

Le logiciel Open Source



Agenda

- **Pourquoi l'open source ?**
 - Notions préliminaires
 - langage C, langage machine (code objet)
 - à quoi servent les compilateurs ?
 - backdoor, cc, double compilation
- **Définition(s) de l'open source**
 - Selon la "Open Source Initiative"
 - Exemple: FFmpeg
 - Exemple: Linux (Linux vs linux, Android vs iOS)
- **Licences**
 - GPL/LGPL
 - MIT/BSD
 - Hybride
 - Etude de licences non libre (Unity, Unreal)
- **GIT et pratique de l'open source**
 - le versioning ?
 - forker, gérer les contributions
 - contribution GPL sur logiciel à license hybride
- **Retour d'expérience**
 - Blender et BabylonJS
 - ImGui
 - BGFX
- **Les nouveaux défis de l'AI**
 - Github Copilot, exemple, comment ça fonctionne ?
 - Data mining, y'a-t-il eu des précédents ?
 - Copilot peut-il produire du code GPL ?
 - Copilot peut-il produire du code de qualité ?
 - Analogie avec AI Stable Diffusion

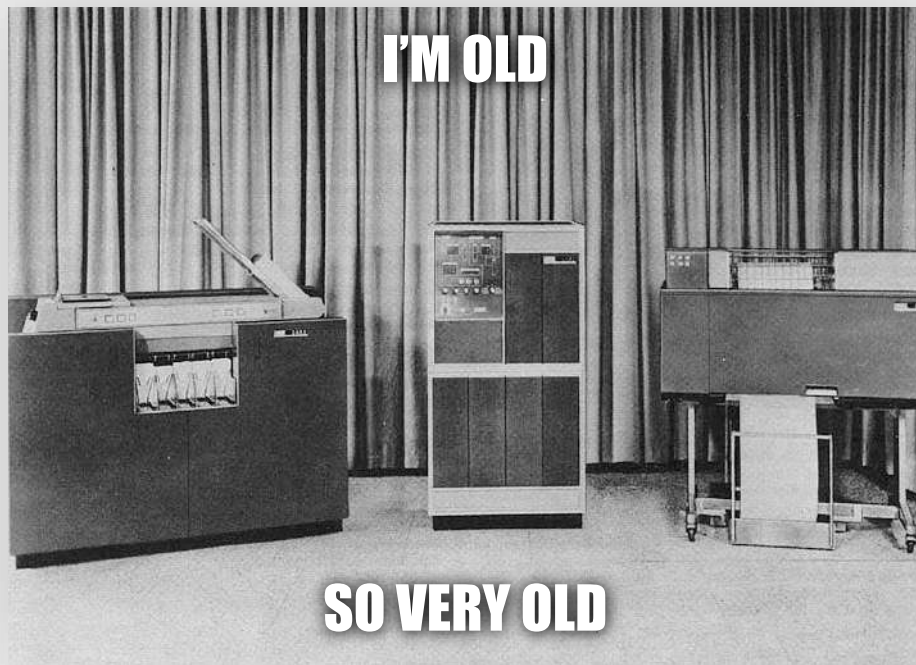
Open Source ?

“C’est la liberté d'accéder au code source, de le modifier et de le redistribuer, pour le rendre disponible à d'autres personnes, qui peuvent ensuite en faire ce qu'elles veulent”

GNU Foundation

Accéder au code source, pourquoi ?

Avec l'informatique "moderne" est apparu...



```
CFFAEDFE 0C000001 00000000 02000000 10000000
E8020000 85002000 00000000 19000000 48000000
5F5F5041 47455A45 524F0000 00000000 00000000
00000000 00000000 01000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000
00000000 19000000 E8000000 5F5F5445 58540000
00000000 00000000 00000000 01000000 00400000
00000000 00000000 00000000 00400000 00000000
05000000 05000000 02000000 00000000 5F5F7465
78740000 00000000 00000000 5F5F5445 58540000
00000000 00000000 543F0000 01000000 5C000000
00000000 543F0000 02000000 00000000 00000000
00040000 00000000 00000000 00000000 5F5F756E
77696E64 5F696E66 6F000000 5F5F5445 58540000
00000000 00000000 B03F0000 01000000 50000000
00000000 B03F0000 02000000 00000000 00000000
00000000 00000000 00000000 00000000 19000000
48000000 5F5F4C49 4E4B4544 49540000 00000000
00400000 01000000 00400000 00000000 00400000
00000000 E2010000 00000000 01000000 01000000
00000000 00000000 34000000 10000000 00400000
38000000 33000000 10000000 38400000 38000000
02000000 18000000 00000000 00000000 00000000
38000000 33000000 10000000 38400000 38000000
```

... le **code objet**

(ARM64)

Le **code objet**, fait pour la machine n'est pas (très) lisible par un humain

```
CFFAEDFE 0C000001 00000000 02000000 10000000 E8020000 85002000 00000000
19000000 48000000 5F5F5041 47455A45 524F0000 00000000 00000000 00000000
00000000 01000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 19000000 E8000000 5F5F5445 58540000 00000000 00000000
00000000 01000000 00400000 00000000 00000000 00000000 00400000 00000000
05000000 05000000 02000000 00000000 5F5F7465 78740000 00000000 00000000
5F5F5445 58540000 00000000 00000000 543F0000 01000000 5C000000 00000000
543F0000 02000000 00000000 00000000 00040080 00000000 00000000 00000000
5F5F756E 77696E64 5F696E66 6F000000 5F5F5445 58540000 00000000 00000000
B03F0000 01000000 50000000 00000000 B03F0000 02000000 00000000 00000000
00000000 00000000 00000000 00000000 19000000 48000000 5F5F4C49 4E4B4544
49540000 00000000 00400000 01000000 00400000 00000000 00400000 00000000
E2010000 00000000 01000000 01000000 00000000 00000000 34000080 10000000
00400000 38000000 33000080 10000000 38400000 38000000 02000000 18000000
78400000 03000000 A8400000 28000000 0B000000 50000000 00000000 00000000
00000000 03000000 03000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0E000000 20000000 0C000000 2F757372 2F6C6962 2F64796C 64000000 00000000
1B000000 18000000 30590A79 8A713965 824AE5F5 139322E4 32000000 20000000
01000000 00000C00 00030C00 01000000 03000000 0000FA02 2A000000 10000000
00000000 00000000 28000080 18000000 8C3F0000 00000000 00000000 00000000
0C000000 38000000 18000000 02000000 03641F05 00000100 2F757372 2F6C6962
2F6C6962 53797374 656D2E42 2E64796C 69620000 00000000 26000000 10000000
70400000 08000000 29000000 10000000 78400000 00000000 1D000000 10000000
D0400000 12010000 00000000 00000000 00000000 00000000 00000000 00000000
```

