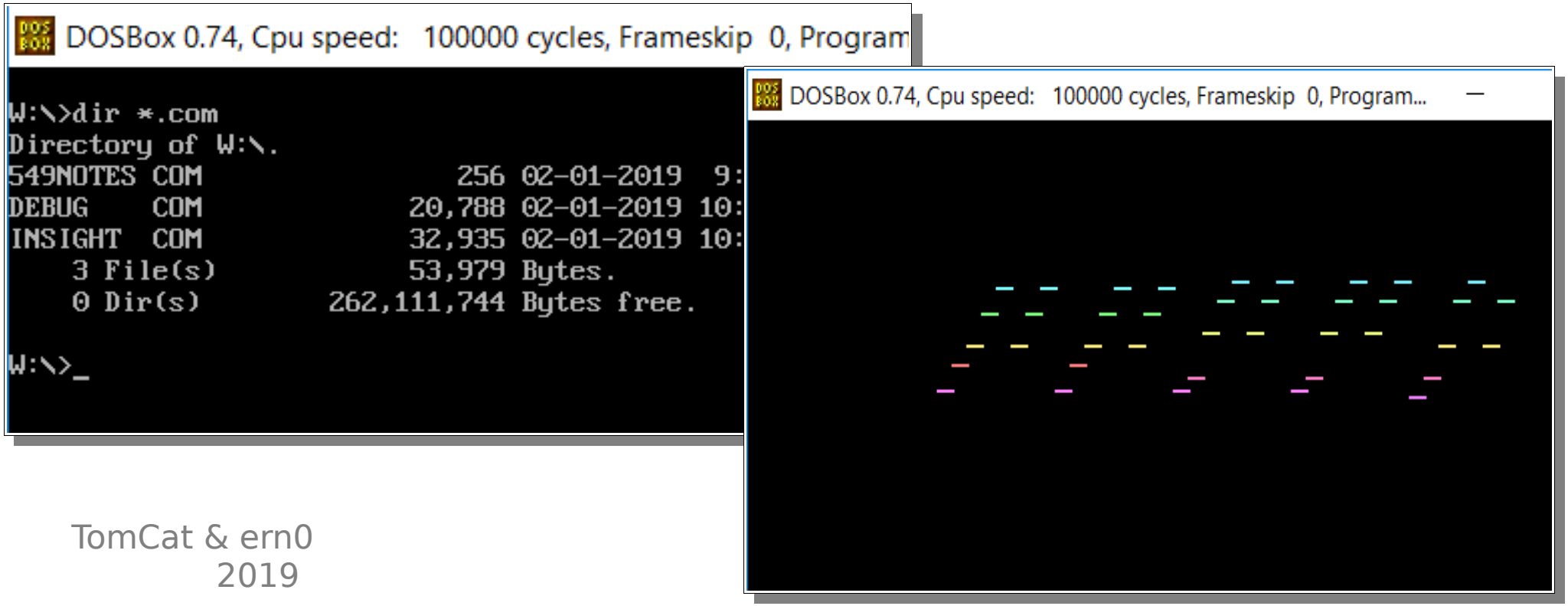
Making of

549NOTES.COM

the 256-byte intro PC-DOS intro which plays 549 notes

