Man 2” i les seves variacions:

Mode: Original



Mode: Practice

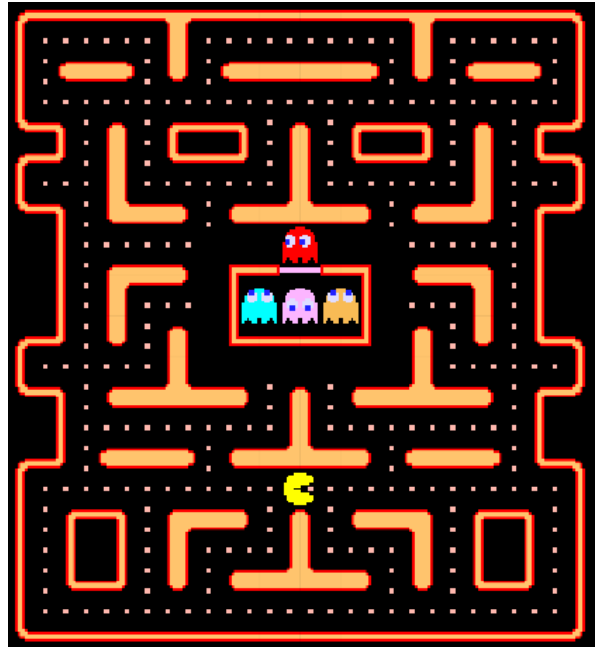


# PACDAN

Mode: Tough (Difícil)

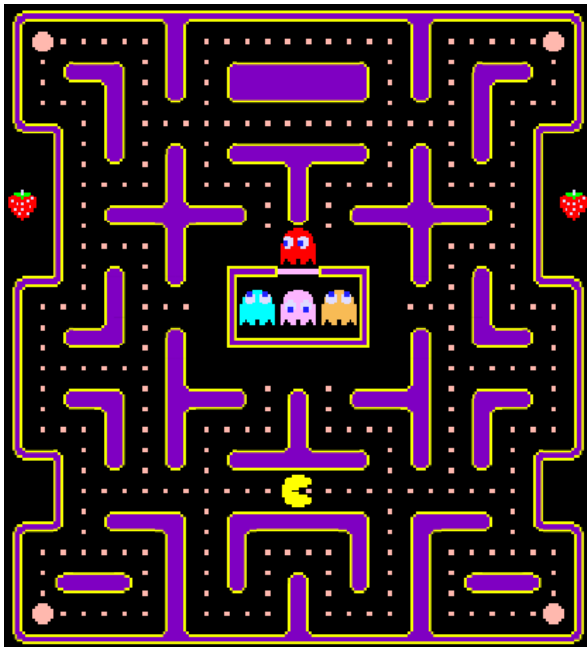


Mode: Impossible

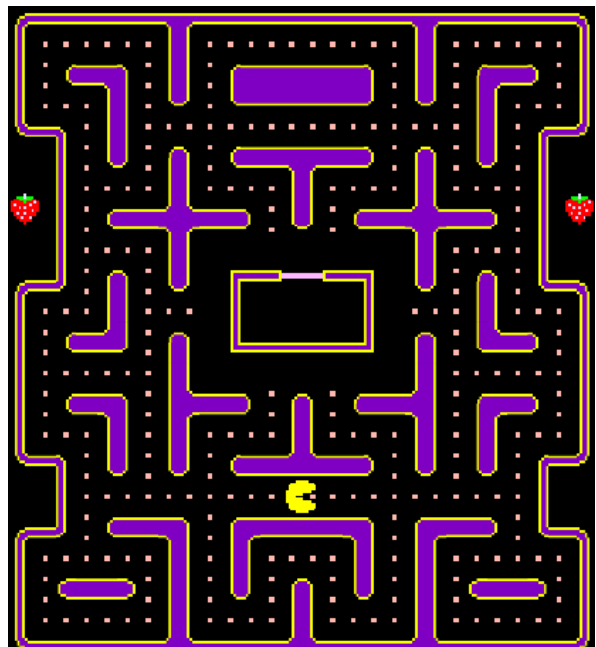


Mapa “MIKI’s Maze” i les seves variacions:

Mode: Original

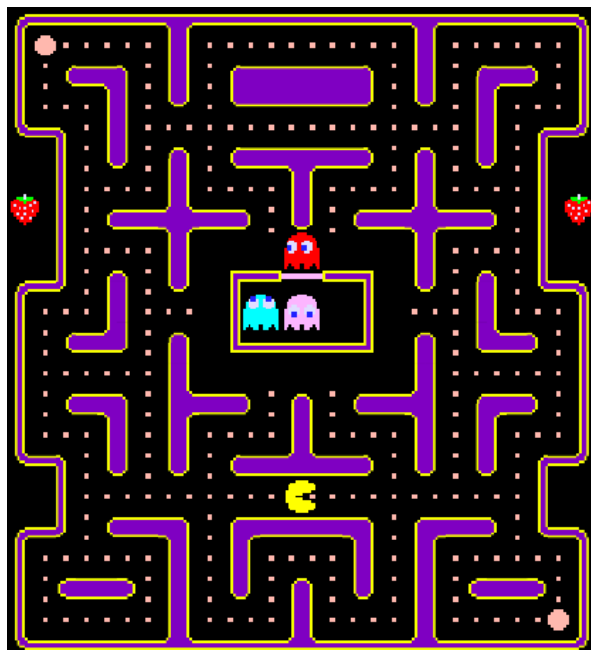


Mode: Practice

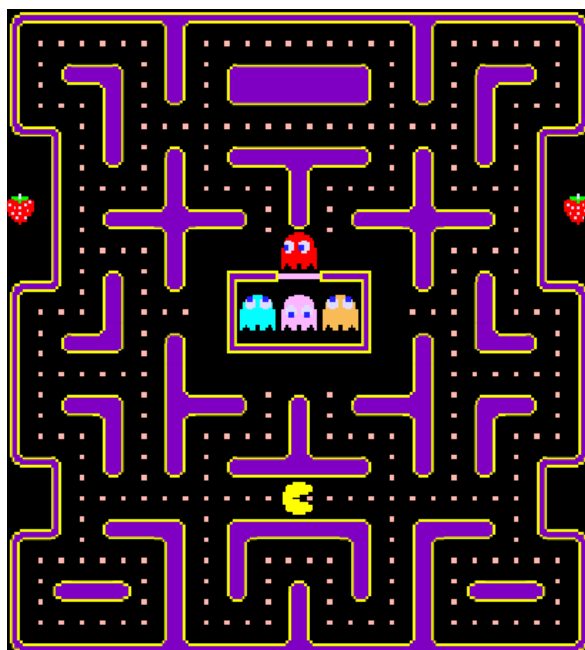


# PACDAN

Mode: Tough (Difícil)



Mode: Impossible



## RA6 - Sonido:

Cada Escena té la seva pròpia música, extreta directament dels arxius dels Jocs “Namco Museum DS”, “PacPix”, “Pac’n’Roll” i “Dragon Ball Z SuperSonic Warriors 2”.