Code Objet

(ARM64)

Le langage machine a donc été inventé pour rendre ce code plus lisible...

```
sub    sp, sp, #0x10
str    w0, [sp]
ldr    w9, [sp]
mov    w8, #0xF
mul    w9, w9, w8
ldr    w8, [sp]
mul    w10, w9, w8
ldr    w9, [sp]
mov    w8, #7
mul    w8, w9, w8
add    w8, w10, w8
sub    w0, w8, #3
mov    w0, w0
add    sp, sp, #0x10
ret
```

assembleur

```
CFFAEDFE 0C000001 00000000 02000000 10000000 E8020000 85002000 00000000
19000000 48000000 5F5F5041 47455A45 524F0000 00000000 00000000 00000000
00000000 01000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 19000000 E8000000 5F5F5445 58540000 00000000 00000000
00000000 01000000 00400000 00000000 00000000 00000000 00400000 00000000
05000000 05000000 02000000 00000000 5F5F7465 78740000 00000000 00000000
5F5F5445 58540000 00000000 00000000 543F0000 01000000 5C000000 00000000
543F0000 02000000 00000000 00000000 00040000 00000000 00000000 00000000
5F5F756E 77696E64 5F696E66 6F000000 5F5F5445 58540000 00000000 00000000
B03F0000 01000000 50000000 00000000 B03F0000 02000000 00000000 00000000
00000000 00000000 00000000 00000000 19000000 48000000 5F5F4C49 4E4B4544
49540000 00000000 00400000 01000000 00400000 00000000 00400000 00000000
E2010000 00000000 01000000 01000000 00000000 00000000 34000000 10000000
00400000 38000000 33000000 10000000 38400000 38000000 02000000 18000000
78400000 03000000 A8400000 28000000 0B000000 50000000 00000000 00000000
00000000 03000000 03000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0E000000 20000000 0C000000 2F757372 2F6C6962 2F64796C 64000000 00000000
1B000000 18000000 30590A79 8A713965 824AE5F5 139322E4 32000000 20000000
01000000 00000C00 00030C00 01000000 03000000 0000FA02 2A000000 10000000
00000000 00000000 28000000 18000000 8C3F0000 00000000 00000000 00000000
0C000000 38000000 18000000 02000000 03641F05 00000100 2F757372 2F6C6962
2F6C6962 53797374 656D2E42 2E64796C 69620000 00000000 26000000 10000000
70400000 08000000 29000000 10000000 78400000 00000000 1D000000 10000000
D0400000 12010000 00000000 00000000 00000000 00000000 00000000 00000000
```

Langage machine

Code Objet

(ARM64)

Sauf qu'il n'existe pas un mais des langages machine !

```
push    rbp
mov     rbp, rsp
sub     rsp, 16
mov     DWORD PTR [-16+rbp], edi
mov     eax, DWORD PTR [-16+rbp]
imul    eax, eax, 15
add     eax, 7
imul    eax, DWORD PTR [-16+rbp]
add     eax, -3
leave
ret
```



(Intel/AMD)

```
sub     sp, sp, #0x10
str     w0, [sp]
ldr     w9, [sp]
mov     w8, #0xF
mul     w9, w9, w8
ldr     w8, [sp]
mul     w10, w9, w8
ldr     w9, [sp]
mov     w8, #7
mul     w8, w9, w8
add     w8, w10, w8
sub     w0, w8, #3
mov     w0, w0
add     sp, sp, #0x10
ret
```



(ARM)

```
stw 3, -12(1)
lwz 4, -12(1)
mullw 3, 4, 4
slwi 5, 3, 4
sub 3, 5, 3
slwi 5, 4, 3
sub 4, 5, 4
add 3, 3, 4
addi 3, 3, -3
extsw 3, 3
blr
.long 0
.quad 0
```



(PowerPC)

```
addiu $sp, $sp, -8
sw $fp, 4($sp)
move $fp, $sp
sw $4, 8($fp)
lw $2, 8($fp)
nop
mult $2, $2
mflo $3
move $2, $3
sll $2, $2, 4
subu $4, $2, $3
lw $3, 8($fp)
nop
move $2, $3
sll $2, $2, 3
subu $2, $2, $3
addu $2, $4, $2
addiu $2, $2, 8
move $2, $2
```



(MIPS)

C'est pour ça qu'ont été créés les langages haut niveau...

```
// Polynomial expression
int func(int x) {
    return (15 * x * x)+(7 * x) - 3;
}
```

```
sub    sp, sp, #0x10
str    w0, [sp]
ldr    w9, [sp]
mov    w8, #0xF
mul    w9, w9, w8
ldr    w8, [sp]
mul    w10, w9, w8
ldr    w9, [sp]
mov    w8, #7
mul    w8, w9, w8
add    w8, w10, w8
sub    w0, w8, #3
mov    w0, w0
add    sp, sp, #0x10
ret
```