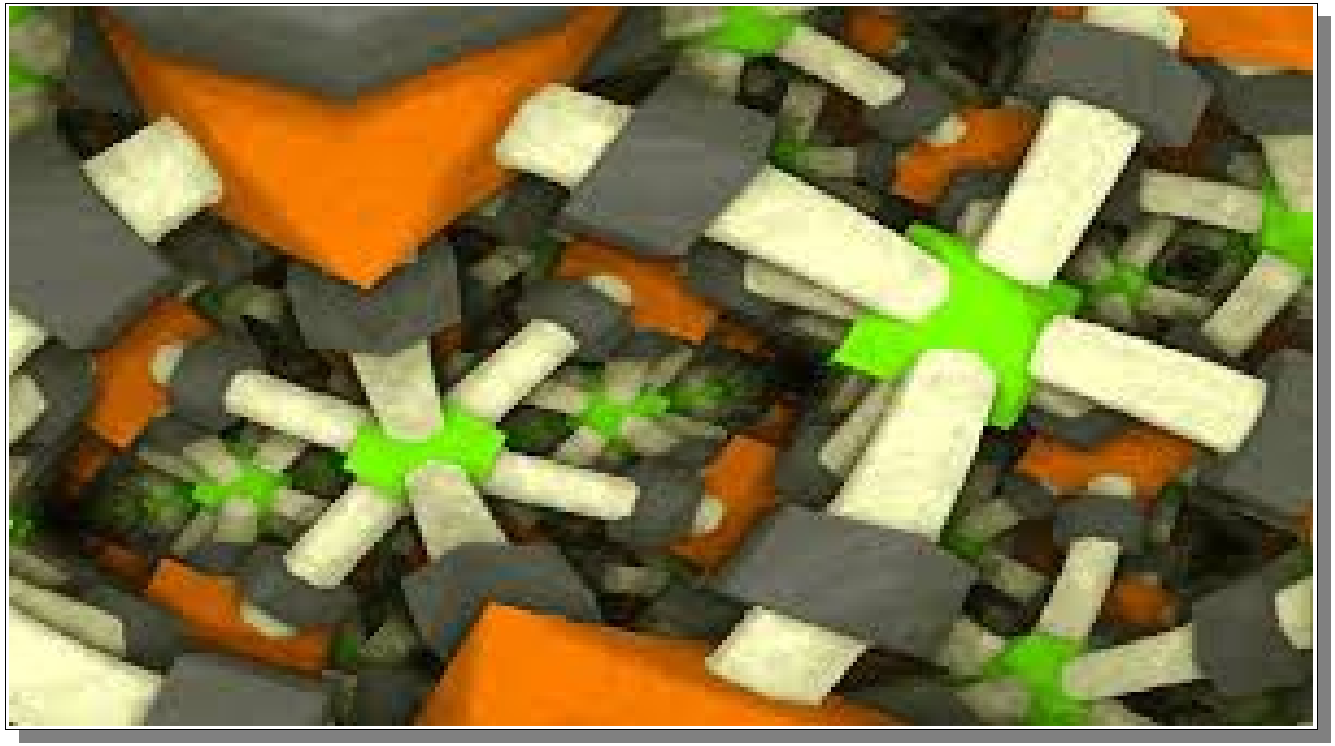


TomCat & ern0

2019

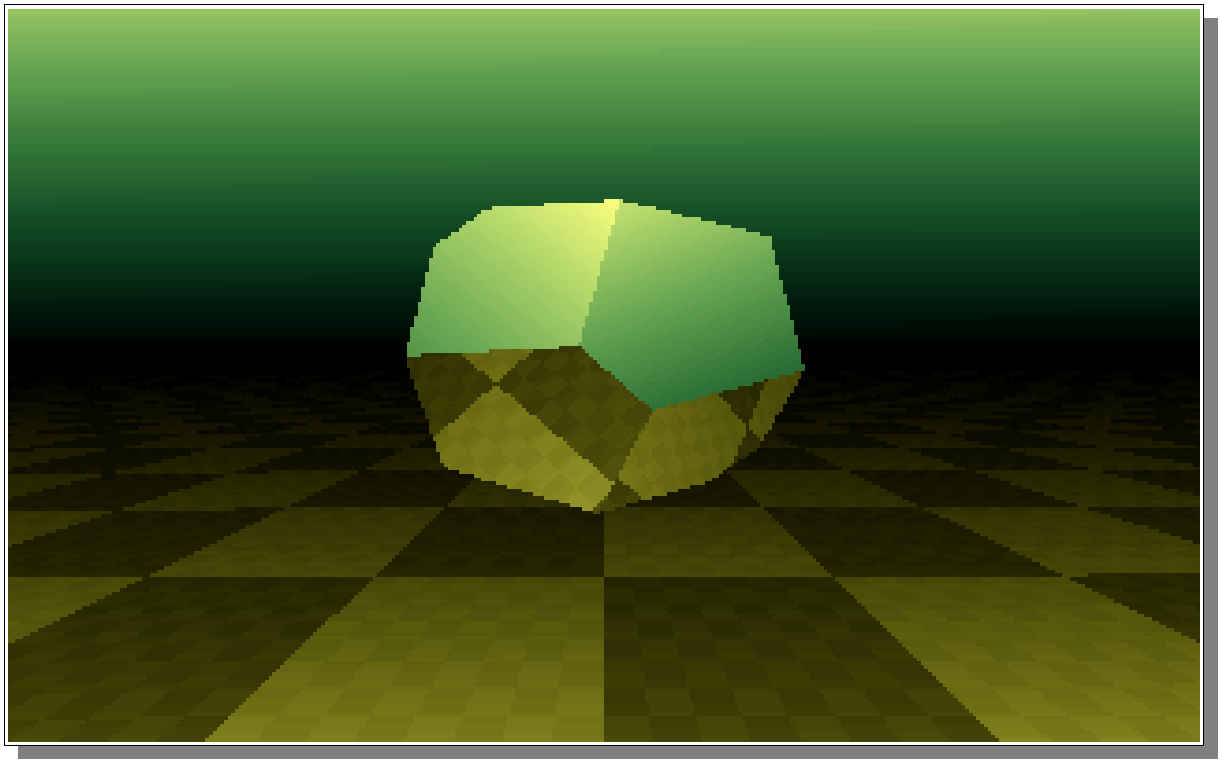
256-byte demoscene: extremly strong competition