De Namco Museum hem agafat la cançó d'introducció de Pac-Man, i l'hem posat al Títol.

De PacPix i Pac’n’Roll hem agafat les cançons del menú principal, Original, Practice i Tough. A més de les cançons del joc Practice, Tough i Impossible.

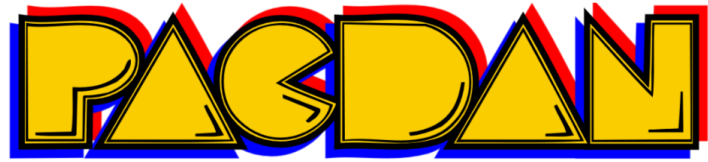
El mode original utilitza el so d'alarma del joc original, per mantenir el realisme.

De “Dragon Ball Z SuperSonic Warriors 2” hem agafat les cançons del Menú d'Ajustaments, i del Menú Impossible.

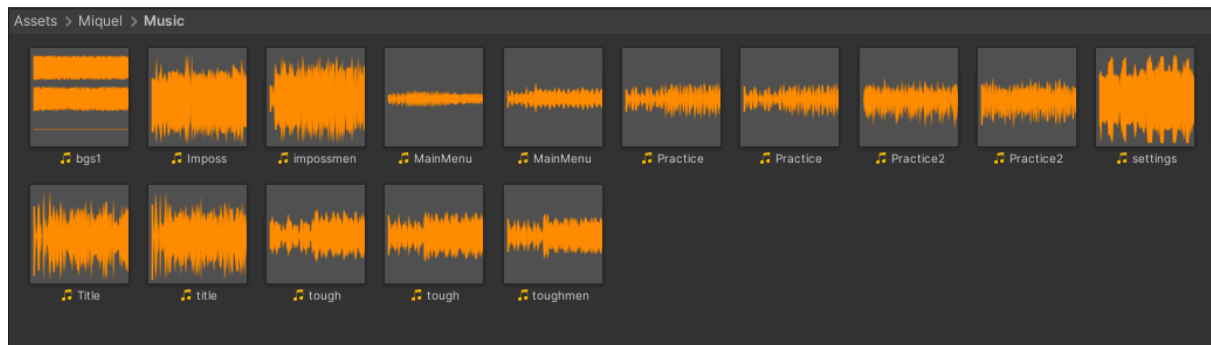
La música és lleugerament dinàmica, ja que teníem accés a modificar les cançons, al Menú Principal, Practice, i Tough els hi hem tret la melodia. En accedir al joc Practice o Tough, les seves cançons canvien lleugerament per incloure melodia, creant una progressió natural.

La música del Menú Principal afegeix melodia al menú Original.

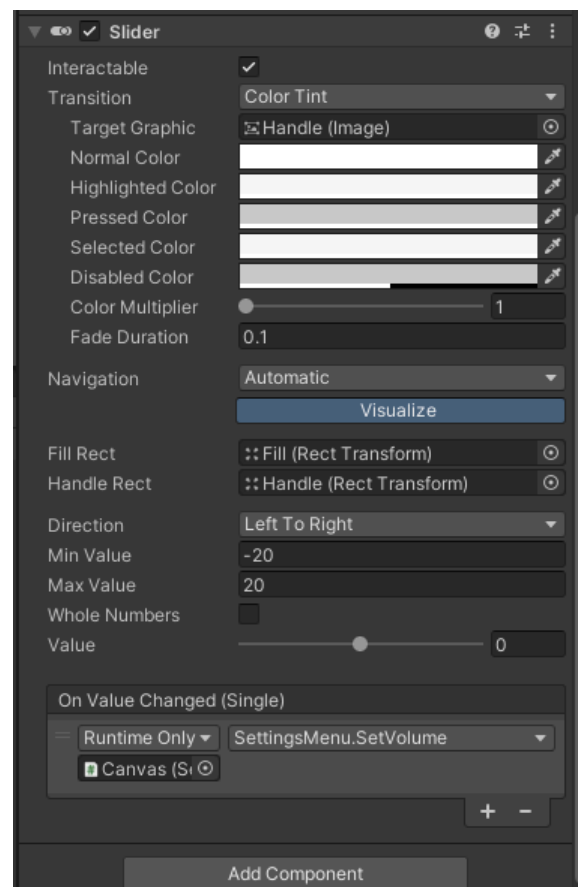
Al menú d'Ajustaments hi ha un lliscador per apujar o abaixar el volum de la música.

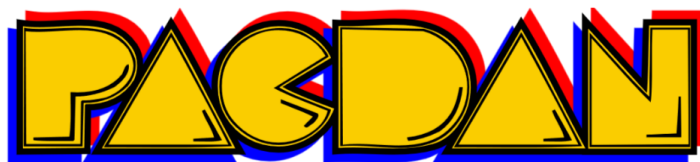


La Carpeta on es guarda la musica:



El lliscador de Volum i les seves propietats:





## Documentació: Videojocs

### RA7 – Animaciones:

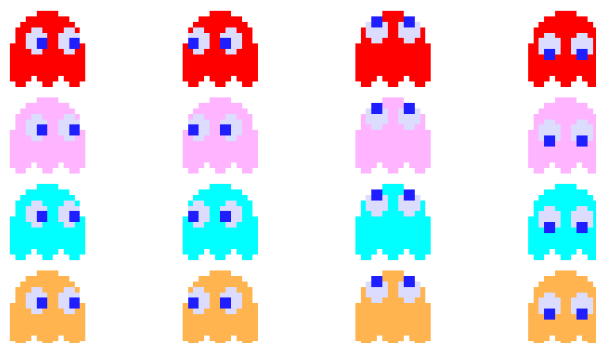
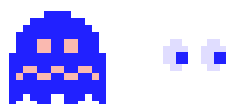
Al Pac-Man Original hi havia poques animacions, al nostre joc hem incorporat les poques que hi havia.

