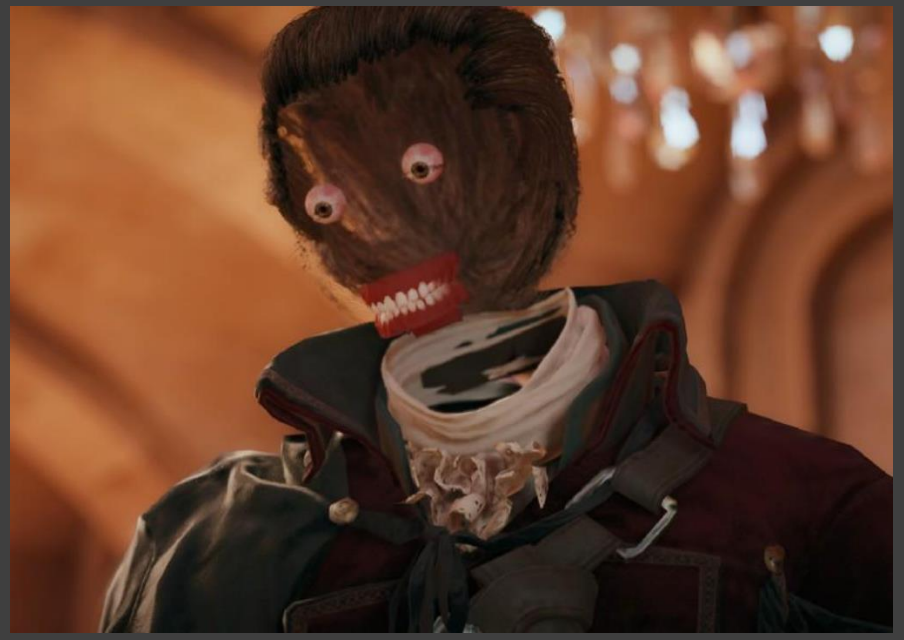
La evolución de los errores de programación en los videojuegos: El curioso caso de Minecraft

Adrián Muelas Gómez

Introducción

- Por qué son interesantes los errores de programación
- Por qué vamos a hablar de Minecraft
- Entender el caso de Minecraft
- Analizar por qué es inestable e intentar evitarlo

Glitch



Assassin's Creed Unity (2014)
Missing geometry

Super Mario 64 (1996)
Out of Bounds

Exploit

Super Mario Bros. (1985)

Infinite 1-UP

Exploit de diseño

Super Mario World (1990)

Credits Warp

Ejecución de Código Arbitrario (ACE)

¿Qué es Minecraft?

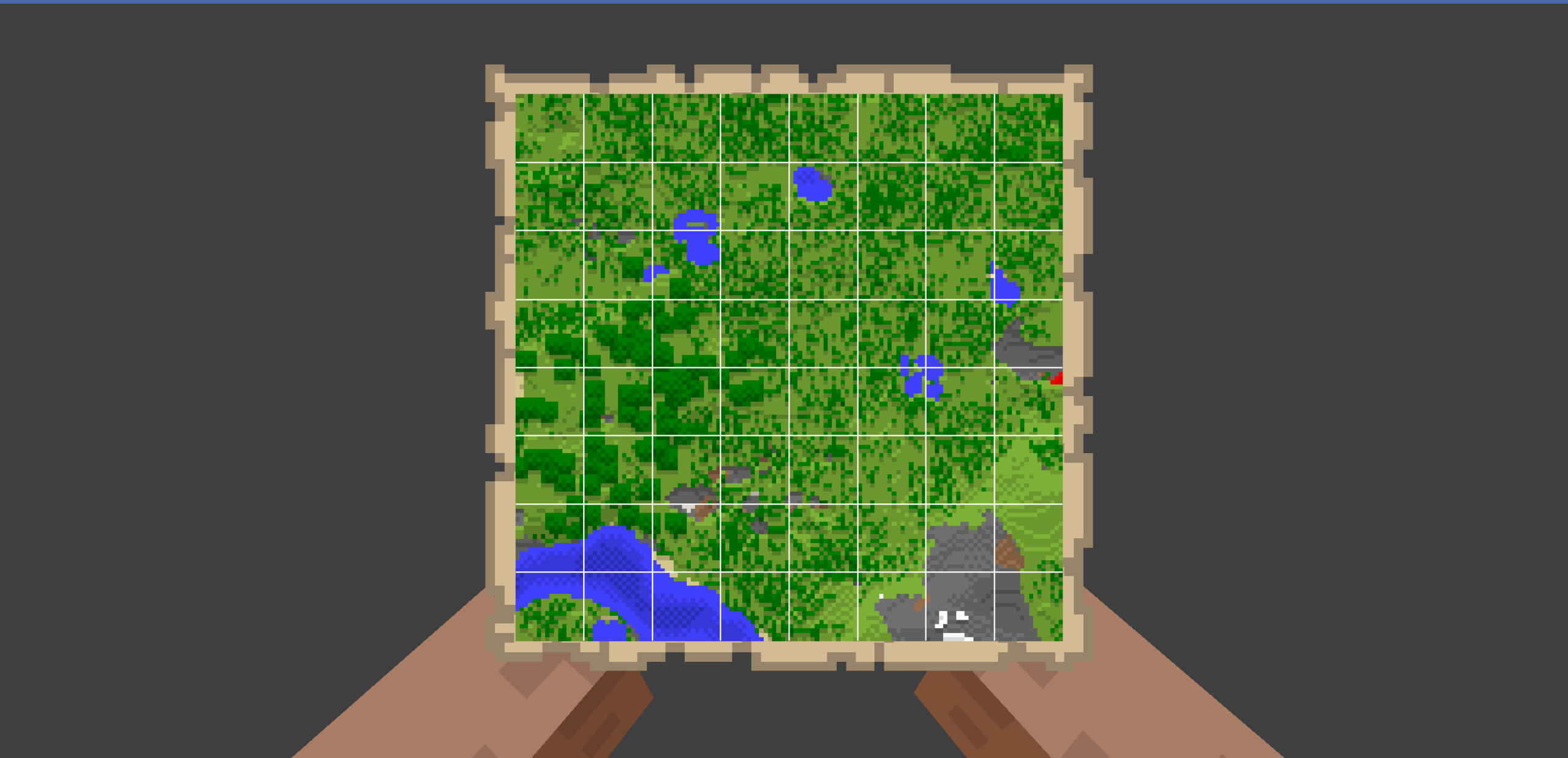
Arte promocional (2021)