256-byte demoscene: extremly strong competition



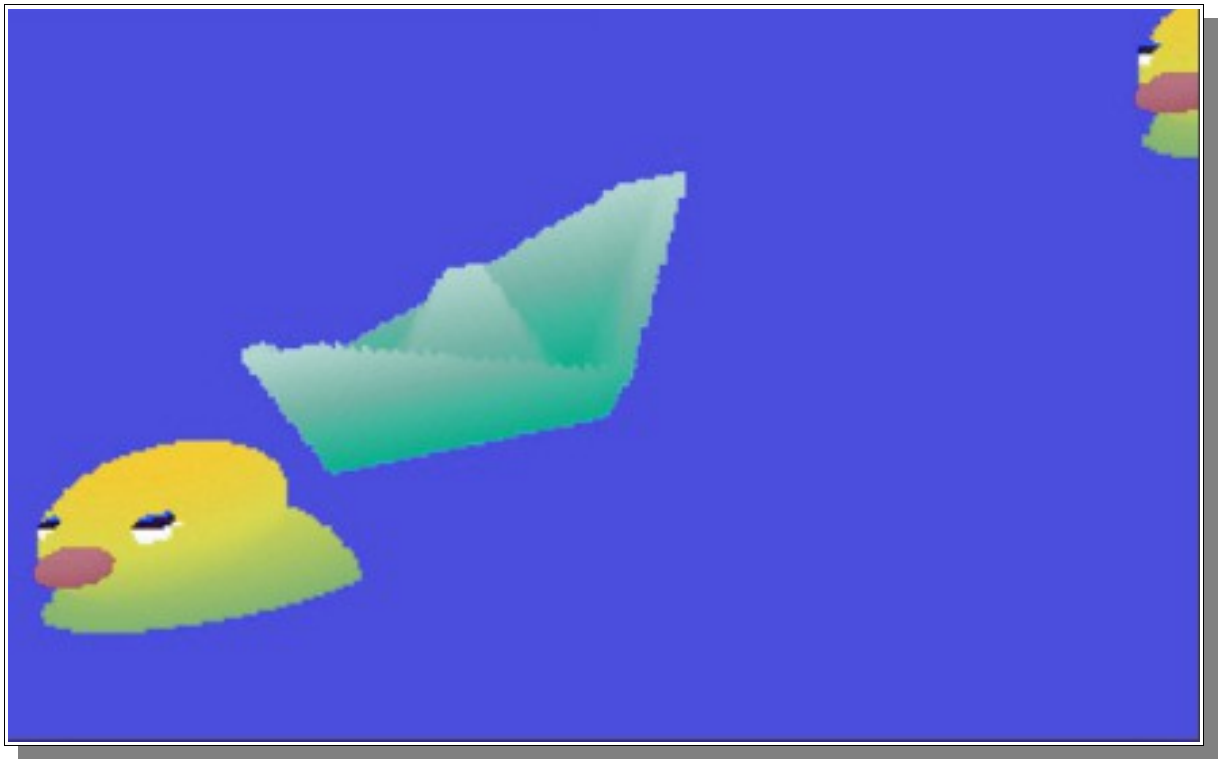
Řrřola: Puls

256-byte demoscene: extremly strong competition



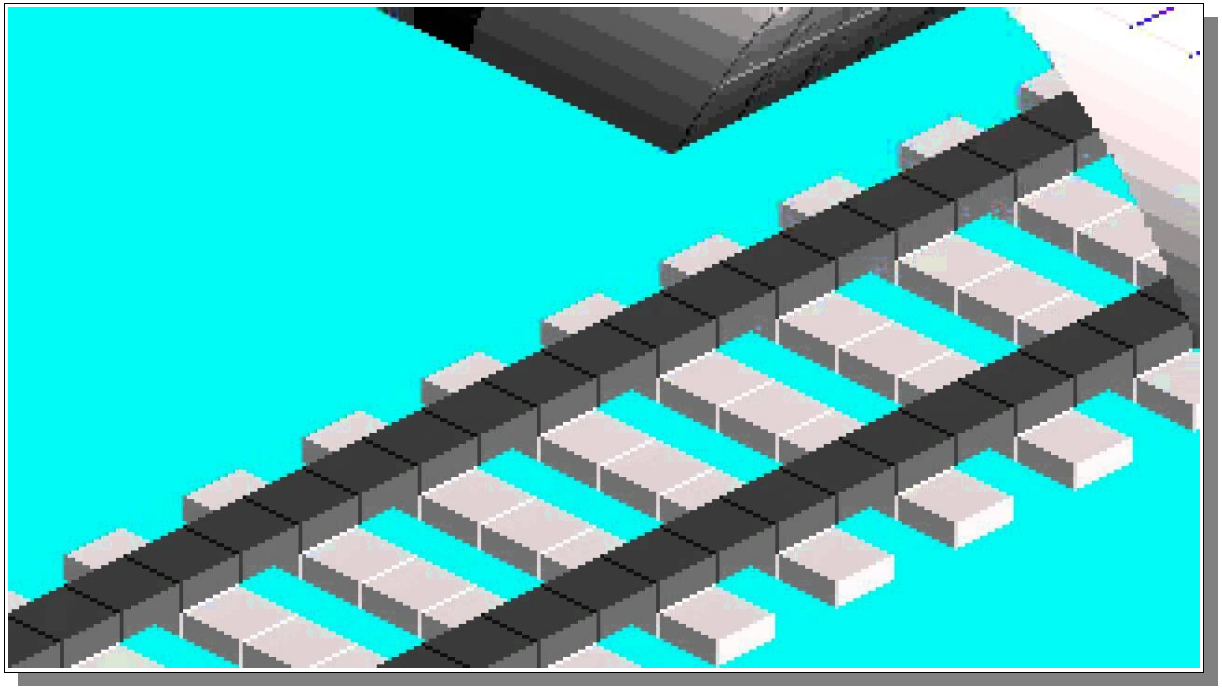
Řrřola: Pyrit

256-byte demoscene: extremly strong competition



