

Documentación de Scripts de Modificación de Super Mario Bros

Este documento describe los scripts principales para modificar Super Mario Bros (NES) desarrollados en este proyecto.

Índice

1. [patch_sky_palette_final.py](#) - Modificar cielo para efecto noche
 2. [patch_mario_full.py](#) - Crear skins personalizados de Mario
 3. [mutate_chr_range_args.py](#) - Mutar tiles específicos en CHR-ROM
 4. [patch_star_invincible.py](#) - Hacer a Mario invencible permanentemente
 5. [main.py](#) - Emulador NES para probar ROMs modificadas
 6. [patch_title_text.py](#) - Modificar texto del título del juego
 7. [mario_rl_run.py](#) - Entrenar PPO y jugar con AI
-

patch_sky_palette_final.py

Descripción

Modifica específicamente los colores del cielo en Super Mario Bros para crear un efecto de "modo noche". Basado en el análisis del assembler, modifica **BackgroundColors** en la dirección **0x85cf**.

Funcionamiento Técnico

- **Ubicación:** **BackgroundColors** en **0x85cf**: **22 22 0f 0f**
- **Modificación:** Cambia ambos colores del cielo (**\$22**) por negro (**\$0f**)
- **Efecto:** El cielo del nivel 1-1 aparece oscuro como en modo noche

Uso

Sin parámetros (crea nueva ROM):

```
python3 scripts/patch_sky_palette_final.py
```

- Lee: **roms/SuperMarioBros.nes**
- Crea: **SuperMarioBros_sky_night_YYYYMMDD_HHMMSS.nes**
- **No modifica la ROM original**

Con parámetros (sobreescribe ROM):

```
python3 scripts/patch_sky_palette_final.py roms/mi_rom.nes
```

- Lee: `roms/mi_rom.nes`
- Sobreescribe: `roms/mi_rom.nes`
- **Modifica directamente la ROM especificada**

Ejemplo de Salida

```
📍 BackgroundColors en offset 0x0005DF (CPU $85cf)
🔍 Colores actuales: 22 22 0F 0F
✅ Cambios aplicados:
  Primer cielo: 0x22 → 0x0F
  Segundo cielo: 0x22 → 0x0F
📁 ROM nueva creada: roms/SuperMarioBros_sky_night_20251029_123456.nes
🌙 El cielo del nivel 1-1 ahora está oscuro!
```

patch_mario_full.py

Descripción

Modifica la paleta completa de Mario (los 4 bytes de color) para crear "skins" personalizados. Reemplaza la paleta original `[0x22, 0x16, 0x27, 0x18]` en todas sus apariciones en la ROM.

Funcionamiento Técnico

- **Paleta original:** `[0x22, 0x16, 0x27, 0x18]` (piel, gorra roja, sombra, overall azul)
- **Búsqueda:** Encuentra todas las apariciones de esta secuencia en la ROM
- **Reemplazo:** Sustituye cada aparición con la nueva paleta especificada

Uso

```
python3 scripts/patch_mario_full.py --c0 COLOR0 --c1 COLOR1 --c2 COLOR2 --c3 COLOR3
```

Parámetros (todos obligatorios):

- `--c0`: Color para slot 0 (piel/tono base de Mario)
- `--c1`: Color para slot 1 (gorra/camisa - color dominante)
- `--c2`: Color para slot 2 (contorno/sombra)
- `--c3`: Color para slot 3 (overall/pantalón)

Ejemplos de Skins

Skin "Wario":

```
python3 scripts/patch_mario_full.py --c0 0x22 --c1 0x30 --c2 0x0F --c3 0x15
```

Skin "Zombie":

```
python3 scripts/patch_mario_full.py --c0 0x29 --c1 0x15 --c2 0x0F --c3 0x27
```

Skin "Fuego":

```
python3 scripts/patch_mario_full.py --c0 0x22 --c1 0x37 --c2 0x0F --c3 0x16
```

Ejemplo de Salida

```
🔍 Encontradas 3 apariciones de la paleta Mario original
Paleta original: [34, 22, 39, 24]
Nueva paleta: [22, 48, 15, 21]

📌 Offset 0x05D7: [34, 22, 39, 24] → [34, 48, 15, 21]
📌 Offset 0x0CB7: [34, 22, 39, 24] → [34, 48, 15, 21]
📌 Offset 0x0CDB: [34, 22, 39, 24] → [34, 48, 15, 21]

✅ ROM modificada escrita en:
roms/SuperMarioBros_mario_full_22_30_0F_15_20251029_123456.nes
🎯 Skin de Mario creado con:
Piel:      0x22
Gorra:     0x30
Contorno:  0x0F
Overall:   0x15
```

mutate_chr_range_args.py

Descripción

Muta un rango específico de tiles en la CHR-ROM cambiando los índices de color **01** ↔ **10**. Permite especificar exactamente qué tiles modificar mediante parámetros de línea de comandos.