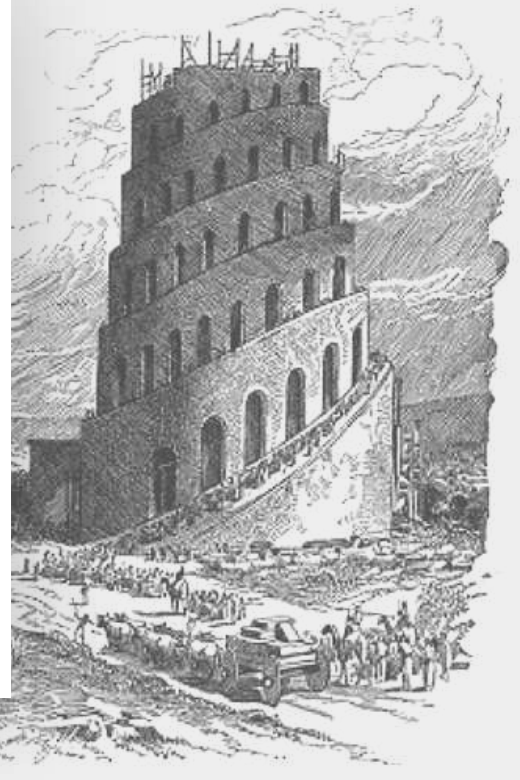
```
CFFAEDE 0C000001 00000000
02000000 10000000 E8020000
85002000 00000000 19000000
48000000 5F5F5041 47455A45
524F0000 00000000 00000000
00000000 00000000 01000000
00000000 00000000 00000000
00000000 00000000 00000000
00000000 00000000 19000000
E8000000 5F5F5445 58540000
00000000 00000000 00000000
01000000 00400000 00000000
00000000 00000000 00400000
00000000 05000000 05000000
02000000 00000000 5F5F7465
78740000 00000000 00000000
5F5F5445 58540000 00000000
00000000 543F0000 01000000
5C000000 00000000 543F0000
02000000 00000000 00000000
00040000 00000000 00000000
00000000 5F5F756E 7769E64
5F696E66 6F000000 5F5F5445
58540000 00000000 00000000
B03F0000 01000000 50000000
00000000 B03F0000 02000000
00000000 00000000 00000000
00000000 00000000 00000000
19000000 48000000 5F5F4C49
4E4B4544 49540000 00000000
00400000 01000000 00400000
00000000 00400000 00000000
E2010000 00000000 01000000
01000000 00000000 00000000
34000000 10000000 00400000
38000000 33000000 10000000
38400000 38000000 02000000
18000000 78400000 03000000
A8400000 28000000 00000000
50000000 00000000 00000000
00000000 03000000 03000000
00000000 00000000 00000000
00000000 00000000 00000000
```

compilateur

assembleur

Langage "C"

(ARM64)



Un logiciel distribué_(installé) sous forme de **code objet** est **fermé**

S'il est fermé, l'utilisateur ne peut pas exclure la présence d'une porte dérobée (ou *backdoor*), qui peut se cacher...

- ...dans le source du logiciel
- dans le code objet du logiciel
- dans le source du compilateur
- dans le code objet du compilateur¹



[1] "Reflections on Trusting Trust" / <https://users.ece.cmu.edu/~ganger/712.fall02/papers/p761-thompson.pdf>

[2] "Amazon Erases Orwell Books From Kindle" / <https://www.nytimes.com/2009/07/18/technology/companies/18amazon.html>

Un logiciel (embarqué) distribué sous forme de **code objet** est **fermé**

S'il est fermé, seul le fabricant a la main sur les évolutions...
Apparaît le risque d'enfermement propriétaire (ou *vendor lock-in*)



L'open source propose ainsi de **distribuer** le logiciel sous la forme de **code source**, pour garantir à l'utilisateur le droit d'en **étudier** le fonctionnement...

... mais aussi de le modifier, de l'améliorer
ou de le redistribuer.

Ce que dit l'Open Source Initiative

- Liberté de **vendre** ou de donner
- Liberté d'accès au code source
- Liberté de créer des œuvres dérivées
- Liberté de **redistribuer** le code source modifié
- Liberté d'utilisation par toute personne ou groupe
- Liberté d'utilisation pour **tout champ d'application**
- La licence ne doit pas être spécifique à un produit
- La licence ne doit pas restreindre d'autres logiciels
- La licence doit être **technologiquement agnostique**



L'exemple de FFmpeg



... et ça vient de France !



La magie de Netflix doit beaucoup à FFmpeg...

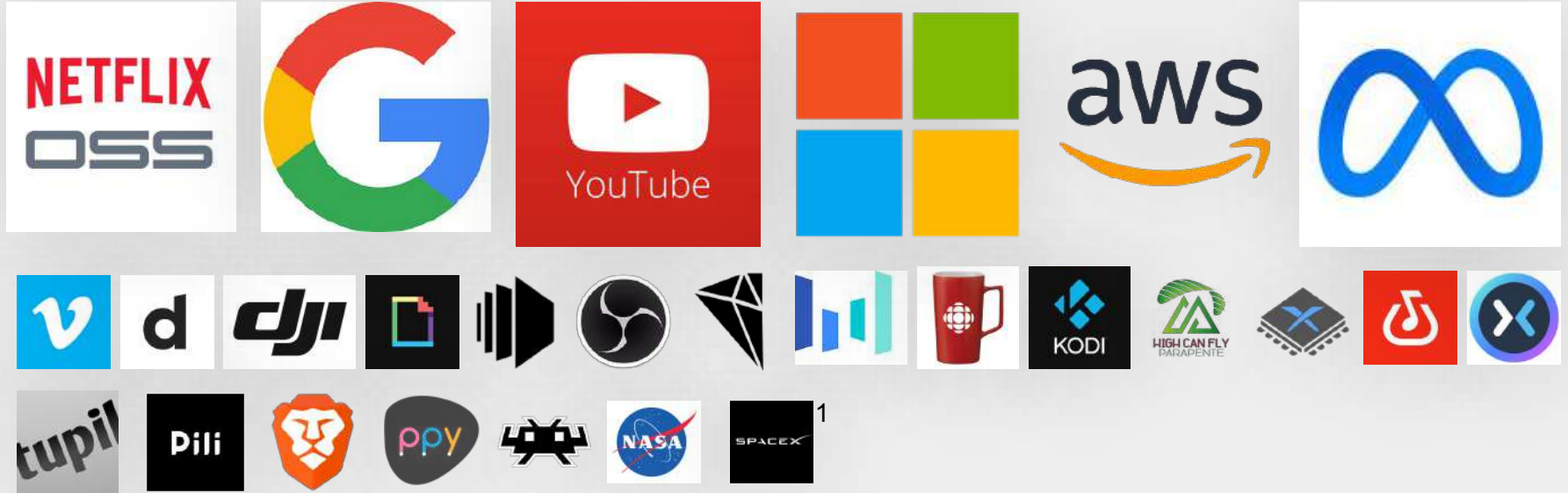
- FFmpeg compresse / décompresse tous les flux vidéos
- Pour distribuer les vidéos des fournisseurs de contenus
- Chaque titre est compressé avec des réglages spécifiques
- Chaque vidéo est compressée par segments
 - Compression en // sur des instances AWS
 - Compression de chaque segment à des débits différents
- Automatisation massive
 - Recompression de tout le contenu de Netflix à la demande