Digimind: Pool Patrol

256-byte demoscene: extremly strong competition



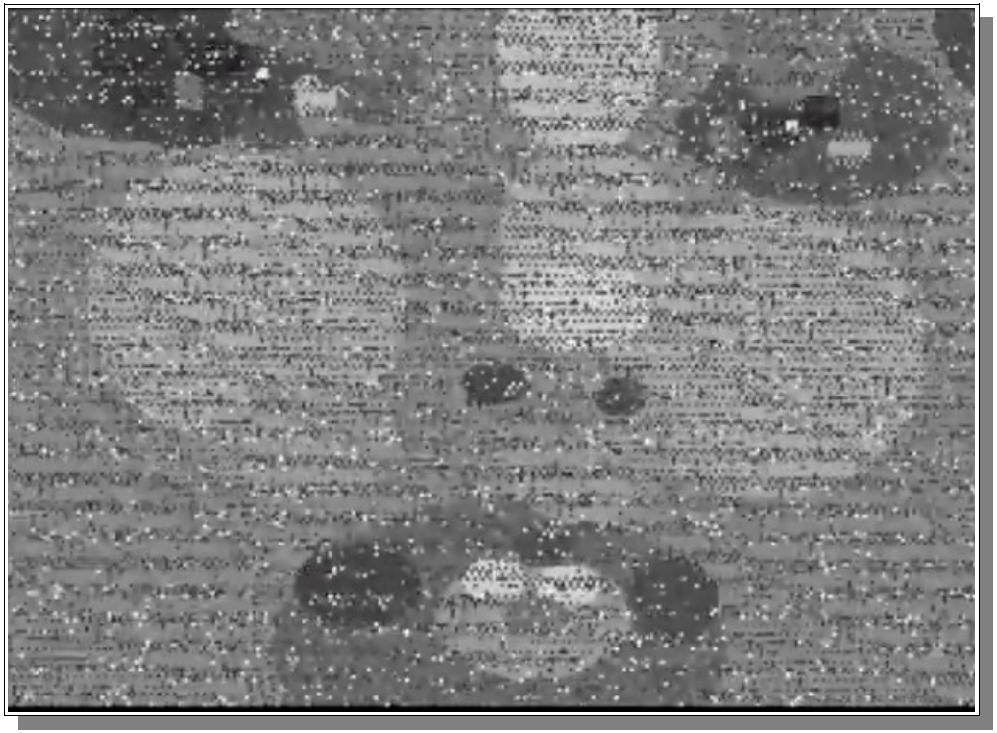
Digimind: Immediate Railways

256-byte demoscene: extremly strong competition

How to shine out of crowd?

256-byte demoscene: how to beat competition?