Inclou les animacions de:

Pacman: Té l'animació de menjar i la de morir.



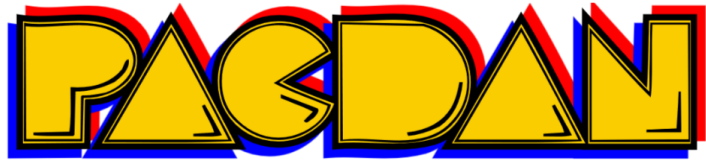
Fantasmes: Animació de moure's, canviar a ser comestibles, ser menjats, i miren a la direcció on es mouran.



### RA8 – Mecanicas:

El personatge PacMan té dues Mecàniques: Moure's pel Laberint, i Menjar-se els fantasmes.

Els Fantasmes en tenen 3, Moure's de forma aleatòria, anar direcció del jugador, i quedar-se descansant a casa quan són menjats.



## RA9 – Patrones de Diseño y tecnicas algoritmicas - Workflow:

A cada escena hi ha un GameManager que controla la partida.

Inclou informació de control necessària, com ara les Power Pellets que hi ha a l'escenari i els fantasmes que hi ha.

Controla com funciona el joc, on és la casa, quan començar, quan reiniciar-se, i més.

Exsert del codi que executa el reinici en perdre del script GameManager:

```
private void NewGame()
{
    gamesPlayed++;
    SetScore(0);
    SetLives(3);
    NewRound();
}

private void NewRound()
{
    gameOverText.enabled = false;

    foreach (Transform pellet in pellets)
    {
        pellet.gameObject.SetActive(true);
    }

    ResetState();
}

private void ResetState()
{
    foreach (Ghost ghost in ghosts)
    {
        ghost.ResetState();
    }

    pacman.ResetState();
}

private void GameOver()
{
    ScoreSaver saver = new ScoreSaver();
    saver.SaveScore(playerName, score);

    gameOverText.enabled = true;

    foreach (Ghost ghost in ghosts)
    {
        ghost.gameObject.SetActive(false);
    }

    pacman.gameObject.SetActive(false);
}
```

# PACDAN

Hem afegit 4 modalitats de joc diferents, els quals tenen petites diferències que canvien el joc completament.



Original: Experiència PacMan Original, res canviat.

Practice: No hi ha fantasmes, ni Power Pellets. Per practicar els mapes individualment.

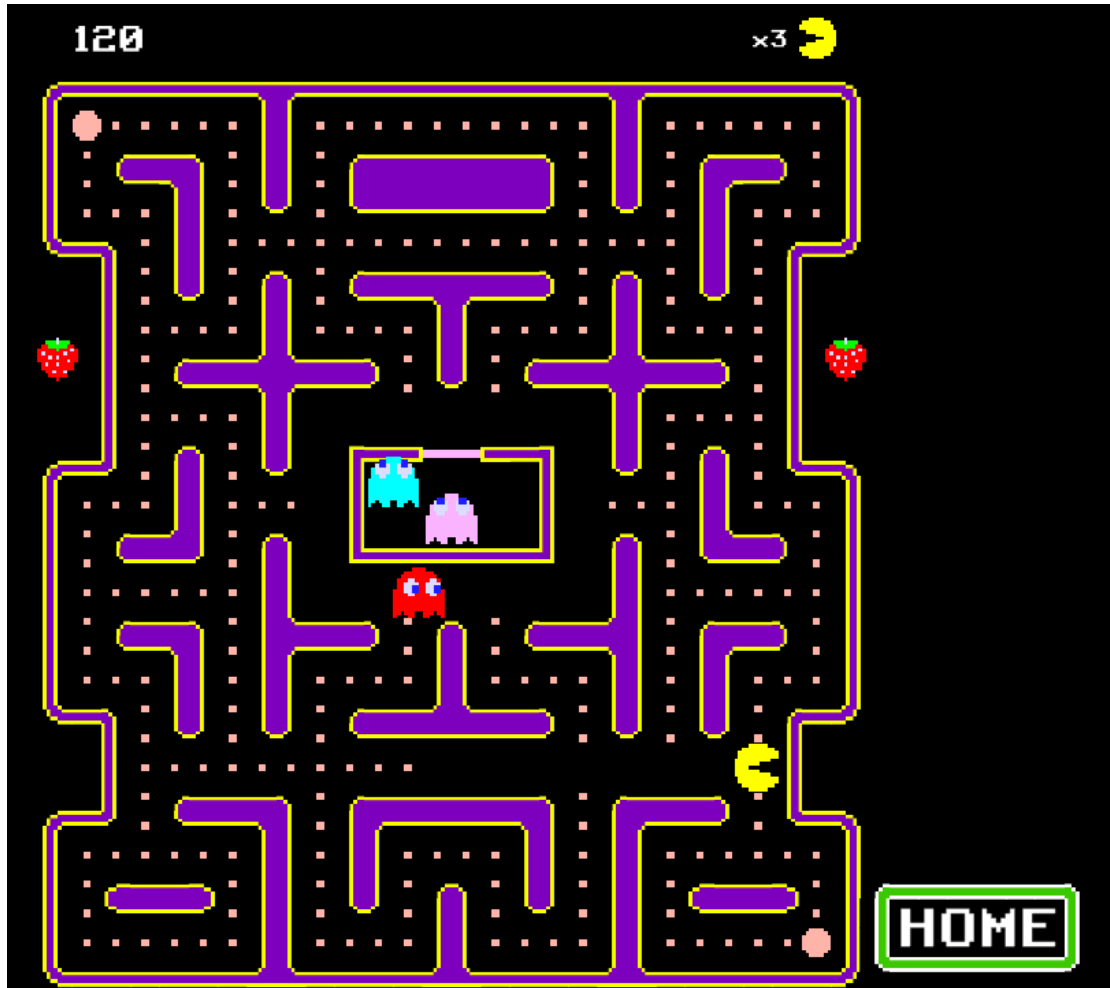
Tough: Tres fantasmes, Dues Power Pellets, una mica més complicat que l'original

Impossible: 4 Fantasmes i Zero Power Pellets. Mode molt difícil, per una experiència diferent.

Aquestes Modalitats són creades a força de modificar els paràmetres del GameManager a cada escena, especificant que hi ha i eliminant el que és necessari perquè funcioni.

