

SUPER QUAKE 64

Super Mario 64

- Surement le jeu le plus analysé de toute l'histoire du jeu vidéo
 - “ But First We Need to Talk About Parallel Universes ...”

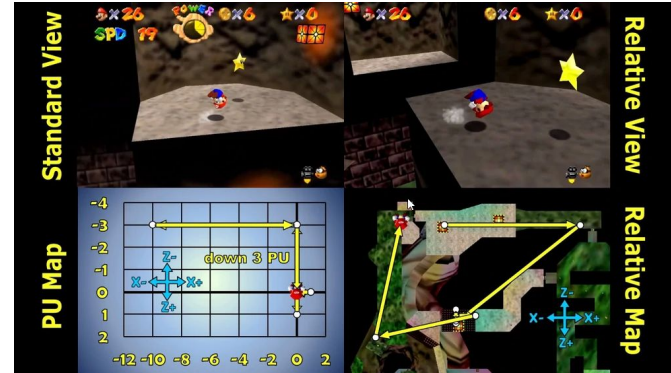
Super Mario 64

- Surely the game the most analyzed of all video game history
 - “ But First We Need to Talk About Parallel Universes ... ”



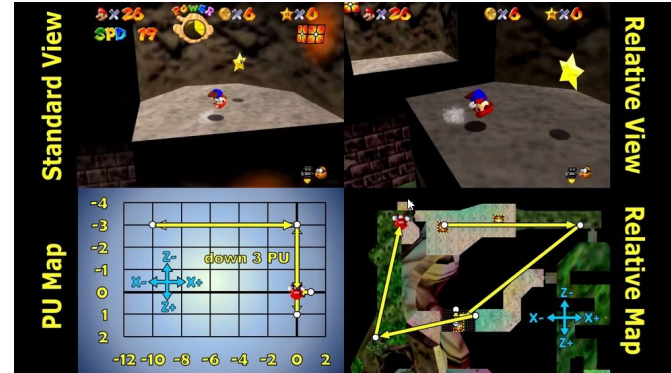
Super Mario 64

- Surely the most analyzed game of all video game history
 - “ But First We Need to Talk About Parallel Universes ... ”
 - The game was completely decompiled



Super Mario 64

- Surement le jeu le plus analysé de toute l’histoire du jeu vidéo
 - “ But First We Need to Talk About Parallel Universes ... ”
 - Le jeu a été entièrement décompilé
 - Pas d’ACE :(