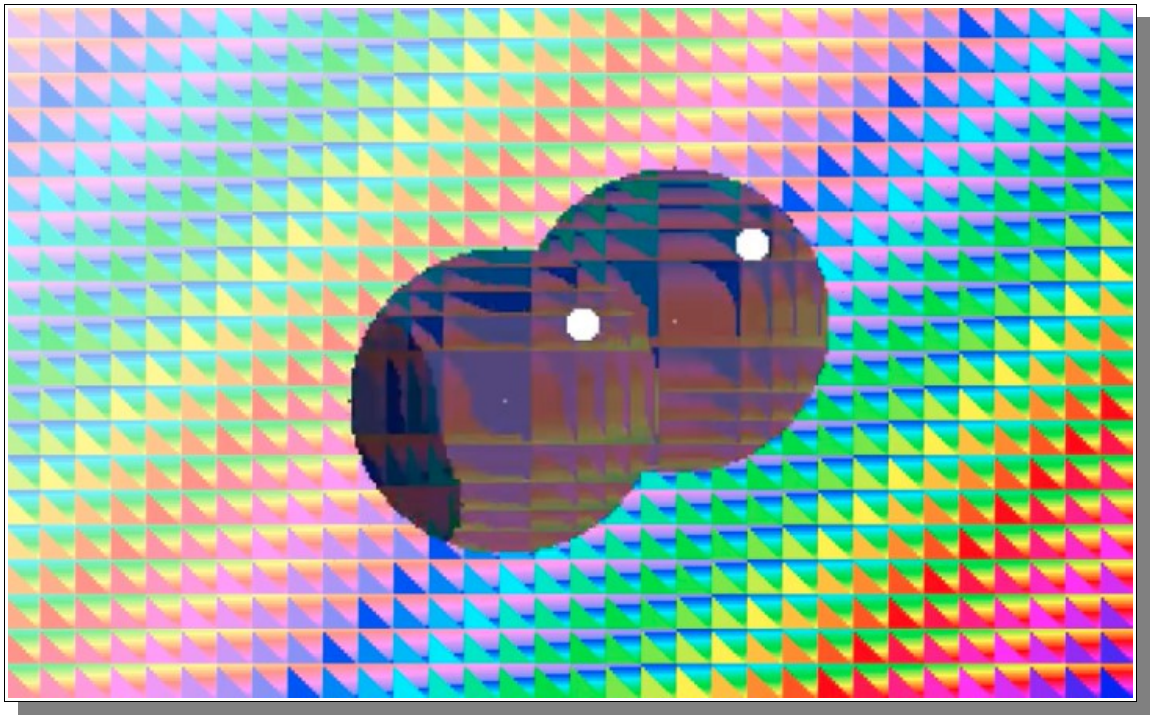
Image processing



TomCat: She – Weak Signal

256-byte demoscene: how to beat competition?

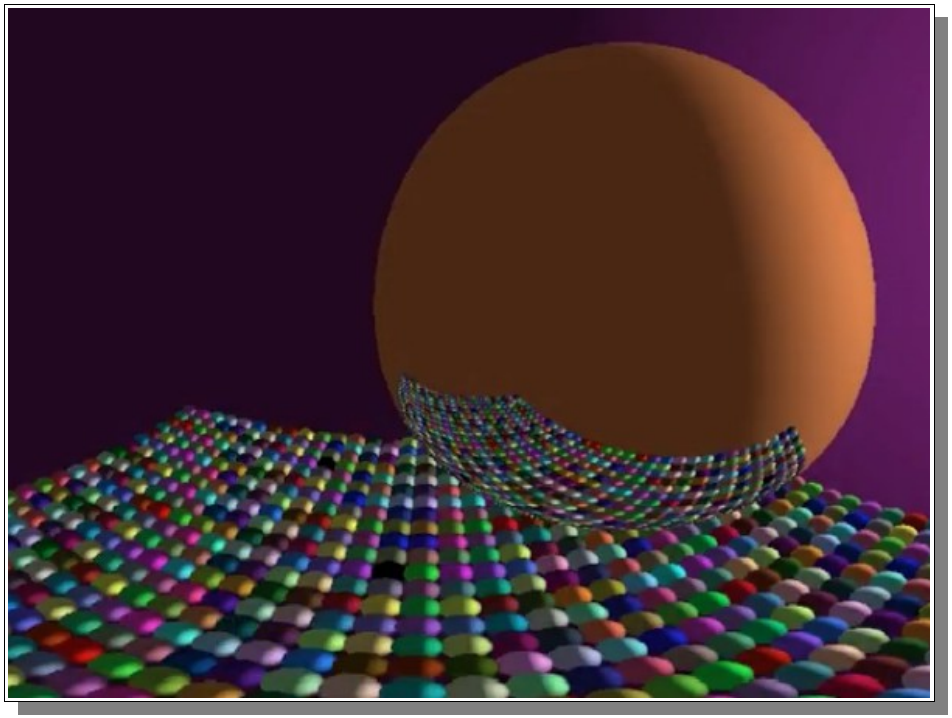
Raytracing



TomCat: Spectrum Rulez!