Funcionamiento Técnico

- **Técnica:** Intercambia bits de los planos de color en tiles de 8x8 píxeles
- **Efecto:** Cambia la apariencia visual de los tiles sin destruir el dibujo
- **Rango:** Permite especificar tile inicial y cantidad de tiles a modificar

Uso

```
python3 scripts/mutate_chr_range_args.py --start TILE_INICIAL --count CANTIDAD
```

Parámetros:

- **--start**: Tile inicial (hexadecimal o decimal)
- **--count**: Cantidad de tiles a modificar (hexadecimal o decimal)

Ejemplos de Uso

Mutar 256 tiles desde 0x100:

```
python3 scripts/mutate_chr_range_args.py --start 0x100 --count 0x100
```

Mutar solo los últimos 32 tiles (0x1E0–0x1FF):

```
python3 scripts/mutate_chr_range_args.py --start 0x1E0 --count 0x20
```

Mutar un solo tile (tile 0x1F8):

```
python3 scripts/mutate_chr_range_args.py --start 0x1F8 --count 1
```

Usar valores decimales:

```
python3 scripts/mutate_chr_range_args.py --start 480 --count 32
```

Ejemplo de Salida

```
✅ ROM mutada parcialmente:  
roms/SuperMarioBros_CHR_1E0_020_20251029_123456.nes  
  Tiles modificados: 0x1e0 .. 0x1ff  
👉 Cargá esa ROM y mirá el Goomba del 1-1.
```

patch_star_invincible.py

Descripción

Hackea la rutina de colisión Mario-enemigo para que Mario SIEMPRE sea tratado como si tuviera el efecto de estrella activo. Esto significa que al tocar cualquier enemigo, el enemigo muere y Mario nunca recibe daño.

Funcionamiento Técnico

- **Ubicación:** Rutina `PlayerEnemyCollision` en dirección `$D88D`
- **Técnica:** Reemplaza la instrucción `BEQ` (Branch if Equal) con `NOP NOP` (No Operation)
- **Efecto:** Bypasea la verificación del timer de estrella, haciendo que Mario siempre tenga invencibilidad

Análisis del Assembler

```
D88A: LDA StarInvincibleTimer
D88D: BEQ HandlePECollisions ; si timer = 0 -> daño normal
D88F: JMP ShellOrBlockDefeat ; si no, matar enemigo
```

Al cambiar `BEQ` por `NOP NOP`, la rutina siempre ejecuta `JMP ShellOrBlockDefeat`, haciendo que Mario mate enemigos sin recibir daño.

Uso

Sin parámetros (crea nueva ROM):

```
python3 scripts/patch_star_invincible.py
```

- Lee: `roms/SuperMarioBros.nes`
- Crea: `SuperMarioBros_star_infinite_YYYYMMDD_HHMMSS.nes`
- **No modifica la ROM original**

Con parámetros (sobreescribe ROM):

```
python3 scripts/patch_star_invincible.py roms/mi_rom.nes
```

- Lee: `roms/mi_rom.nes`
- Sobreescribe: `roms/mi_rom.nes`
- **Modifica directamente la ROM especificada**





Ejemplo de Salida

```
Antes del parche en 0xd88d:
  opcode=0xf0 arg=0x06
Aplicado parche: BEQ -> NOP,NOP
✅ ROM hackeada escrita en:
```

```
roms/SuperMarioBros_star_infinite_20251029_123456.nes
```

- Esta ROM debería hacer que, al tocar un Goomba, el Goomba muera siempre, y Mario no reciba daño nunca (efecto estrella permanente).

Efectos del Hack

-  **Mario invencible:** Nunca recibe daño al tocar enemigos
-  **Enemigos mueren:** Al tocar Mario, todos los enemigos mueren instantáneamente
-  **Efecto permanente:** No necesitas recoger estrellas, el efecto está siempre activo
-  **Compatible:** Funciona con todos los tipos de enemigos (Goombas, Koopas, etc.)

main.py

Descripción

Emulador NES integrado para probar las ROMs modificadas directamente desde Python. Utiliza la biblioteca **nes** para ejecutar ROMs de Nintendo Entertainment System con diferentes modos de sincronización.