Situación actual

Entendiendo el exploit

- Chunks y generación
- Beacon y cristal tintado
- Chunk Swap
- Falling Block Swap
- Word Tearing
- Generic Method

Chunks



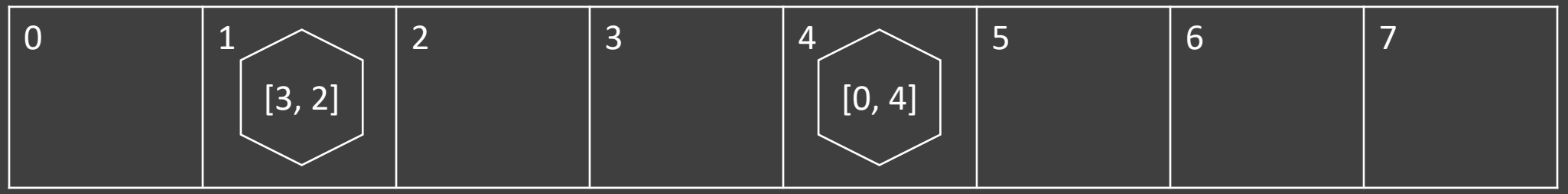
Chunks: Generación

Chunks: Generación

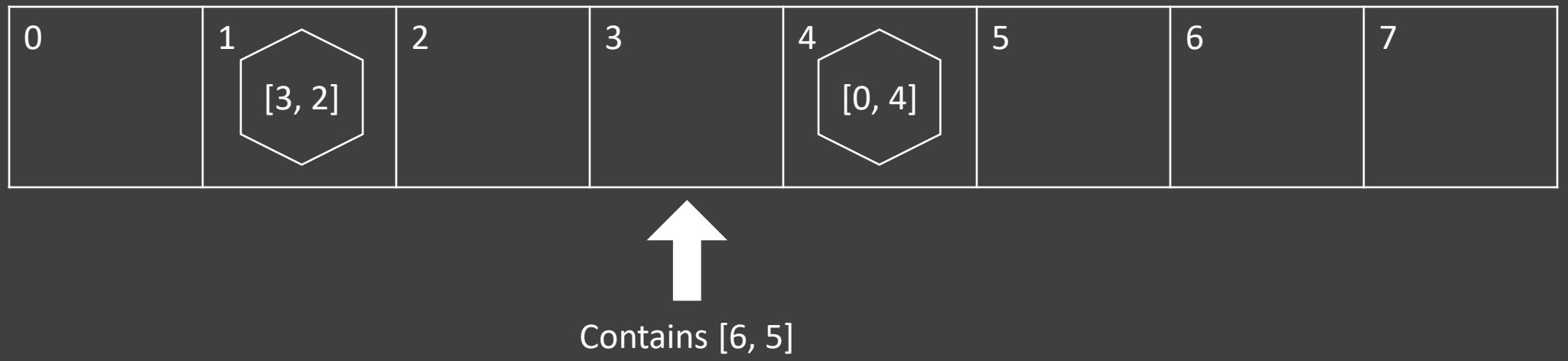
Beacon y cristal tintado

```
updateColorAsync(world, position):
  Thread.run:
    while position.y >= 0:
      block = world.getBlock(position)
      if block is BEACON:
        world.addSynchronizedTask(lambda:
          block.updateColor(position))
      position = position.down()
```