256-byte demoscene: how to beat competition?

Raytracing

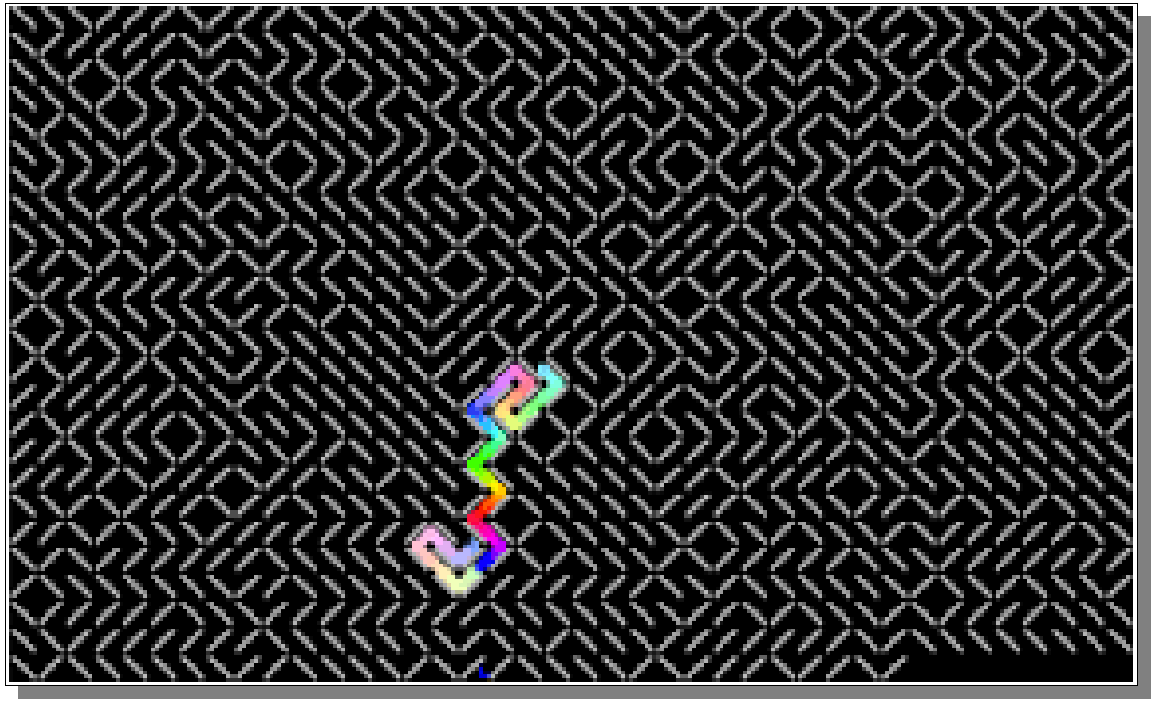


TomCat: Colorful

256-byte demoscene: how to beat competition?

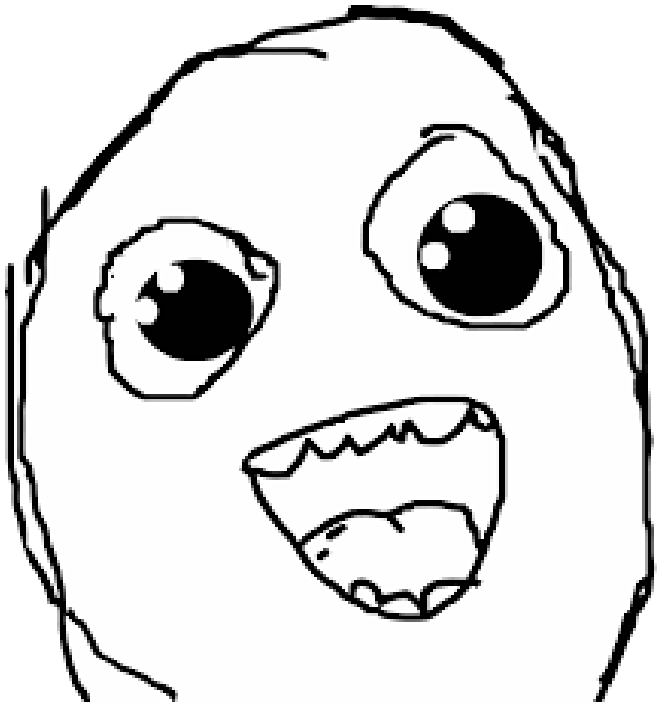
Fun

(if you are not a hardcore sizecoder)