# PACDAN

No hi ha cap pantalla de pausa afegida, ja que com a l'original, perdria tota la dinàmica i el risc que inclou el joc. Si es pogués aturar a qualsevol moment el joc no tindria gràcia. Per compensar, hem permès que es pugui sortir en qualsevol moment.

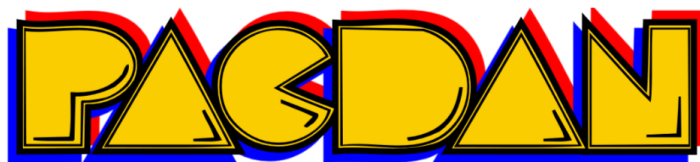




# PACDAN

Hem assegurat que hi hagi un menú per tot el necessari, Ajustaments, Títol, Selecció de Modalitat i Selecció de Mapa.





# Documentació: Bases de Dades i Llenguatge de marques

## MongoDB:

Primer vam crear un compte a MongoDB Atlas per a poder crear la nostra propia Base de dades i poder-nos connectar.

### Connect to TragabolasPacdan



You need to secure your MongoDB Atlas cluster before you can use it. Set which users and IP addresses can access your cluster now. [Read more](#)

#### 1. Add a connection IP address

✓ Your current IP address (85.192.70.145) has been added to enable local connectivity. Only an IP address you add to your Access List will be able to connect to your project's clusters. Add more later in [Network Access](#).

#### 2. Create a database user

This first user will have [atlasAdmin](#) permissions for this project.

We autogenerated a username and password. You can use this or create your own.

ⓘ You'll need your database user's credentials in the next step. Copy the database user password.

Vam configurar-la donant accés a tots els ordinadors i afegint tot el necessari per al seu bon funcionament.

Atlas Ashley A24... Access Manager Billing All Clusters Get Help Ashley A24

Project 0 Data Services Charts

Clusters SERVICES Atlas Search Stream Processing Triggers Migration Data Federation SECURITY Quickstart Backup Database Access Network Access

ASHLEY A24'S ORG - 2025-05-09 > PROJECT 0

## Network Access

IP Access List Peering Private Endpoint

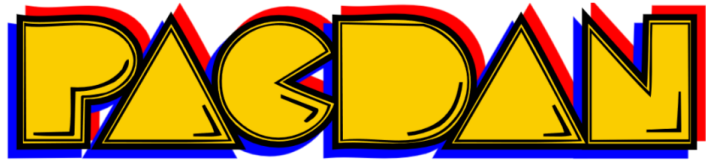
+ADD IP ADDRESS

ⓘ You will only be able to connect to your cluster from the following list of IP Addresses:

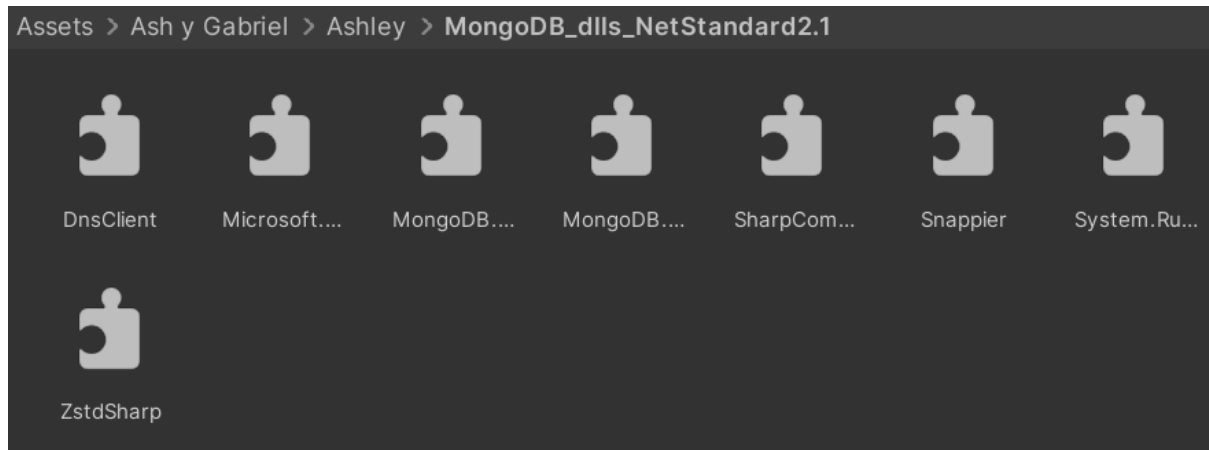
IP Address	Comment	Status	Actions
0.0.0.0/0 (includes your current IP address)		Active	EDIT DELETE

System Status: All Good

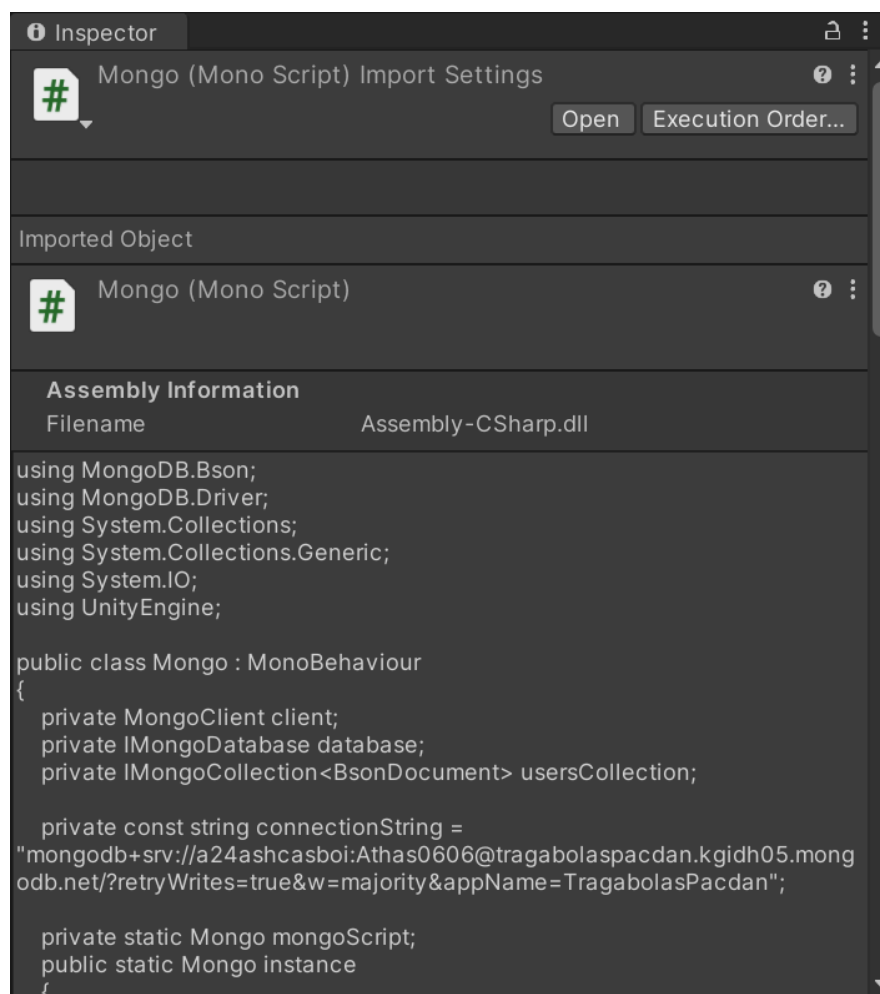
©2025 MongoDB, Inc. Status Terms Privacy Atlas Blog Contact Sales