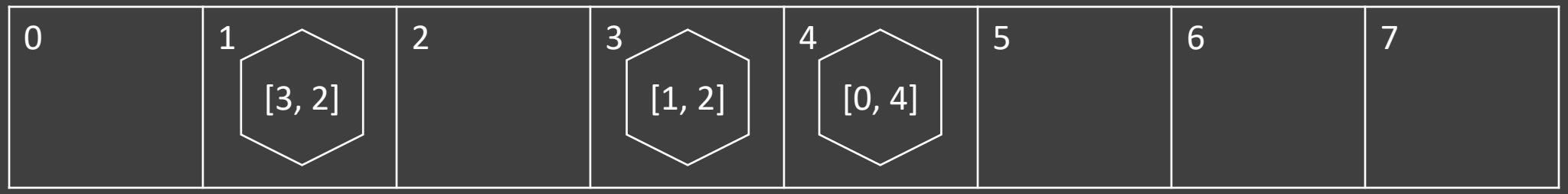
Chunk Swap: Tabla hash



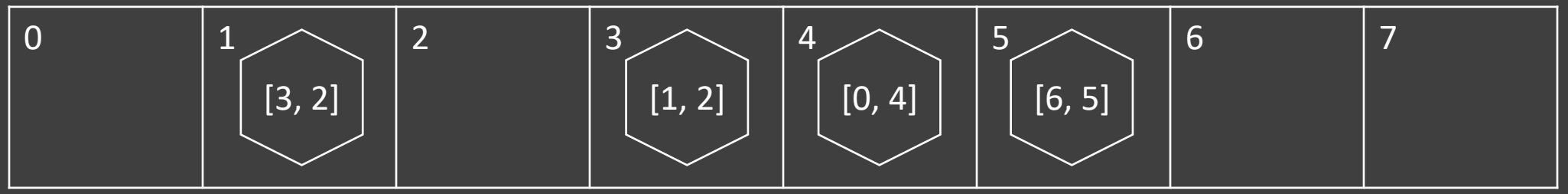
Chunk Swap: Tabla hash



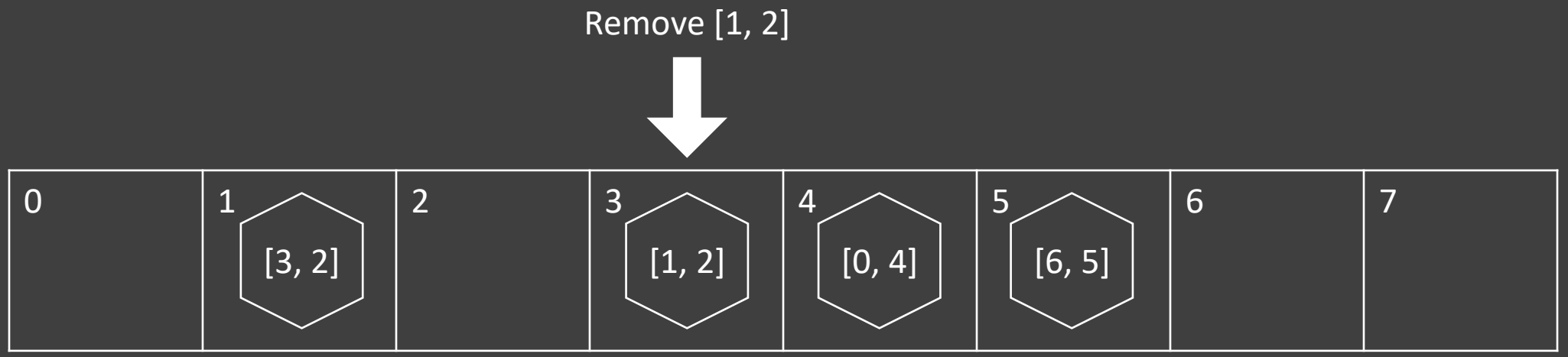
Chunk Swap: Tabla hash



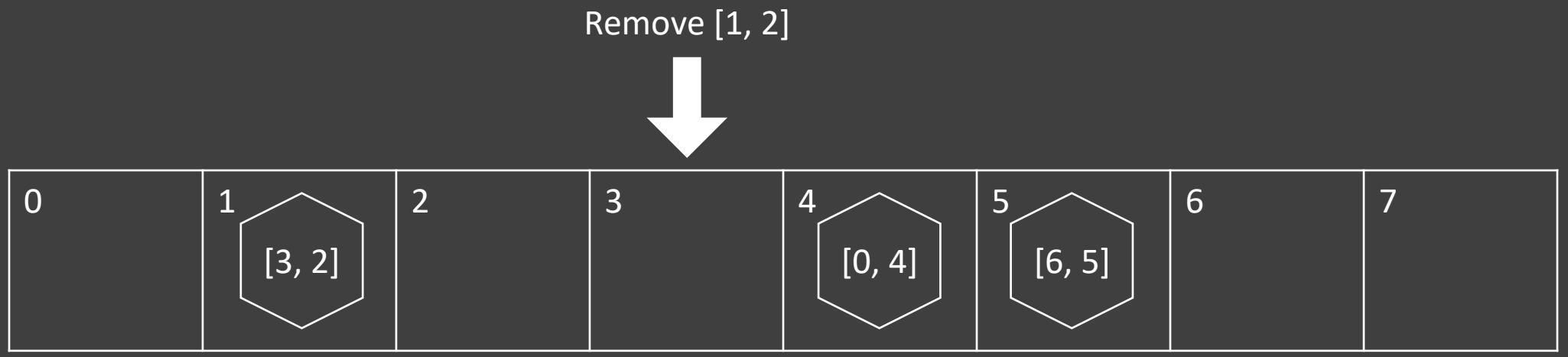
Chunk Swap: Tabla hash



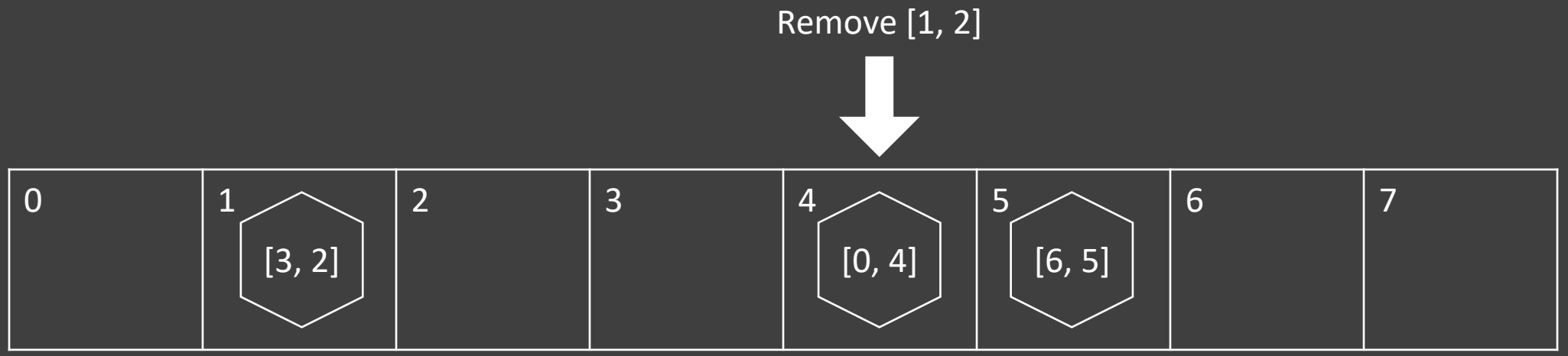