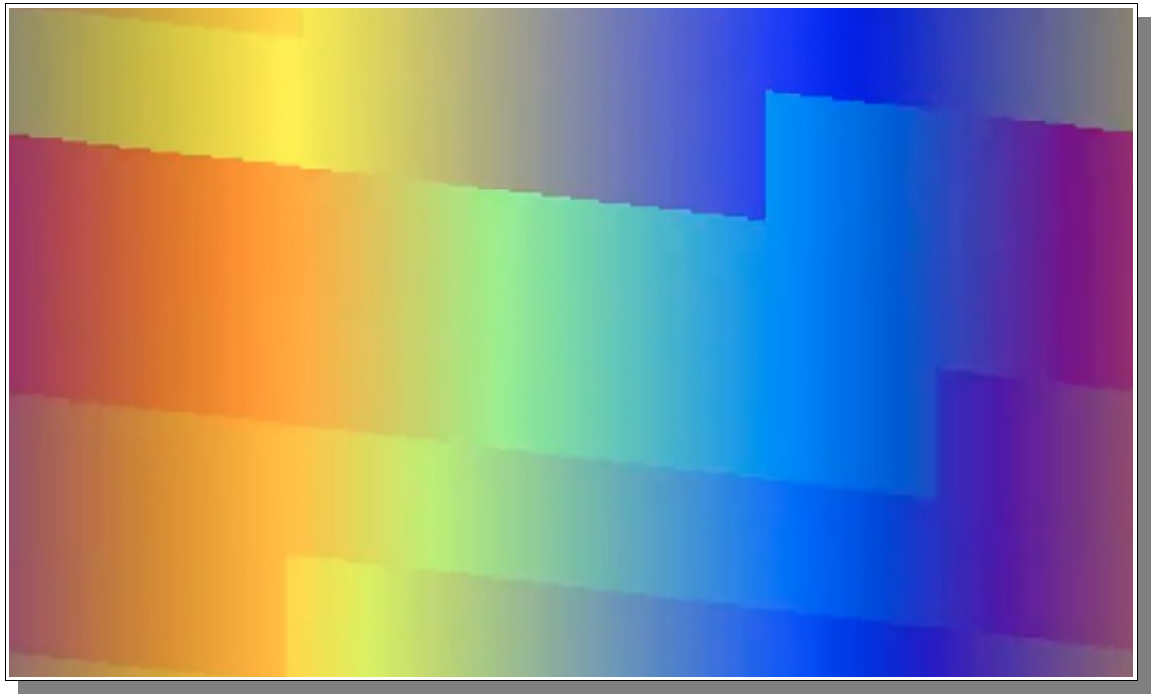
ern0: Maze Solver

256-byte demoscene: how to beat competition?



Music! Add music in 256-byte intros!

256 byte intro with music



TomCat: 2(56)unlimited

(bytebeat music by ern0)

256 byte intro with music

TomCat: No Sleep!

(buzzer music by ern0)

Create universal bytebeat tool

- Bytebeat player & editor
  TomCat

Create universal bytebeat tool

- Bytebeat player & editor
  TomCat

- Formula pre-compiler for assembly
  ern0

Create universal bytebeat tool

- Bytebeat player & editor
  TomCat

- Formula pre-compiler for assembly
  ern0

Create universal bytebeat tool

- Bytebeat player & editor
  TomCat

- Formula pre-compiler for assembly
  ern0

Bytebeat Editor (TomCat)

...

Assemblyzator (ern0)

Transform bytebeat formula to assembly code...

Assemblyzator (ern0)

Transform bytebeat formula to assembly code
...using a modern C compiler!

Assemblyzator (ern0)

Transform bytebeat formula to assembly code using a modern C compiler!

```
int main() {

    int result = 0;

    for (int i = 0; i < 100; i++) {
        for (int j = 0; j < 100; j++) {
            result += i * j;
        }
    }

    return result;
}
```

```
b8 e4 e0 75 01
5c3
```

```
main:
 mov    eax,0x175e0e4
 ret
```

Assemblyzator (ern0)

Transform bytebeat formula to assembly code using a modern C compiler!