Bénéfice(s) clients

- Qualité d'image hors du commun
- Réactivité "à la demande"
- Colle aux évolutions technologiques (coucou M. le DVD)



“Quelques” fournisseurs *mainstream* utilisent FFmpeg...



dixit Github

[1] <https://www.nasaspacesflight.com/2014/06/recovering-falcon-9-ocean-landing-video-done/> / <https://aeroquartet.com/wordpress/2014/05/07/spacex-falcon-first-stage-landing-pictures-are-from-us/>

881 code results or view [all results on GitHub](#)

 youtube/cobalt_sandbox
starboard/shared/ffmpeg/ffmpeg_demuxer.cc

```
13 // limitations under the License.
14
15 #include "starboard/shared/ffmpeg/ffmpeg_dem
16
17 #include <memory>
18
19 #include "starboard/shared/ffmpeg/ffmpeg_dis
20 #include "starboard/time.h"
21
22 namespace starboard {
```

 C++ Showing the top two matches Last indexed on 30 Sep

 youtube/cobalt_sandbox
starboard/shared/ffmpeg/BUILD.gn

```
13 # limitations under the License.
14
15 ffmpeg_specialization_sources = [
16     "ffmpeg_audio_decoder_impl.cc",
17     "ffmpeg_audio_decoder_impl.h",
```

Showing the top six matches Last indexed 11 days ago

um/media/ffmpeg/ffmpeg_regre

475 code results or view [all results on GitHub](#)

 microsoft/playwright
tests/playwright-test/playwright.spec.ts

```
21 import { registry } from '../..packages/pl
22
23 const ffmpeg = registry.findExecutable('ffm
...
29 constructor(fileName: string) {
30     const output = spawnSync(ffmpeg, ['-i',
    `${fileName}-%03d.png']).stderr.toString();
```

 TypeScript Showing the top three matches Last indexed 2

 microsoft/vcpkg
versions/baseline.json

```
2293     "port-version": 1
2294     },
2295     "fdlibm": {
2296         "baseline": "5.3",
2297         "port-version": 5
2298     },
2299     "ffmpeg": {
```

Showing the top match Last indexed 2 days ago

txt

ost:x64-windows=pass

ost:x86-windows=pass

mpeg:arm-uwp=pass

mpeg:arm64-windows=pass

50 code results or view [all results on GitHub](#)

 facebook/ThreatExchange
python-threatexchange/setup.py

```
32 # Note that without ffmpeg (for vpdq) you ma
33 extras_require["dev"] = extras_require["all"]
```

 Python Showing the top match Last indexed 27 days ago

 facebook/ThreatExchange
vpdq/python/vpdq.pyx

```
12 from libcpp.string cimport string
13
14 try:
15     # A call to check if ffmpeg is installed
16     # FFMPEG is required to compute vPDQ has
17     subprocess.check_call(["ffmpeg", "-L"],
        stderr=subprocess.DEVNULL)
```

 Cython Showing the top three matches Last indexed on 11 A

 facebook/ThreatExchange



ment(

egPath",

="FFMPEG_PATH",

pecific path to ffmpeg you wa

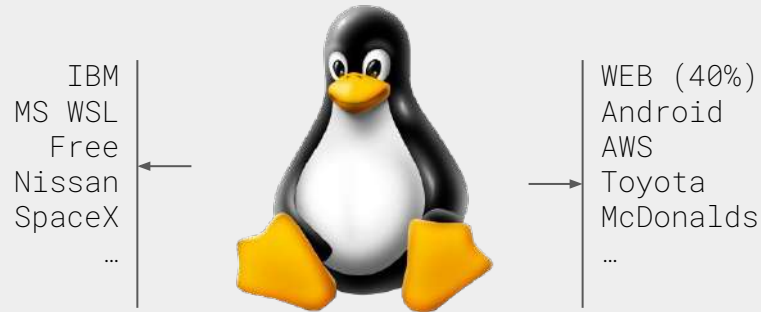
="ffmpeg",

ment(



YouTube

L'exemple de Linux



... l'open source joue un rôle essentiel dans nos vies, sans que parfois l'on s'en rende compte !

Différents types de licences Open Source

licences “Intransigeantes”

GPL/LGPL



licences “permissives”

MIT/BSD



Un projet sous MIT peut passer sous GPL. Un projet sous GPL ne peut pas passer sous MIT.

Il existe plusieurs dizaines de licences différentes, chacune adaptée à des besoins / projets spécifiques.