Chunk Swap: Tabla hash



Chunk Swap: Tabla hash

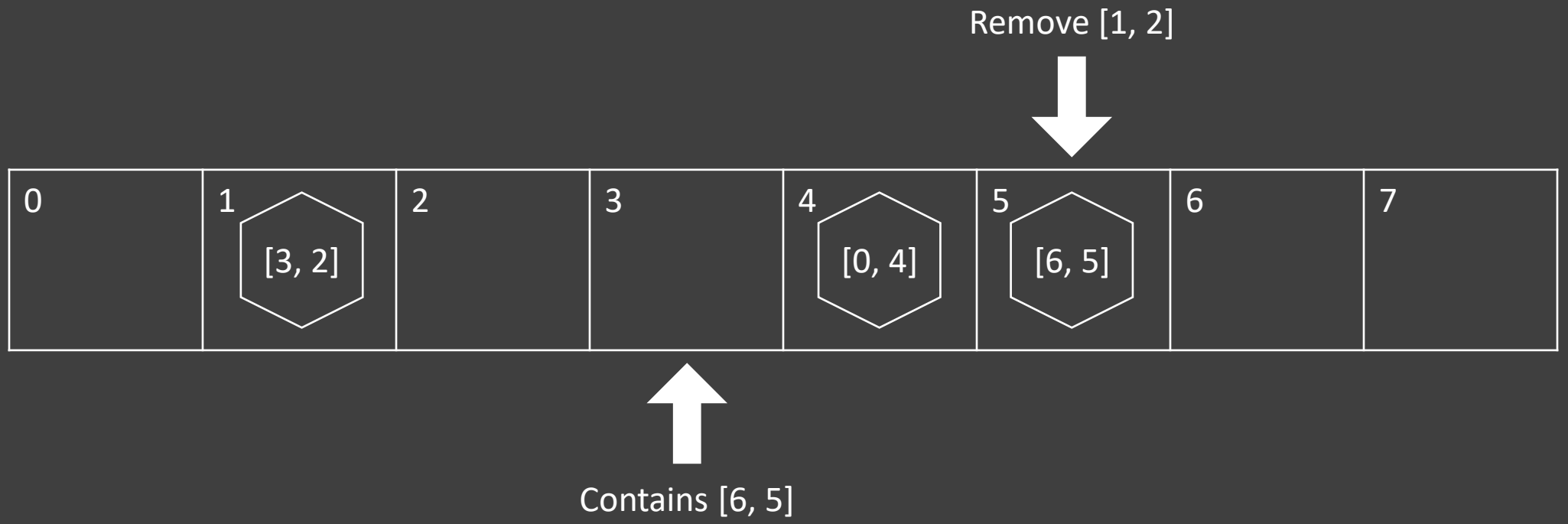


Chunk Swap: Tabla hash



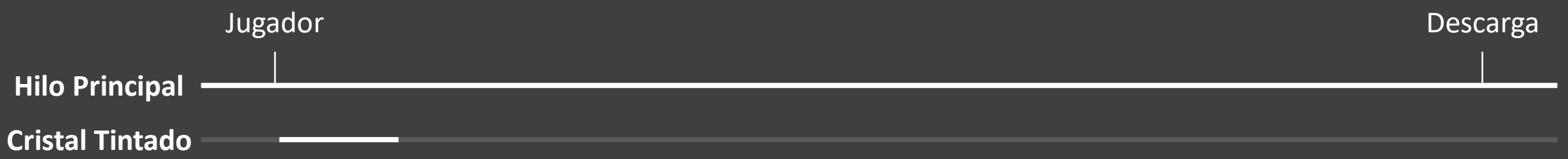
Chunk Swap: Tabla hash

Chunk Swap: Tabla hash



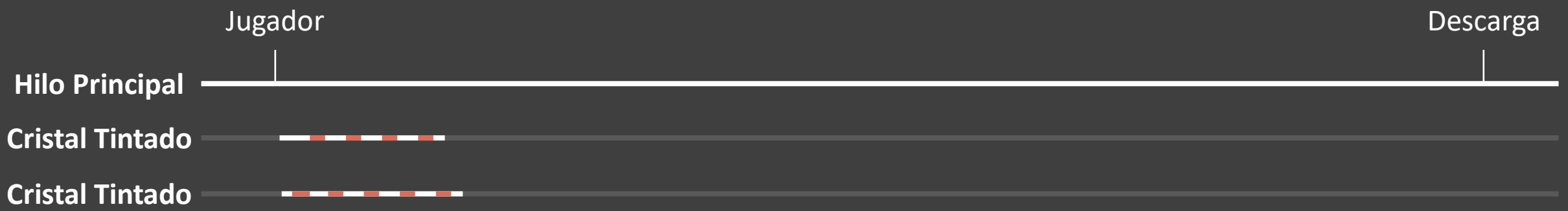
Chunk Swap: Sincronización

Chunk Swap: Sincronización