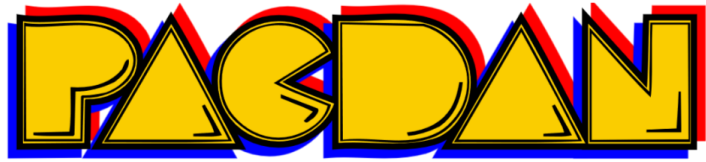


Després vam anar a Unity i vam importar tots els assets necessaris per al bon funcionament de MongoDB al nostre projecte.



Vam crear el script necessari per a poder connectar la nostra base de dades amb el Unity.

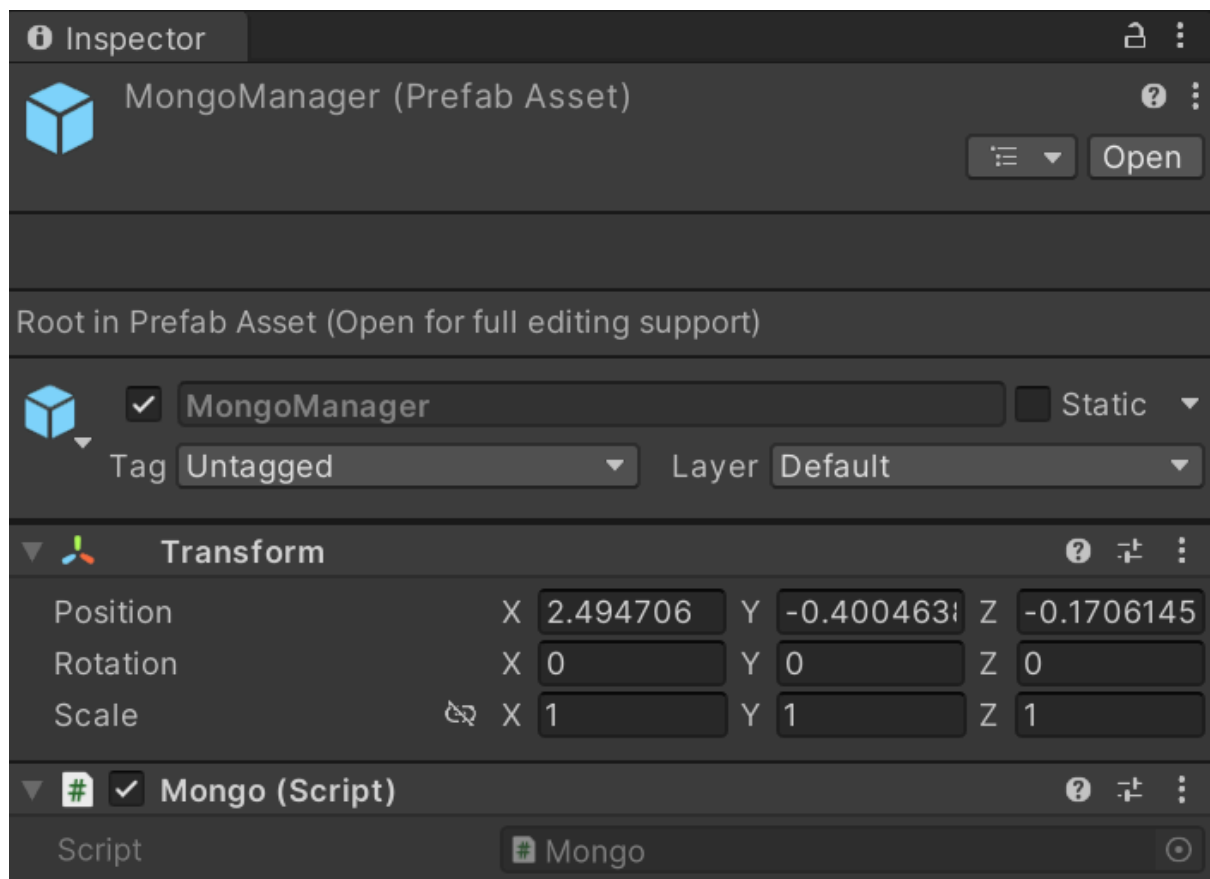


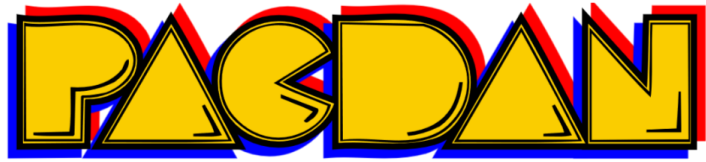


I vam posar el codi necessari al Game Manager perquè s'integrés la base de dades al joc i poder agafar totes les dades que necessitàvem.

```
public BsonDocument CreateUserTelemetryBSON()
{
    BsonDocument bsonDoc = new BsonDocument {
        { "Puntos obtenidos", score },
        { "Fantasmas comidos", ghostsEaten },
        { "Tiempo de juego", gameSessionTime },
        { "Niveles completados", levelsCompleted },
        { "Partidas jugadas", gamesPlayed }
    };
    return bsonDoc;
}
```

Per últim, necessitàvem poder executar-lo i llavors vam crear un prefab buit amb el script del mongo i una part de codi que feia que s'activés al pulsar la lletra "K".