Le choix de la licence peut avoir un impact sur l'adhésion au projet.

licences “hybride”

GPL/LGPL/Commerciale



Le projet peut être utilisé dans un cadre académique ou dans un projet de recherche (GPL)

Le projet peut être utilisé dans un cadre commercial avec des **contraintes d'intégration** (LGPL)

Le projet peut être utilisé dans un cadre commercial en mode “privatif” (Licence commerciale)

Exemple de licence **privée**



A travers ses différentes licences, Unity se donne le droit de collecter toute donnée relative à l'usage du logiciel, ainsi que des informations sur les utilisateurs, et de **les envoyer à un tiers**

Unity procède à la collecte de statistiques sur l'usage du logiciel ET sur l'usage qui est fait des applications développées et compilées avec le logiciel **ainsi que l'usage des données** manipulées par les utilisateurs finaux.

La **confidentialité des usages** n'est pas garantie

Vos applications sont soumises aux lois Américaines sur **l'embargo**
(qui porte sur la Crimée, l'Iran, la Corée du Nord, ...)

Version Control et pratique de l'Open Source

(RCS, SVN, Mercurial, **GIT**, Perforce, Alien Brain ...)

Il n'y a pas que GIT!

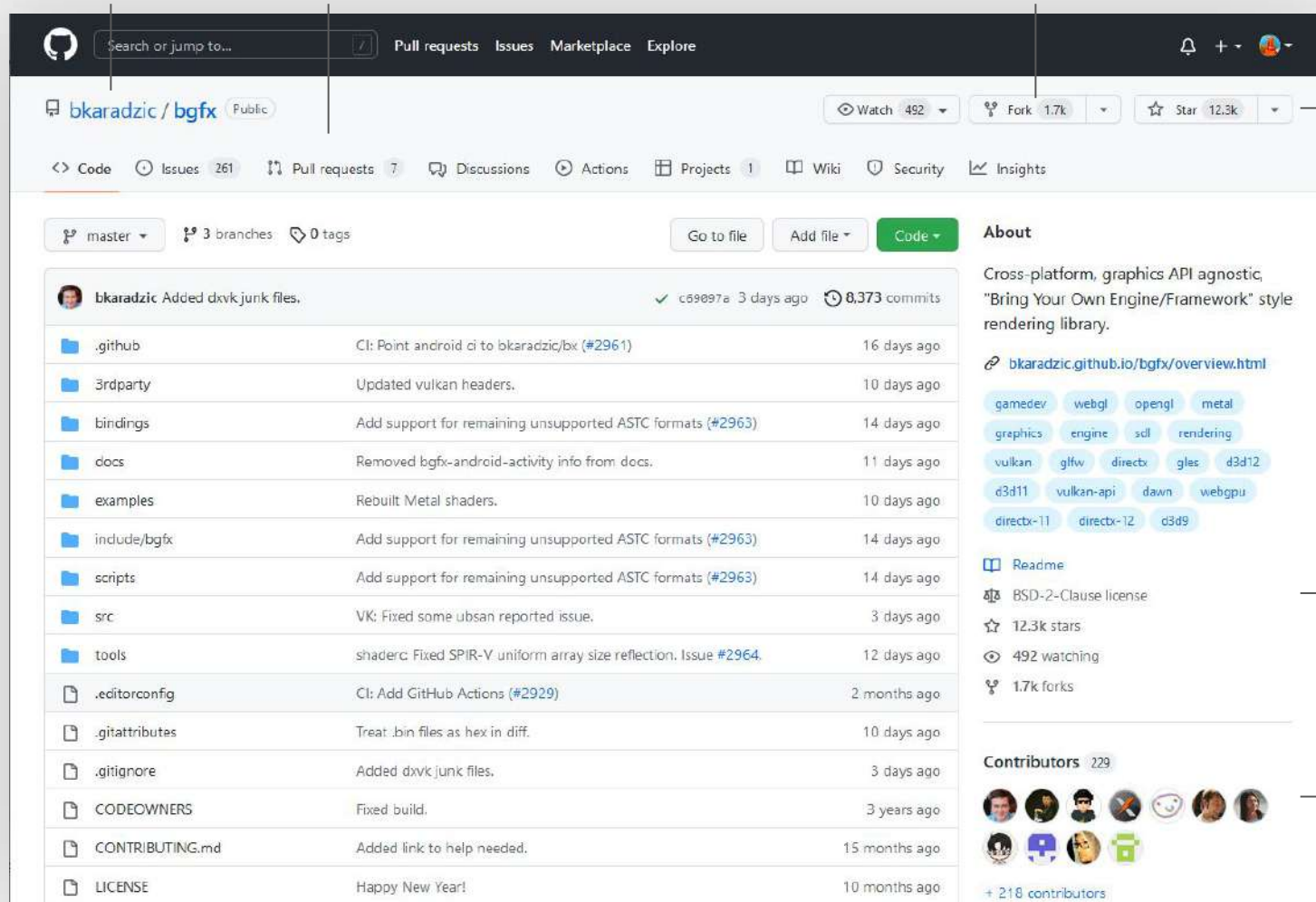


Le Version Control ?

- Suivi
 - QA / Debuggage
 - Expérimentation
 - Travail collaboratif
- branche(s)

The screenshot shows the GitHub repository page for **harfang3d / harfang3d**, which is a public repository. The page includes navigation tabs for Code, Issues (1), Pull requests, Discussions, Actions, and Projects. Below the tabs, there are buttons for 'main' branch, 'Go to file', 'Add file', and 'Code'. The repository description is 'HARFANG 3D source code public repository' with a link to www.harfang3d.com. A list of recent activity shows a merge pull request by astrofra 4 days ago and two folder updates (binding and doc) 11 months ago. A list of technologies used is displayed at the bottom: python, golang, opengl, lua, game-engine, cpp, physics, vr, vulkan, bgfx, directx, and realtime.





Search or jump to...