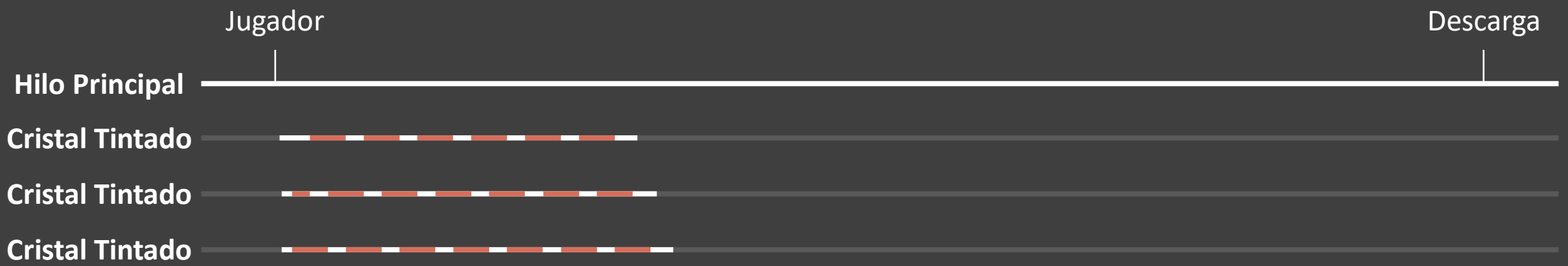



```
updateColorAsync(world, position):
  Thread.run:
    while position.y >= 0:
      block = world.getBlock(position)
      if block is BEACON:
        world.addSynchronizedTask(lambda:
          block.updateColor(position))
      position = position.down()
```