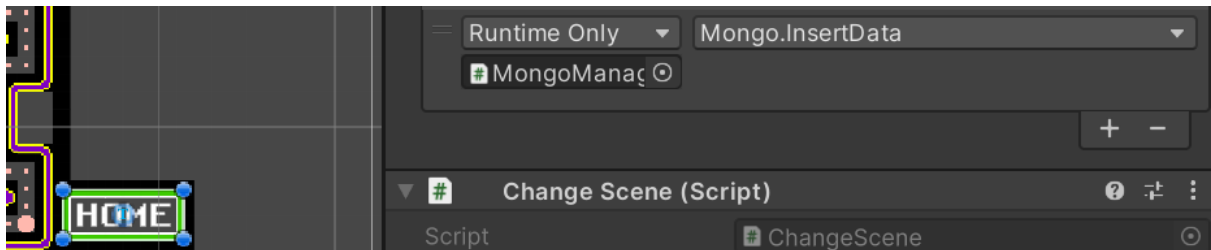


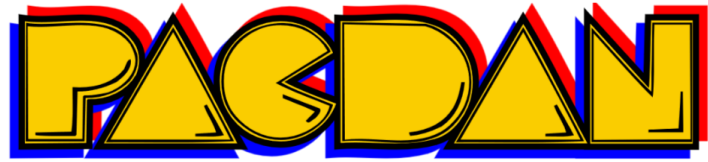


Part del script el qual activa el procés en prémer la lletra “K”:

```
void Update()
{
    if (Input.GetKeyDown(KeyCode.K))
    {
        InsertData();
    }
}
```

Més tard vam implementar que en finalitzar una partida prenent el botó “HOME” (la qual és l'única forma de finalitzar aquesta) s'activés automàticament la funció InsertData().





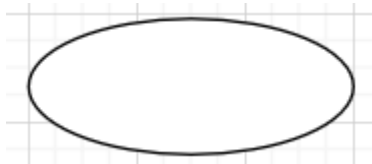
# Documentació: Entorns de desenvolupament

## RA6 – Diagrames de flux i d'estat:

Hem utilitzat [Draw.io](https://draw.io) per fer el diagrama de Flux d'aquest projecte. Un diagrama de flux és una representació gràfica d'un procés. Utilitza formes com fletxes, rectangles i rombes per mostrar l'ordre de les accions, decisions i resultats. Cada figura representa un pas concret i les fletxes indiquen el flux o la direcció del procés. S'utilitza per entendre, analitzar o millorar procediments. Per a fer el Diagrama de flux hem utilitzat 4 tipus de blocs per a aclarir que feia cada situació. Els blocs utilitzats han sigut els següents:

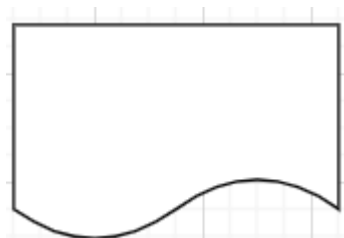
### El·lipse:

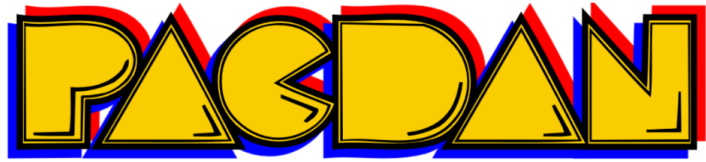
Bloc utilitzat per marcar l'inici i el final del programa.



### Document:

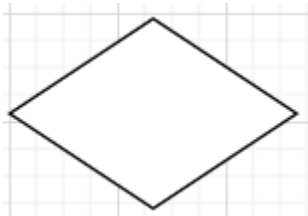
Utilitzat per a senyalitzar que s'obre un document, nosaltres l'hem posat quan l'acció anterior involucrava obrir un altre nivell o mapa.





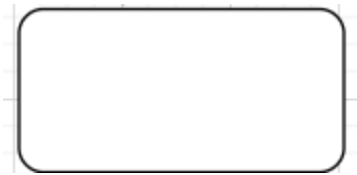
### Diamant:

Bloc utilitzat per a senyalitzar una acció. Serveix per a fer una pregunta amb resposta de si o no, s'utilitza per a tots els menús i les eleccions del jugador.



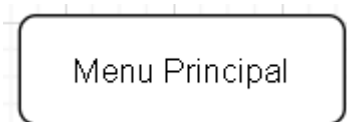
### Rectangle arrodonit:

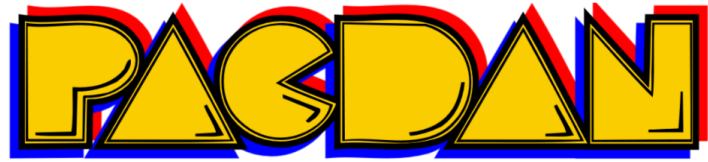
Aquest últim bloc utilitzat serveix per a indicar el resultat d'una acció. Pot ser un canvi en una variable o la mostra de text per pantalla així com una acció desencadenada a través d'una altra.



### Rectangle de "Menú Principal":

Aquest és un rectangle el qual hem utilitzat de forma una mica diferent. El que vol dir és que el bloc anterior a aquest hauria d'anar encadenat al bloc de menú principal, però, per tal de no passar línies pel mig del diagrama i que quedés lleig hem preferit utilitzar aquest mètode.





# Documentació: Entorns de desenvolupament

## RA5 – Diagrames UML

Un UML és un diagrama que representa visualment com està dissenyat un sistema, mostrant-ne les parts i com es relacionen. Al llarg d'aquest projecte hem realitzat un total de tres UML.

Per què tres UML?

- Abans: Per planificar i definir la idea inicial.
- A mig projecte: Per reflectir canvis i ajustar el disseny segons l'avanç.
- Al final: Per documentar el disseny real del projecte acabat.

## Programació

A l'hora d'utilitzar la Base de Dades en el nostre projecte de Pacman, vam arribar a la conclusió d'utilitzar-lo per emmagatzemar el nom de l'usuari i la seva puntuació amb el propòsit de tenir una taula de classificacions amb els 10 millors jugadors amb major puntuació.