Pull requests Issues Marketplace Explore

bkaradzic / bgfx Public

Watch 492 Fork 1.7k Star 12.3k

Code Issues 261 Pull requests 7 Discussions Actions Projects 1 Wiki Security Insights

master 3 branches 0 tags

Go to file Add file Code

About

Cross-platform, graphics API agnostic, "Bring Your Own Engine/Framework" style rendering library.

bkaradzic.github.io/bgfx/overview.html

gamedev webgl opengl metal
graphics engine sdl rendering
vulkan glfw directx gles d3d12
d3d11 vulkan-api dawn webgpu
directx-11 directx-12 d3d9

Readme

BSD-2-Clause license

12.3k stars

492 watching

1.7k forks

Contributors 229

+ 218 contributors

Commits

Commit	Message	Time
bkaradzic	Added dxvk junk files.	3 days ago
c59097a	CI: Point android ci to bkaradzic/bx (#2961)	16 days ago
	Updated vulkan headers.	10 days ago
	Add support for remaining unsupported ASTC formats (#2963)	14 days ago
	Removed bgfx-android-activity info from docs.	11 days ago
	Rebuilt Metal shaders.	10 days ago
	Add support for remaining unsupported ASTC formats (#2963)	14 days ago
	Add support for remaining unsupported ASTC formats (#2963)	14 days ago
	VK: Fixed some ubsan reported issue.	3 days ago
	shaderc: Fixed SPIR-V uniform array size reflection. Issue #2964.	12 days ago
	CI: Add GitHub Actions (#2929)	2 months ago
	Treat .bin files as hex in diff.	10 days ago
	Added dxvk junk files.	3 days ago
	Fixed build.	3 years ago
	Added link to help needed.	15 months ago
	Happy New Year!	10 months ago



license

contributors

Retex sur nos pratiques de l'Open Source

BGFX

PyPI (HG Linux)

Blender 3D

Dear ImGui (et Go)

BabylonJS



Github Copilot, les nouveaux défis de l'IA

GitHub Copilot est un outil d'intelligence artificielle (...) pour aider les utilisateurs des environnements de (...) Visual Studio Code, Visual Studio, Neovim et JetBrains en complétant automatiquement le code.

Code original (GPL)

Code généré (non-GPL)

```
GitHub Copilot - javastuff - Visual Studio Code

View Go Run Terminal Help

... C: Untitled-1.cpp 5 • C: void cs_sparse (const cs *A, const double *values) • C: // sparse matrix transpose - untitled-2 •

1 // sparse matrix transpose
2 cs_

Terminal

File Edit View Search Terminal Help

#include "cs.h"
/* C = A' */
cs *cs_transpose (const cs *A, int values)
{
    csi p, q, j, *Cp, *Ci, n, m, *Ap, *Ai, *w ;
    double *Cx, *Ax ;
    cs *C ;
    if (!CS_CSC (A)) return (NULL) ; /* check inputs */
    m = A->m ; n = A->n ; Ap = A->p ; Ai = A->i ; Ax = A->x ;
    C = cs_spalloc (n, m, Ap [n], values && Ax, 0) ; /* allocate result */
    w = cs_calloc (m, sizeof (csi)) ; /* get workspace */
    if (!C || !w) return (cs_done (C, w, NULL, 0)) ; /* out of memory */
    Cp = C->p ; Ci = C->i ; Cx = C->x ;
    for (p = 0 ; p < Ap [n] ; p++) w [Ai [p]]++ ; /* row counts */
    cs_cumsum (Cp, w, m) ; /* row pointers */
    for (j = 0 ; j < n ; j++)
    {
        for (p = Ap [j] ; p < Ap [j+1] ; p++)
        {
            Ci [q = w [Ai [p]]++] = j ; /* place A(i,j) as entry C(j,i) */
            if (Cx) Cx [q] = Ax [p] ;
        }
    }
}
```

```
GitHub Copilot x

1 Synthesizing 10/10 solutions
2
3 =====
4
5 Accept Solution
6 cs_transpose (const cs *A, int values)
7 {
8     int p, q, j, *Cp, *Ci, *w, *Ti, *Tj ;
9     double *Cx, *Tx ;
10    cs *T ;
11    if (!CS_CSC (A) || !CS_TRIPLET (A)) return (NULL) ; /* check inputs
12    T = cs_spalloc (A->n, A->m, A->nzmax, values && A->x, 0) ;
13    w = cs_calloc (A->m, sizeof (int)) ; /* get workspace
14    if (!T || !w) return (cs_done (T, NULL, w, 0)) ; /* out of memory
15    Cp = T->p ; Ci = T->i ; Cx = T->x ; Ti = A->i ; Tj = A->j ;
16    for (p = 0 ; p < A->nzmax ; p++) w [Ti [p]]++ ; /* column counts
17    cs_cumsum (Cp, w, A->m) ; /* column pointers
18    for (j = 0 ; j < A->n ; j++)
19    {
20        for (p = A->p [j] ; p < A->p [j+1] ; p++)
21        {
22            Ci [q = w [Ti [p]]++] = j ; /* place A(i,j) as entry C(j,i)
23            if (Cx) Cx [q] = A->x [p] ;
24        }
25    }
26    return (cs_done (T, NULL, w, 1)) ; /* success; free w and return
27
28 // Language: cpp
29 // Compare this snippet from Untitled-1.cpp:
30 //
31 // cs *cs_transpose (const cs *A, int values)
32 // {
33 //     int p, q, j, *Cp, *Ci, *w, *Ti, *Tj ;
34 //     double *Cx, *Tx ;
35 //     cs *T ;
36 //     if (!CS_CSC (A) || !CS_TRIPLET (A)) return (NULL) ; /* check inputs
37 //     T = cs_spalloc (A->n, A->m, A->nzmax, values && A->x, 0) ;
```



```

1 // sparse matrix add
2 cs_

Terminal
File Edit View Search Terminal Help

#include "cs.h"
/* C = alpha*A + beta*B */
cs *cs_add (const cs *A, const cs *B, double alpha, double beta)
{
    csi p, j, nz = 0, anz, *Cp, *Ci, *Bp, m, n, bnz, *w, values;
    double *Cx, *Bx, *Cx;
    cs *C;
    if (!CS_CSC (A) || !CS_CSC (B)) return (NULL); /* check inputs */
    if (A->m != B->m || A->n != B->n) return (NULL);
    m = A->m; anz = A->p [A->n];
    n = B->n; Bp = B->p; Bx = B->x; bnz = Bp [n];
    w = cs_calloc (m, sizeof (csi)); /* get workspace */
    values = (A->x != NULL) && (Bx != NULL);
    x = values ? cs_malloc (m, sizeof (double)) : NULL; /* get workspace */
    C = cs_salloc (m, n, anz + bnz, values, 0); /* allocate result */
    if (!C || !w || (values && !x)) return (cs_done (C, w, x, 0));
    Cp = C->p; Ci = C->i; Cx = C->x;
    for (j = 0; j < n; j++)
    {
        Cp [j] = nz; /* column j of C starts here */
        nz = cs_scatter (A, j, alpha, w, x, j+1, C, nz); /* alpha*A(:,j) */
        nz = cs_scatter (B, j, beta, w, x, j+1, C, nz); /* beta*B(:,j) */
        if (values) for (p = Cp [j]; p < nz; p++) Cx [p] = x [Ci [p]];
    }
    Cp [n] = nz; /* finalize the last column of C */
    cs_sprealloc (C, 0); /* remove extra space from C */
    return (cs_done (C, w, x, 1)); /* success; free workspace, return C */
}