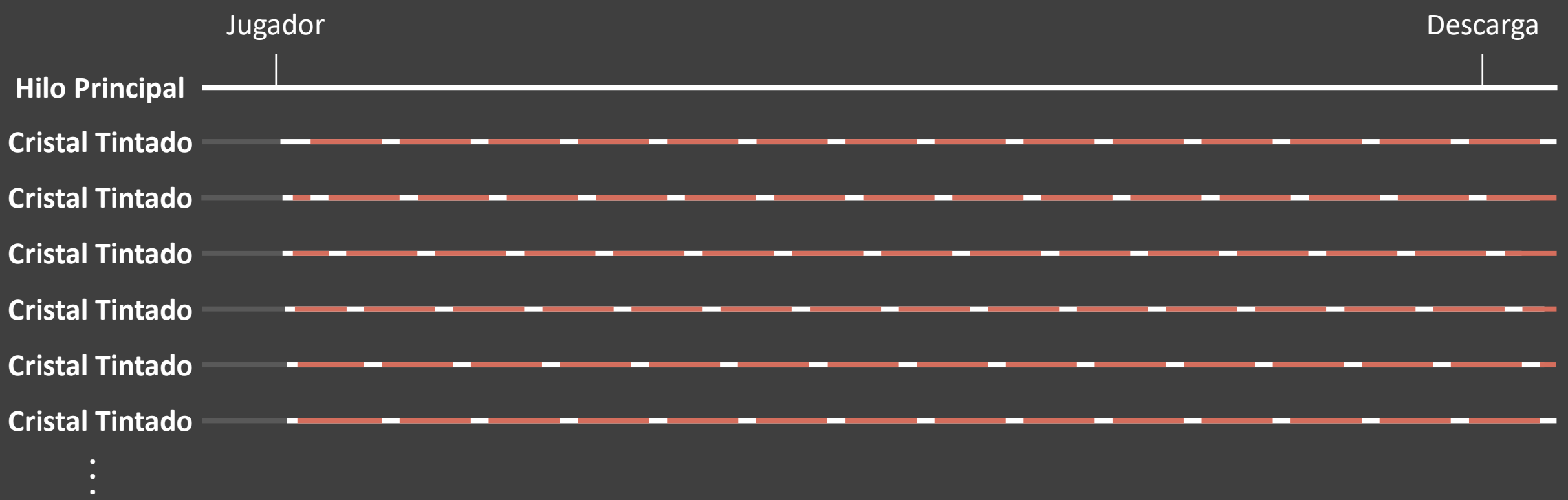
Chunk Swap: Sincronización



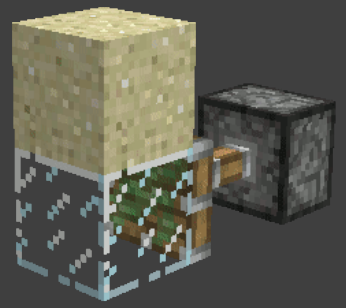
Chunk Swap: Sincronización



Chunk Swap: Sincronización

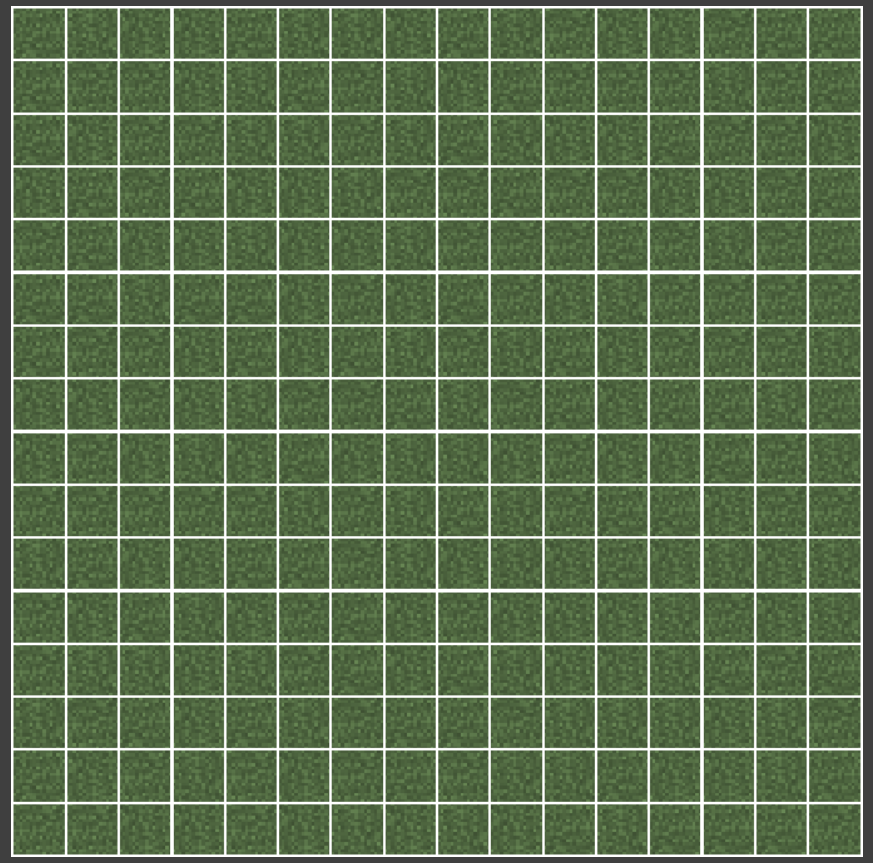


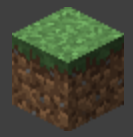
Falling Block Swap



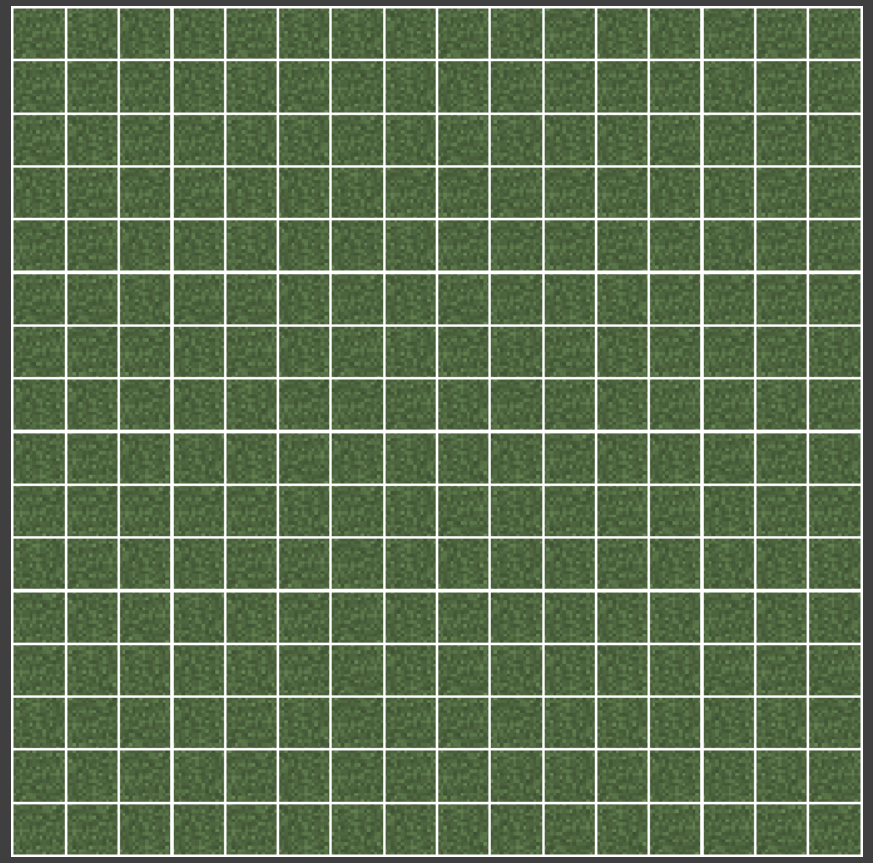
```
# scheduleUpdate
...
currentBlock = world.getBlock(position)
if currentBlock == block:
  currentBlock.update(world, position, currentBlock)
...

# FallingBlock.update
update(world, position, block):
  if not canFall(world, position): return
  if canProcessEntity(world, position):
    spawnEntity(world, position, world.getBlock(position))
```