libsm64

- La physique du jeu et le rendu de Mario dans une bibliothèque

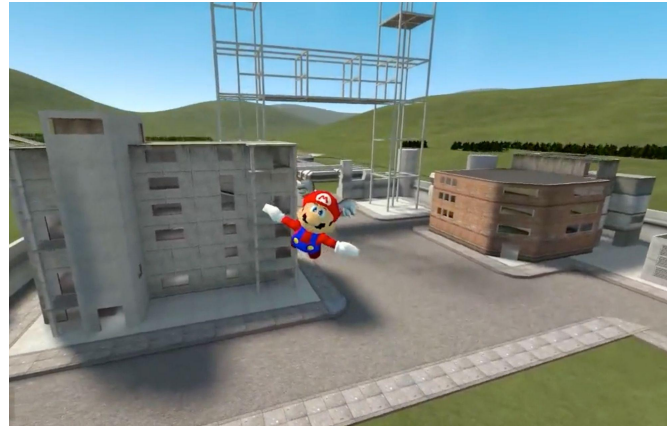
libsm64

- La physique du jeu et le rendu de Mario dans une bibliothèque



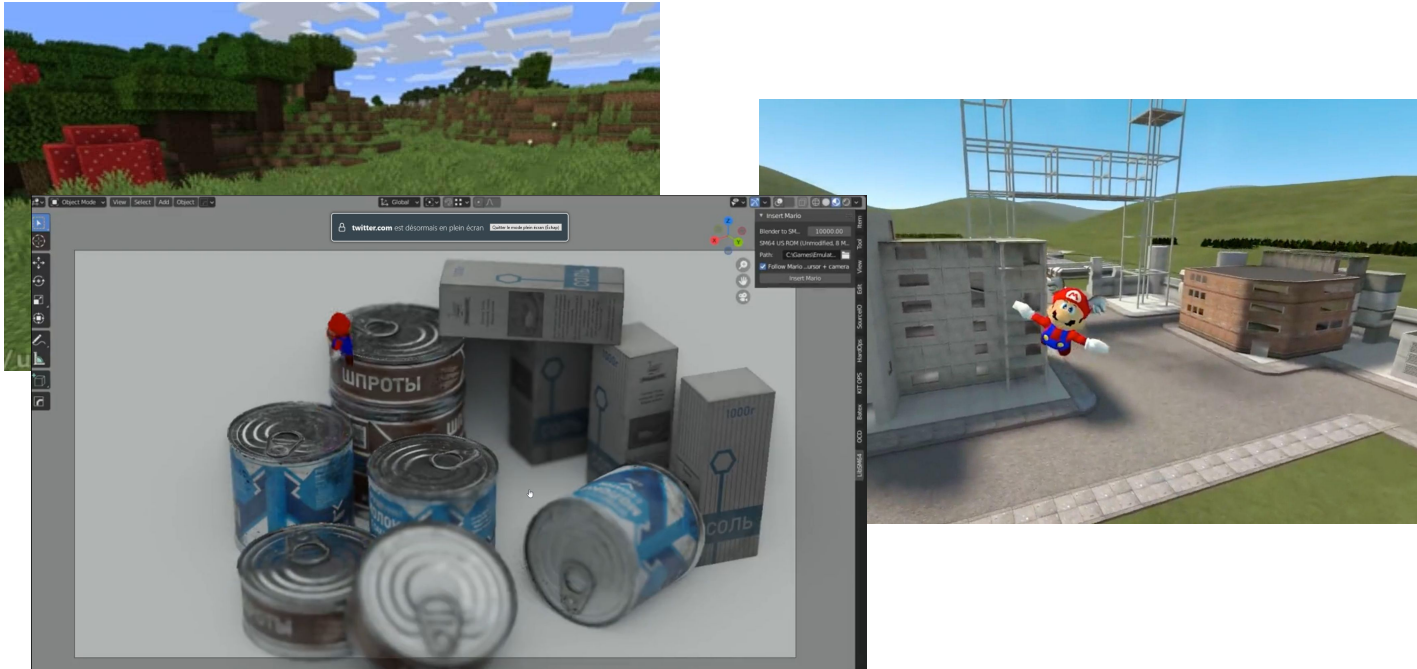
libsm64

- La physique du jeu et le rendu de Mario dans une bibliothèque



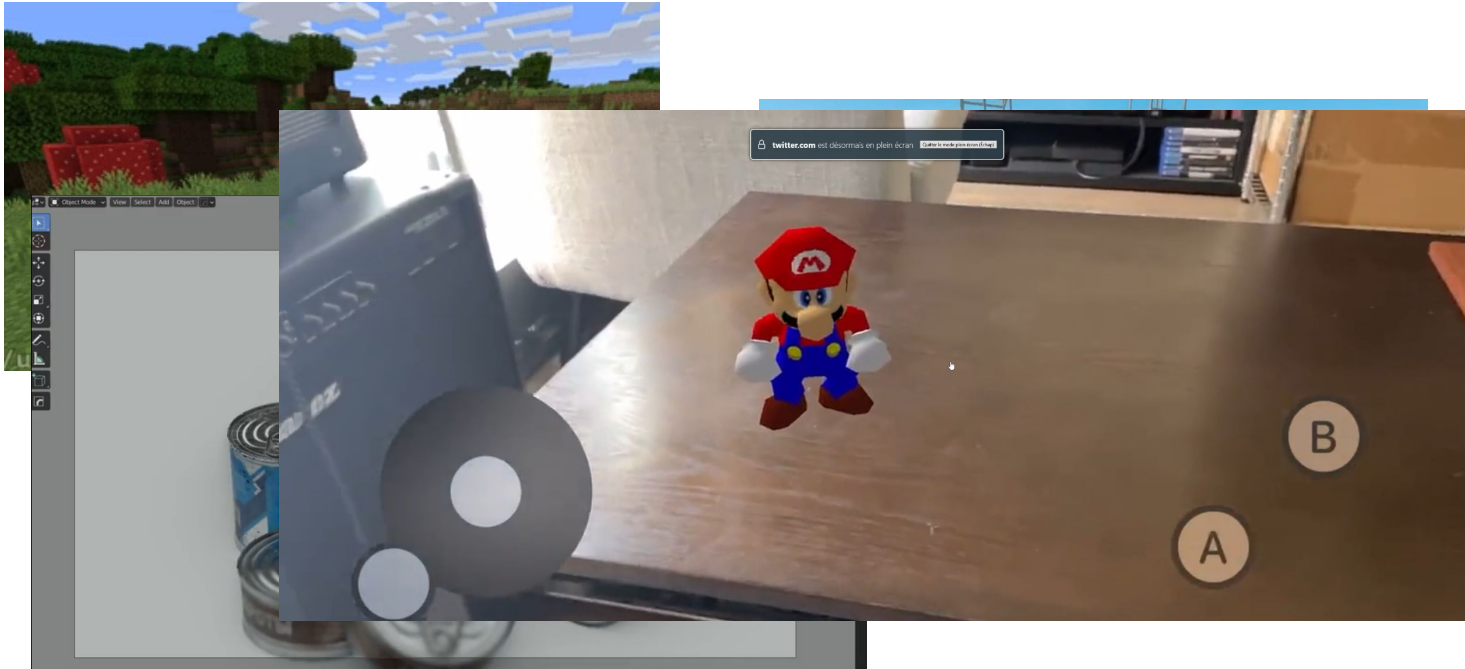
libsm64

- La physique du jeu et le rendu de Mario dans une bibliothèque



libsm64

- La physique du jeu et le rendu de Mario dans une bibliothèque



Comment ça marche

- On clone le repo <https://github.com/libsm64/libsm64>

Comment ça marche

- On clone le repo <https://github.com/libsm64/libsm64>
- On fournit une rom Super Mario 64

Comment ça marche

- On clone le repo <https://github.com/libsm64/libsm64>
- On fournit une rom Super Mario 64
- On obtient un(e) dll/so

Comment ça marche

- On clone le repo <https://github.com/libsm64/libsm64>
- On fournit une rom Super Mario 64
- On obtient un(e) dll/so
- Fonctions importantes:
 - `sm64_static_surfaces_load(surfaces, surfaces_count);`

Comment ça marche

- On clone le repo <https://github.com/libsm64/libsm64>
- On fournit une rom Super Mario 64
- On obtient un(e) dll/so
- Fonctions importantes:
 - `sm64_static_surfaces_load(surfaces, surfaces_count);`
 - `sm64_mario_tick(mariold, &marioInputs, &marioState, &marioGeometry);`