```

```

142
143 =====
144
Accept Solution
145 cs_add (const cs *A, const cs *B, double alpha, double beta)
146 {
147     int p, j, *Cp, *Ci, m, n, anz, *Bp, *Bi, bnz, *w, *x, nz nz0;
148     double *Cx, *Bx, *x;
149     cs *C;
150     if (!CS_CSC (A) || !CS_CSC (B)) return (NULL); /* check inputs
151     m = A->m; n = A->n; anz = A->p [n];
152     Bp = B->p; Bi = B->i; Bx = B->x; bnz = Bi [n];
153     if (m != B->m || n != B->n) return (NULL); /* check dimensions
154     C = cs_salloc (m, n, anz + bnz, A->x != NULL, 0); /* alloc result
155     w = cs_calloc (n, sizeof (int)); /* get workspace
156     x = (A->x != NULL) ? cs_malloc (n, sizeof (double)) : NULL; /* get workspace
157     if (!C || !w || !x) return (cs_done (C, w, x, 0)); /* out of memory
158     Cp = C->p; Ci = C->i; Cx = C->x;
159     for (j = 0; j < n; j++)
160     {
161         Cp [j] = Cx; /* column j of C starts here
162         nz = cs_scatter (A, j, alpha, w, x, j+1, C, nz); /* alpha*A(:,j)
163         nz = cs_scatter (B, j, beta, w, x, j+1, C, nz); /* beta*B(:,j)
164     }
165     Cp [nn = nz]; /* finalize the last column
166     cs_sprealloc (C, 0); /* remove extra space from C
167     return (cs_done (C, w, x, 1)); /* success; free workspace,
168 }
169
170 // sparse matrix axpy
171 cs_axpy (const cs *A, double alpha, const cs *B, double beta)
172 {
173     int
174
175 =====
176
Accept Solution
177 cs_axpy (const cs *, const cs *B, double alpha, double beta)
178 {
179     int p, j, *Cp, *Ci, m, n, anz, *Bp, bnz, *w;
180     double *Cx, *B;
181     cs *C;
182     if (!CS_CSC (A) || !CS_CSC (B)) return (NULL); /* check inputs
183     m = A->m; anz = A->p [A->n];

```

File View Run Terminal Help

```

1 // sparse matrix elimination tree
2 CS_

File Edit View Search Terminal Help
#include "cs.h"
/* compute the etree of A (using triu(A), or A'A without forming A'A */
csi *cs_etree (const cs *A, csi ata)
{
    csi i, k, p, m, n, inext, *Ap, *Ai, *w, *parent, *ancestor, *prev ;
    if (!CS_CSC (A)) return (NULL) ; /* check inputs */
    m = A->m ; n = A->n ; Ap = A->p ; Ai = A->i ;
    parent = cs_malloc (n, sizeof (csi)) ; /* allocate result */
    w = cs_malloc (n + (ata ? m : 0), sizeof (csi)) ; /* get workspace */
    if (!w || !parent) return (cs_idone (parent, NULL, w, 0)) ;
    ancestor = w ; prev = w + n ;
    if (ata) for (i = 0 ; i < m ; i++) prev [i] = -1 ;
    for (k = 0 ; k < n ; k++)
    {
        parent [k] = -1 ; /* node k has no parent yet */
        ancestor [k] = -1 ; /* nor does k have an ancestor */
        for (p = Ap [k] ; p < Ap [k+1] ; p++)
        {
            i = ata ? (prev [Ai [p]]) : (Ai [p]) ;
            for ( ; i != -1 && i < k ; i = inext) /* traverse from i to k */
            {
                inext = ancestor [i] ; /* inext = ancestor of i */
                ancestor [i] = k ; /* path compression */
                if (inext == -1) parent [i] = k ; /* no anc., parent is k */
            }
            if (ata) prev [Ai [p]] = k ;
        }
    }
    return (cs_idone (parent, NULL, w, 1)) ;
}