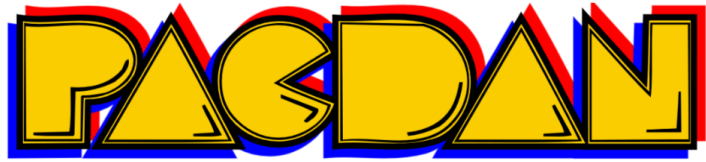
#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra
<input type="checkbox"/> 1	id	int			No	Ninguna		AUTO_INCREMENT
<input type="checkbox"/> 2	nombre	varchar(50)	utf8mb4_0900_ai_ci		Sí	NULL		
<input type="checkbox"/> 3	puntuacion	int			Sí	NULL		

## RA8-RA9 – Connexió amb BDD

Per connectar-se a la Base de Dades, requereix una connexió i els permisos requerits per accedir a aquest.

```
Class.forName(className: "com.mysql.cj.jdbc.Driver");  
Connection connection = DriverManager.getConnection(  
    url: "jdbc:mysql://dam.inspedralbes.cat/Pacman_PacmanData", user: "Pacman_Admin", password: "Wompwomp123"  
);
```





## RA8-RA9 – Inserció d'informació en la BDD

Per inserir informació a la Base de Dades, s'ha d'agafar la connexió que ja havíem creat abans per fer una declaració ("Statement"). En aquest cas, es crea una funció per inserir automàticament en nom amb la seva puntuació.

```
private static void subir(Connection c, List<String> nuevas, Set<String> enviados) throws SQLException {
    PreparedStatement ps = c.prepareStatement(
        string: "INSERT INTO puntuaciones (nombre, puntuacion) VALUES (?, ?)"
    );
    for (String entry : nuevas) {
        String[] p = entry.split(regex: ",");
        if (p.length < 2) continue;
        ps.setString(i: 1, p[0]);
        ps.setInt(i: 2, i1: Integer.parseInt(p[1]));
        ps.executeUpdate();
        enviados.add(e: entry);
    }
    ps.close();
}
```

## RA8-RA9 – Recuperació d'informació d'una BDD

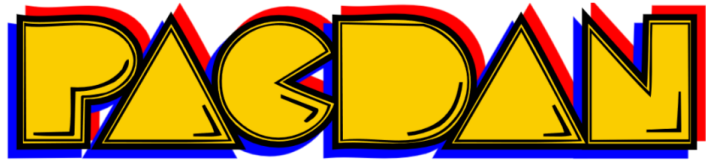
Per recuperar informació de la Base de Dades, es fa com abans, es fa una declaració a partir de la connexió amb la Base de Dades. En aquest cas, S'agafa la informació dels 10 jugadors amb la puntuació més alta.

```
private static void generarHighscore(Connection c, String outputPath) throws SQLException, IOException {
    String sql = "SELECT nombre, puntuacion FROM puntuaciones ORDER BY puntuacion DESC LIMIT 10";
    try (ResultSet rs = c.createStatement().executeQuery(string: sql);
        DataOutputStream out = new DataOutputStream(
            new BufferedOutputStream(new FileOutputStream(name: outputPath)))) {

        while (rs.next()) {
            byte[] nameBytes = rs.getString(string: "nombre").getBytes(charset: StandardCharsets.UTF_8);
            out.writeInt(v: nameBytes.length);
            out.write(b: nameBytes);
            out.writeInt(v: rs.getInt(string: "puntuacion"));
        }
        out.writeInt(v: -1);
    }
}
```

## RA8-RA9 – Actualització d'informació en una BDD

Per que la Base de Dades estigui sempre actualitzada, es comprova cada cert temps si hi ha nova informació. En cas que hi rebi de nova, l'actualitzarà i marcarà aquella informació com a actualitzada. Per tant, no s'enviarà la mateixa informació dues vegades a la base de dades.



```
while (true) {
    try {
        List<String> nuevas = leerNuevas(scorePath:SCORES_FILE, enviados);

        // Siempre generar ranking actualizado
        generarHighscore(c: connection, outputPath: HIGHSCORE_FILE);

        if (!nuevas.isEmpty()) {
            subir(c: connection, nuevas, enviados);
            guardarEnviados(sentPath: SENT_RECORD, enviados);
            generarHighscore(c: connection, outputPath: HIGHSCORE_FILE);
            System.out.println(x: "✓ Nuevas puntuaciones subidas.");
        } else {
            System.out.println(x: "✗ Sin cambios nuevos. Esperando...");
        }

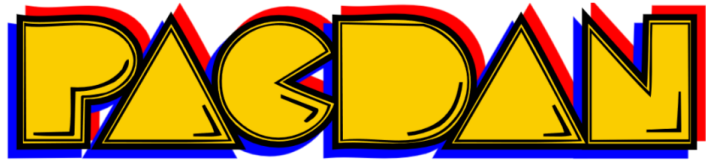
        Thread.sleep(millis:5000); // Espera 5 segundos
    }
}
```

## RA8-RA9 – Integració de la informació en un programa

Tal com vaig comentar abans, una vegada s'hagin enviat, es marca i es guarda com a informació ja enviada. Per tant, no es pot enviar més d'una vegada.

```
private static Set<String> cargarPrevios(String sentPath) throws IOException {
    Set<String> enviados = new HashSet<>();
    File f = new File(pathname: sentPath);
    if (!f.exists()) return enviados;
    BufferedReader r = new BufferedReader(new FileReader(file: f));
    String line;
    while ((line = r.readLine()) != null) enviados.add(e: line.trim());
    r.close();
    return enviados;
}

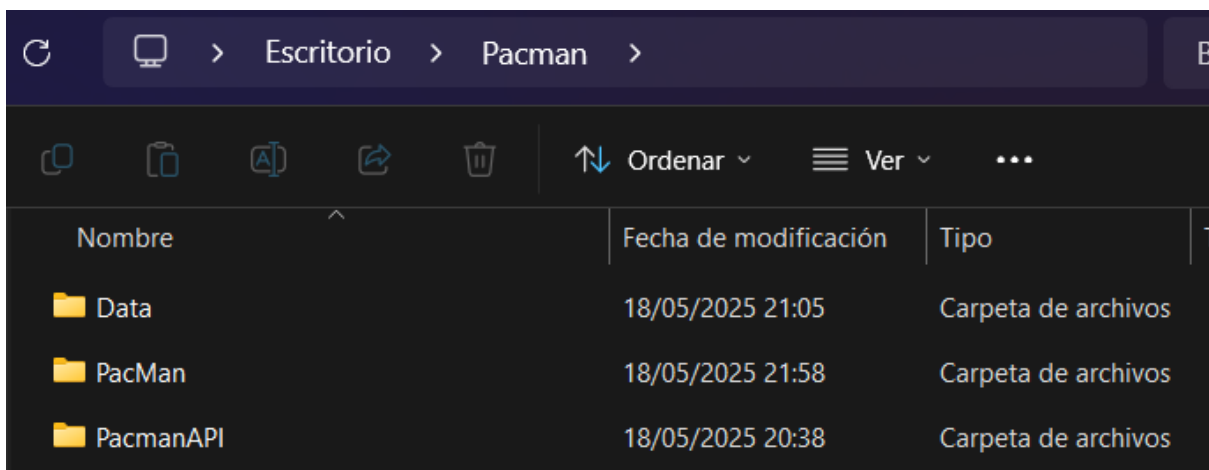
private static List<String> leerNuevas(String scorePath, Set<String> enviados) throws IOException {
    List<String> nuevas = new ArrayList<>();
    File f = new File(pathname: scorePath);
    if (!f.exists()) return nuevas;
    BufferedReader r = new BufferedReader(new FileReader(file: f));
    String line;
    while ((line = r.readLine()) != null) {
        line = line.trim();
        if (!line.isEmpty() && !enviados.contains(o: line)) nuevas.add(e: line);
    }
    r.close();
    return nuevas;
}
```



```
private static void guardarEnviados(String sentPath, Set<String> enviados) throws IOException {  
    BufferedWriter w = new BufferedWriter(new FileWriter(fileName: sentPath));  
    for (String line : enviados) {  
        w.write(str: line);  
        w.newLine();  
    }  
    w.close();  
}
```