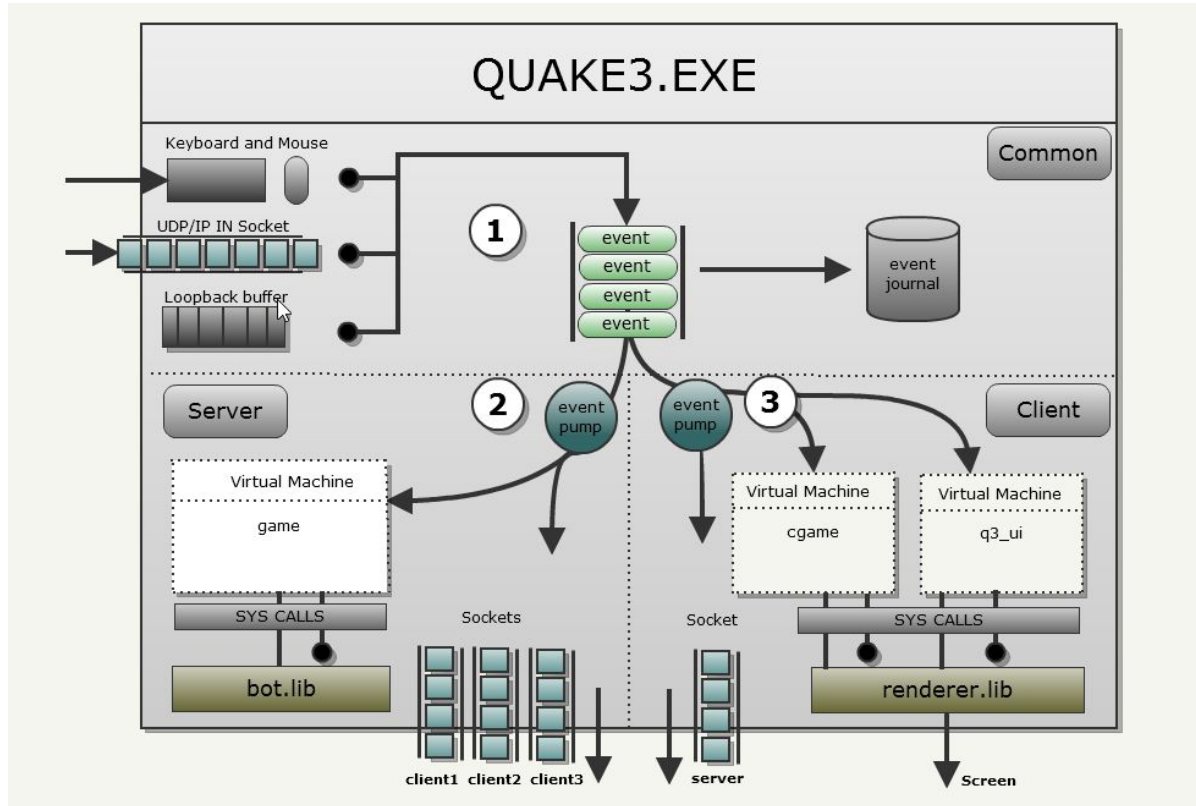
Quake 3

- Sortie en 1999, open source en 2005

Quake 3

- Sortie en 1999, open source en 2005
- le Fork ioquake3 est le mieux maintenu actuellement

Architecture Quake3 (source: Fabien Sanglard)



Compilation pour la VM

- Langage source : C

Compilation pour la VM

- Langage source : C
- q3lcc, q3asm fourni

Compilation pour la VM

- Langage source : C
- q3lcc, q3asm fourni
- Dérivé de lcc, un compilateur C pédagogique

Compilation pour la VM

- Langage source : C
- q3lcc, q3asm fourni
- Dérivé de lcc, un compilateur C pédagogique
- Compilateur limité

Compilation pour la VM

- Langage source : C
- q3lcc, q3asm fourni
- Dérivé de lcc, un compilateur C pédagogique
- Compilateur limité
- Pas de stdlib: pas de malloc, ... :(

Archi client/serveur

- On met la partie Simulation dans le serveur

Archi client/serveur

- La partie Simulation dans le serveur
- La partie captation des inputs et rendu sur le client

Archi client/serveur

- La partie Simulation dans le serveur
- La partie captation des inputs et rendu sur le client
- Partage d'info VM->serveur, serveur->client, ... pas forcément évident

On met tout dans la partie client :S

- Dans `cl_clgame.c` on a le gestionnaire de “syscalls” de la VM coté client

On met tout dans la partie client :S

- Dans `cl_clgame.c` on a le gestionnaire de “syscalls” de la VM coté client
- On initialise dans la fonction `cl_initcgame`

On met tout dans la partie client :S

- Dans `cl_clgame.c` on a le gestionnaire de “syscalls” de la VM coté client
- On initialise dans la fonction `cl_initcgame`
- On parasite l’appel à `RenderScene` pour insérer notre Mario