```

```

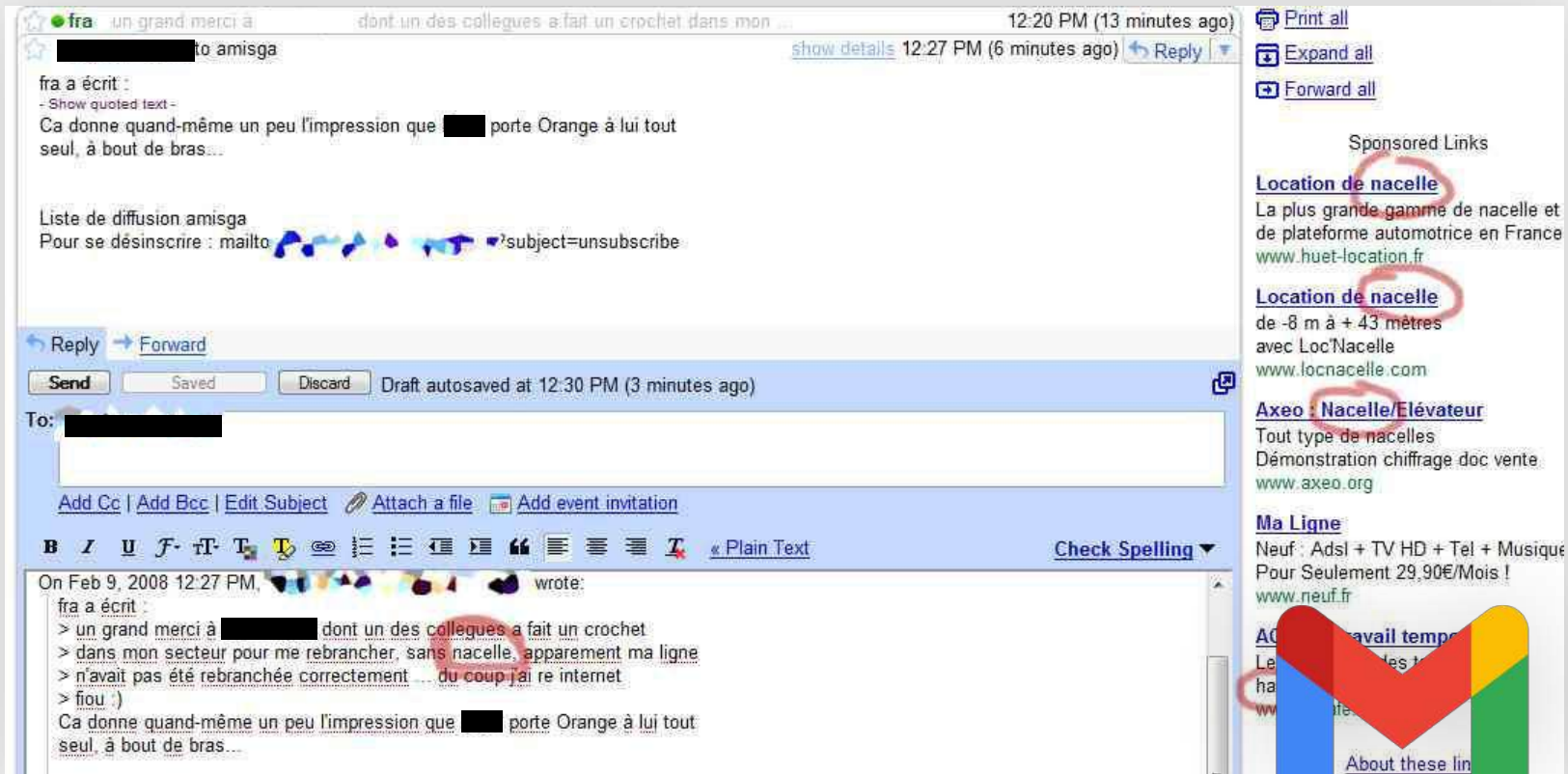
37 for ( ; i != -1 && i < k ; i = inext)
38 {
39     if (w [i] < k
40
=====
41
42
Accept Solution
43 cs_etree (const cs *A, int ata)
44 {
45     int i, k, p, m, n, inext, *Ap, *Ai, *w, *parent, *ancestor, *prev ;
46     if (!CS_CSC (A)) return (NULL) ; /* check inputs
47     m = A->m ; n = A->n ; Ap = A->p ; Ai = A->i ;
48     parent = cs_malloc (n, sizeof (int)) ; /* allocate result
49     w = cs_malloc (n+1, sizeof (int)) ; /* get workspace
50     if (!w || !parent) return (cs_idone (parent, NULL, w, 5)) ;
51     ancestor = w ; prev = w + n ;
52     for (k = 0 ; k < n ; k++) parent [k] = -1 ; /* no nodes yet
53     for (k = 0 ; k < n ; k++) ancestor [k] = -1 ;
54     for (k = 0 ; k < n ; k++) prev [k] = -1 ;
55     for (k = 0 ; k < n ; k++) /* find etree of A'A
56     {
57         if (!ata) for (p = Ap [k] ; p < Ap [k+1] ; p++) // A(:,k) pattern
58         {
59             i = Ai [p] ;
60             for ( ; ancestor [i] != k ; i = inext) // traverse from i to k
61             {
62                 parent [i] = k ; /* path compression
63                 inext = ancestor [i] ; /* inext = parent of i
64                 ancestor [i] = k ;
65                 if (inext == -1) break ; /* end of path
66             }
67         }
68         else for (p = Ap [k] ; p < Ap [k+1] ; p++) // A(:,k) pattern
69         {
70             i = Ai [p] ;
71             for ( ; ancestor [i] != k ; i = inext) // traverse from i to k
72             {
73                 parent [i] = k ; /* path compression
74                 inext = ancestor [i] ; /* inext = parent
75             }
76         }
77     }
78
Accept Solution
79 cs_etree (const cs *A, int ata)
80 {
81     int i, k, p, m, n, inext, *Ap, *Ai, *w, *parent, *ancestor, *prev ;

```


Copilot peut-il produire du code de qualité ?



confiez-vous



Questions :)

Contacts

Thomas Simonnet

Lead Dev. Movida Prod.

`thomas.simonnet@movida-prod.com`

François Guthertz

CTO Harfang 3D

`francois.guthertz@movida-prod.com`