Funcionamiento Técnico

- **Biblioteca:** Utiliza **nes** y **nes.pycore.system.NES** para emulación
- **Mappers:** Soporta Mapper 0 (estándar para Super Mario Bros)
- **Sincronización:** Configurable para diferentes velocidades de emulación
- **Modo gráfico:** Ejecuta ROMs con interfaz visual completa

Uso

```
python3 scripts/main.py ROM_PATH [--sync-mode MOD0]
```

Parámetros:

- **rom_path** (obligatorio): Ruta a la ROM de NES a ejecutar
- **--sync-mode** (opcional): Modo de sincronización (default: 2)

Ejemplos de Uso

Ejecutar ROM original:

```
python3 scripts/main.py roms/SuperMarioBros.nes
```

Ejecutar ROM modificada:

```
python3 scripts/main.py roms/SuperMarioBros_sky_night_20251029_123456.nes
```

Con modo de sincronización específico:

```
python3 scripts/main.py roms/SuperMarioBros.nes --sync-mode 1
```

Ejemplo de Salida

```
✓ Emulador NES iniciado
🎮 Cargando ROM: roms/SuperMarioBros.nes
🔄 Modo de sincronización: 2
▶ Ejecutando ROM...
```

Modos de Sincronización

- **Modo 1:** Sincronización rápida (para pruebas)
- **Modo 2:** Sincronización estándar (default)
- **Modo 3:** Sincronización precisa (para análisis)

Casos de Uso

- ✓ **Probar modificaciones:** Ejecutar ROMs modificadas por los scripts
- ✓ **Comparar resultados:** Alternar entre ROM original y modificada
- ✓ **Desarrollo:** Probar cambios en tiempo real
- ✓ **Debugging:** Verificar que las modificaciones funcionan correctamente

Requisitos

- **Biblioteca nes:** Instalada en el entorno Python
- **ROM válida:** Archivo .nes con header iNES correcto
- **Interfaz gráfica:** Para mostrar la emulación visual

patch_title_text.py

Descripción

Modifica el texto "WORLD TIME" en la pantalla de título de Super Mario Bros para mostrar un mensaje personalizado como "PYTHON MEETUP MVD". Este script busca la secuencia de bytes que corresponde al texto original y la reemplaza con el nuevo texto.

Funcionamiento Técnico

- **Búsqueda:** Localiza la secuencia de bytes `0x20, 0x52, 0x0b, 0x20, 0x18, 0x1b, 0x15, 0x0d` que corresponde a "WORLD TIME"
- **Reemplazo:** Sustituye los bytes con la secuencia del nuevo texto
- **Validación:** Verifica que el nuevo texto tenga el mismo largo que el original

- **Padding:** Si es necesario, rellena con espacios para mantener el largo

Uso

Sin parámetros (crea nueva ROM):

```
python3 scripts/patch_title_text.py
```

- Lee: `roms/SuperMarioBros.nes`
- Crea: `SuperMarioBros_titlemsg_YYYYMMDD_HHMMSS.nes`
- **No modifica la ROM original**

Con parámetros (sobrescribe ROM):

```
python3 scripts/patch_title_text.py roms/mi_rom.nes
```

- Lee: `roms/mi_rom.nes`
- Sobrescribe: `roms/mi_rom.nes`
- **Modifica directamente la ROM especificada**

Ejemplo de Salida

✅ Encontré 'WORLD TIME' en offset 0x000123
📁 ROM nueva creada: `roms/SuperMarioBros_titlemsg_20251029_123456.nes`
🔄 Esa ROM debería mostrar tu texto custom en el lugar donde iba 'WORLD TIME' en la pantalla de título.
👉 Eso sirve perfecto para la demo: queda el mensaje 'PYTHON MEETUP MVD' en pantalla principal.





Mapeo de Caracteres

El script incluye un mapeo de caracteres a tiles NES:

- **P:** `0x19`
- **Y:** `0x22`
- **T:** `0x1d`
- **H:** `0x11`
- **O:** `0x18`
- **N:** `0x17`
- **Espacio:** `0x24`
- **M:** `0x16`
- **E:** `0x0e`
- **U:** `0x1e`
- **V:** `0x1f`

- **D:** 0x0d

Casos de Uso

-  **Personalización:** Cambiar el texto del título para eventos especiales
-  **Branding:** Agregar mensajes personalizados al juego
-  **Demostraciones:** Mostrar modificaciones en presentaciones
-  **Eventos:** Crear ROMs especiales para meetups o conferencias

Notas Importantes

- **Mapeo de tiles:** Los valores de los caracteres necesitan ser verificados con la ROM específica
- **Largo del texto:** El nuevo texto debe tener el mismo largo que "WORLD TIME"
- **Ubicación:** El texto aparece en la pantalla de título del juego
- **Compatibilidad:** Funciona con ROMs estándar de Super Mario Bros

mario_rl_run.py

Descripción

Ejecuta un pipeline RL con `gym_super_mario_bros` y `stable-baselines3 (PP0)` para entrenar brevemente y luego jugar automáticamente renderizando. Incluye fixes (copias contiguas y batch dim) y silencia un warning inofensivo de overflow del entorno.