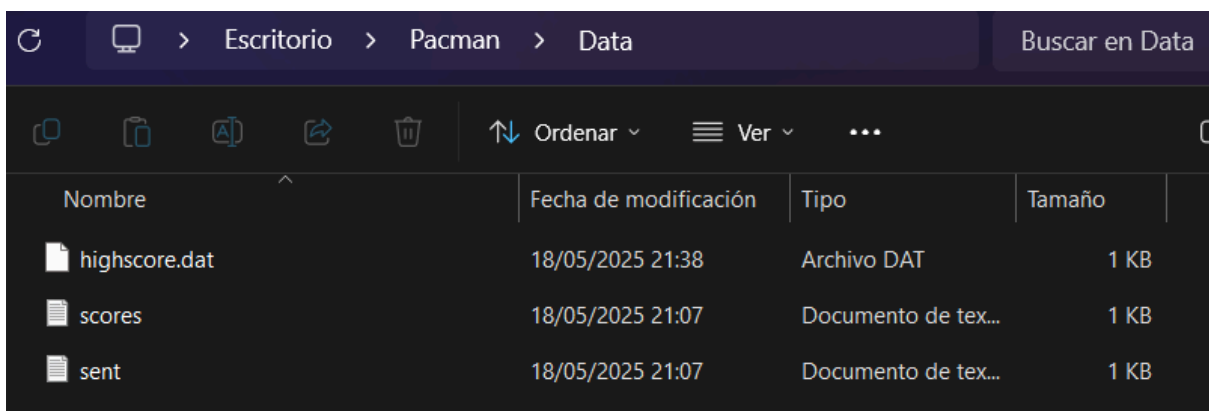
## RA5 – Entrada/sortida del llenguatge i les llibreries associades.

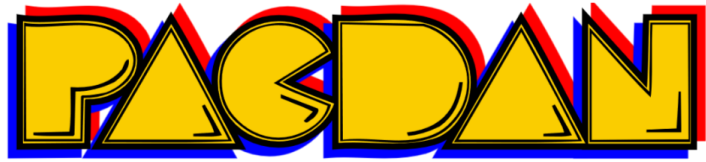
Per transportar la informació des de la BDD al Joc, es necessita una carpeta on s'enviarà la informació necessària perquè els dos programes facin el que necessitin. En aquest cas, es crearà la carpeta anomenada Data.



## RA5 – Fitxers per emmagatzemar i recuperar informació.

En cas de l'arxiu java, generarà un .dat amb les millors puntuacions per que el programa de Unity pugui llegir-les. I en cas del Unity, generarà un arxíu .txt amb les puntuacions i els noms per guardar-les a la BDD.





## RA5 – Programes que utilitzen mètodes d'accés al contingut dels fitxers.

Per accedir a aquests, java ho detecta perquè és un String. En cas del Unity, ho ha de desxifrar per a després passar-ho a String i int.

```
private void CargarHighscores()
{
    string oneUp = Path.GetDirectoryName(Application.dataPath);
    string rootDir = Path.GetDirectoryName(oneUp);
    string highDat = Path.Combine(rootDir, "Data", "highscore.dat");

    if (!File.Exists(highDat))
    {
        rankingText.text = "No hay puntuaciones aún.";
        return;
    }

    StringBuilder sb = new StringBuilder();

    using (BinaryReader br = new BinaryReader(File.Open(highDat, FileMode.Open, FileAccess.Read)))
    {
        int idx = 1;

        while (br.BaseStream.Position < br.BaseStream.Length && idx <= 10)
        {
            // 1) ¿quedan 4 bytes para la longitud?
            if (br.BaseStream.Length - br.BaseStream.Position < 4) break;

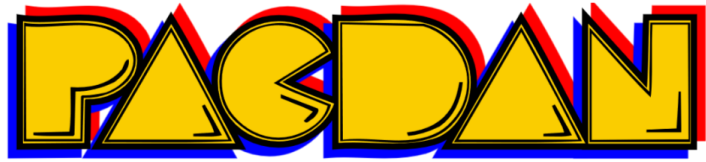
            int len = ReadInt32BigEndian(br);           // <-- big-endian

            // 2) marca fin
            if (len == -1) break;

            // 3) ¿quedan nombre + 4 bytes?
            if (br.BaseStream.Length - br.BaseStream.Position < len + 4) break;

            string nombre = Encoding.UTF8.GetString(br.ReadBytes(len));
            int puntos = ReadInt32BigEndian(br);       // <-- big-endian

            string corto = nombre.Length > 7 ? nombre[..7] : nombre;
            sb.AppendLine($"{idx}. {corto} - {puntos} pts");
            idx++;
        }
    }
}
```



```
private static List<String> leerNuevas(String scorePath, Set<String> enviados) throws IOException {
    List<String> nuevas = new ArrayList<>();
    File f = new File(pathname: scorePath);
    if (!f.exists()) return nuevas;
    BufferedReader r = new BufferedReader(new FileReader(file: f));
    String line;
    while ((line = r.readLine()) != null) {
        line = line.trim();
        if (!line.isEmpty() && !enviados.contains(o: line)) nuevas.add(e: line);
    }
    r.close();
    return nuevas;
}
```