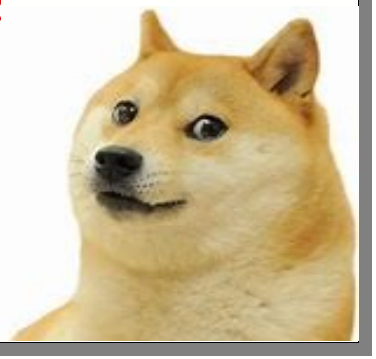
```c
int main() {

    int result = 0;

    for (int i = 0; i < 100; i++) {
        for (int j = 0; j < 100; j++) {
            result += i * j;
        }
    }

    return result;
}
```

```
b8 e4 e0 75 01
5c3
```

```
main:
 mov     eax,0x175e0e4
 ret
```

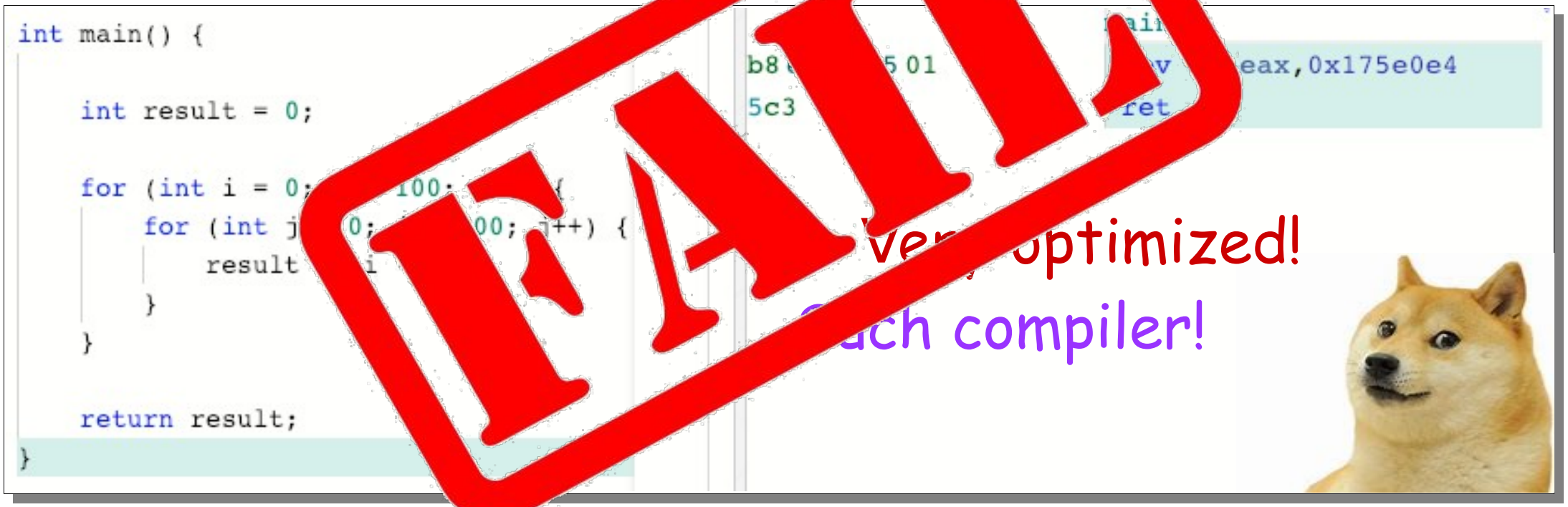
Very optimized!

Such compiler!



Assemblyzator (ern0)

Transform bytebeat formula to assembly code using a modern C compiler!



No Such **16-bit** Compiler Exists. Period.

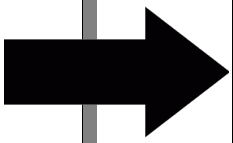
Assemblyzator (ern0)

Split bytebeat formula to simple instructions which are close to assembly:

Assemblyzator

Assemblyzator (ern0)

```
((t<<1)^((t<<1)+
(t>>7)&t>>12))|
t>>(4-(1^7&(t>>19)))
|t>>7
```



```
var3 = t << 1
var7 = t >> 7
var5 = var3 + var7
var6 = t >> 12
var4 = var5 & var6
var1 = var3 ^ var4
var12 = t >> 19
var11 = 7 & var12
var10 = 1 ^ var11
var9 = -var10
var9 = var9 + 4
var8 = t >> var9
var2 = var8 | var7
result = var1 | var2
```

Assemblyzator (ern0)

Pros:

- splits formula
- handles num arrays
- handles string arrays
- removes duplications

Cons:

- 3-op (A = B x C)
- fails on (A x B ? C : D)

Assemblyzator (ern0)

Pros:

- splits formula
- handles num arrays
- handles string arrays
- removes duplications

Cons:

- 3-op (A = B x C)
- fails on (A x B ? C : D)

Summary:

- nice try, but does not help much
- writing a compiler is not as easy as it looks first

Assemblyzator (ern0)

Pros:

- splits formula
- handles num arrays
- handles string arrays
- removes duplications

Cons:

- 3-op (A = B x C)
- fails on (A x B ? C : D)

Summary:

- nice try, but does not help much
- writing a compiler is not as easy as it looks first