Requisitos sugeridos

```
gym==0.21.0
gym_super_mario_bros==7.3.0
nes_py==8.2.1
stable-baselines3==1.6.2
torch==1.13.1 torchvision==0.14.1 torchaudio==0.13.1
```

Uso

- Entrenar y jugar:

```
python scripts/mario_rl_run.py --seconds 20 --timesteps 10000
```

- Cargar modelo guardado y jugar (demo):

```
python scripts/mario_rl_run.py --load mario_ppo_model.zip --seconds 30
```

Flags

- `--seconds`: segundos de juego en modo demo/tras entrenar.
- `--timesteps`: timesteps de entrenamiento PPO (ignorado si `--load`).
- `--load`: ruta a `.zip` de modelo para saltar entrenamiento.
- `--save`: ruta donde guardar el modelo entrenado (default `mario_ppo_model.zip`).

Salida esperada

```
🏆 Entrenando PPO (10000 timesteps)...  
... métricas de SB3 ...  
💾 Modelo guardado en: mario_ppo_model.zip  
🕒 Tiempo de entrenamiento: 65.2s  
🎮 Jugando 20s...  
✅ Listo. Pasos jugados: 2100, ciclos completados: 1
```

Notas

- Se silencia el warning `overflow encountered in scalar subtract`; es inofensivo.
- Se usa copia contigua y batch dim en `predict` para evitar errores de strides negativos y acciones no hasheables.

Colores Comunes en Super Mario Bros:

- `0x22`: Azul/celeste claro (cielo original)
- `0x16`: Rojo (gorra original de Mario)
- `0x27`: Marrón oscuro (sombra)
- `0x18`: Azul (overall original de Mario)
- `0x30`: Verde lima (Luigi)
- `0x37`: Amarillo (Mario fuego)
- `0x0F`: Negro
- `0x29`: Verde claro
- `0x15`: Rojo oscuro

Rango de Colores NES:

- **Válido**: `0x00` - `0x3F` (64 colores)
- **Típico**: `0x00` - `0x3F` para paletas PPU
- **Extendido**: `0x00` - `0xFF` para algunos casos especiales

🔧 Requisitos Técnicos

Archivos Necesarios:

- `roms/SuperMarioBros.nes` (ROM original)
- Python 3.6+
- Módulos: `os`, `argparse`, `datetime`

Estructura de Archivos:

```
charla/
├── scripts/
│   ├── patch_sky_palette_final.py
│   ├── patch_mario_full.py
│   ├── mutate_chr_range_args.py
│   ├── patch_star_invincible.py
│   ├── patch_title_text.py
│   └── main.py
└── roms/
    ├── SuperMarioBros.nes
    └── [ROMs generadas]
```

Flujo de Trabajo Recomendado

1. **Copia de seguridad:** Siempre mantén una copia de [SuperMarioBros.nes](#)
2. **Modificaciones incrementales:** Usa ROMs específicas para cada modificación
3. **Pruebas:** Carga cada ROM modificada en el emulador para verificar el resultado
4. **Combinaciones:** Puedes aplicar múltiples modificaciones en secuencia

Ejemplo de Flujo:

```
# 1. Crear ROM con cielo nocturno
python3 scripts/patch_sky_palette_final.py

# 2. Probar la ROM modificada
python3 scripts/main.py roms/SuperMarioBros_sky_night_20251029_123456.nes

# 3. Usar esa ROM para crear skin de Mario
python3 scripts/patch_mario_full.py --c0 0x22 --c1 0x30 --c2 0x0F --c3
0x15 roms/SuperMarioBros_sky_night_20251029_123456.nes

# 4. Probar la ROM con skin de Mario
python3 scripts/main.py
roms/SuperMarioBros_mario_full_22_30_0F_15_20251029_123456.nes

# 5. Hacer a Mario invencible en la ROM resultante
python3 scripts/patch_star_invincible.py
roms/SuperMarioBros_mario_full_22_30_0F_15_20251029_123456.nes

# 6. Modificar el texto del título
python3 scripts/patch_title_text.py
roms/SuperMarioBros_star_infinite_20251029_123456.nes

# 7. Probar la ROM final con todas las modificaciones
python3 scripts/main.py roms/SuperMarioBros_tittlemsg_20251029_123456.nes
```



Notas Importantes

- **Backup:** Siempre mantén copias de seguridad de tus ROMs originales
- **Compatibilidad:** Estos scripts están diseñados específicamente para Super Mario Bros (NES)
- **Emulador:** Usa un emulador compatible con ROMs modificadas para probar los resultados
- **Experimentación:** Los colores pueden variar según el emulador y configuración de paleta

Documentación generada para el proyecto de modificación de Super Mario Bros