Word Tearing



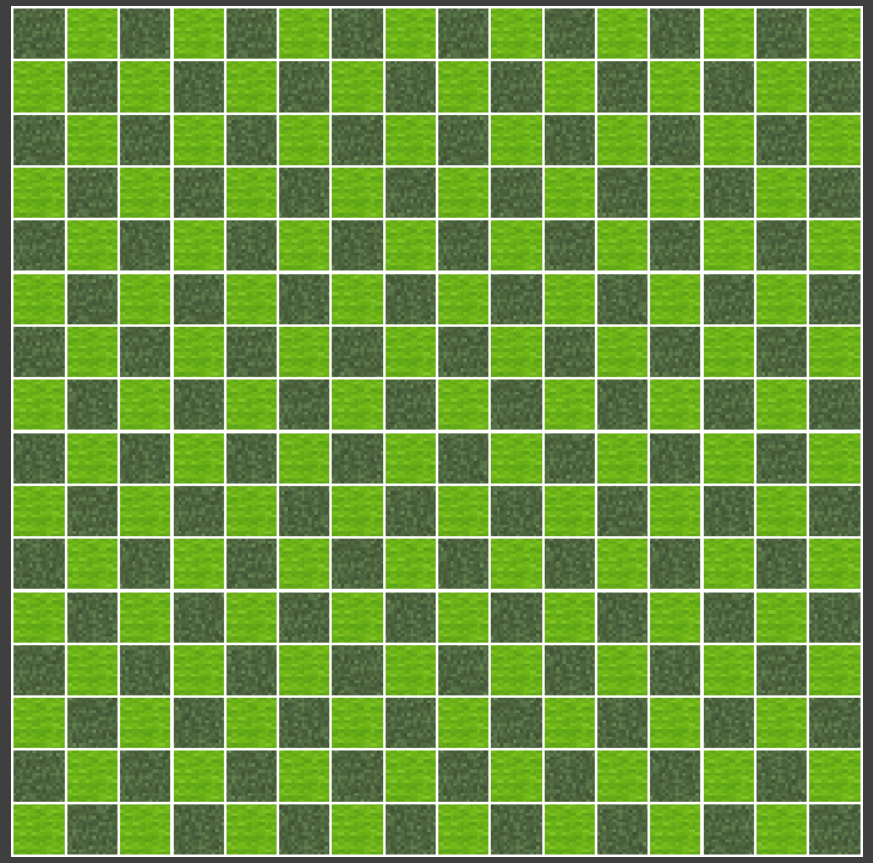
 = 0000000100000

Word Tearing



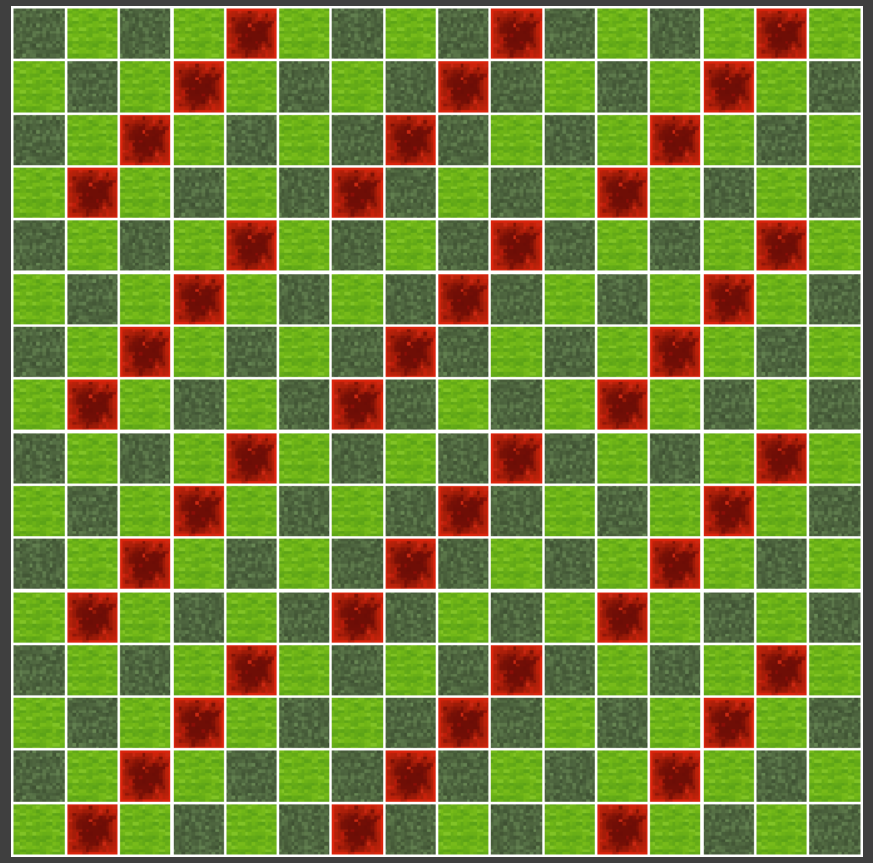
```
   long[] longArray = {
0  0000010000000000001000000000001000000000001000000000001000000
1  0000010000000000001000000000001000000000001000000000001000000
2  0000100000000000100000000000100000000000100000000000100000000
3  0001000000000001000000000001000000000001000000000001000000000
4  0010000000000010000000000010000000000010000000000010000000000
5  0100000000000100000000000100000000000100000000000100000000000
   ...
```

Word Tearing



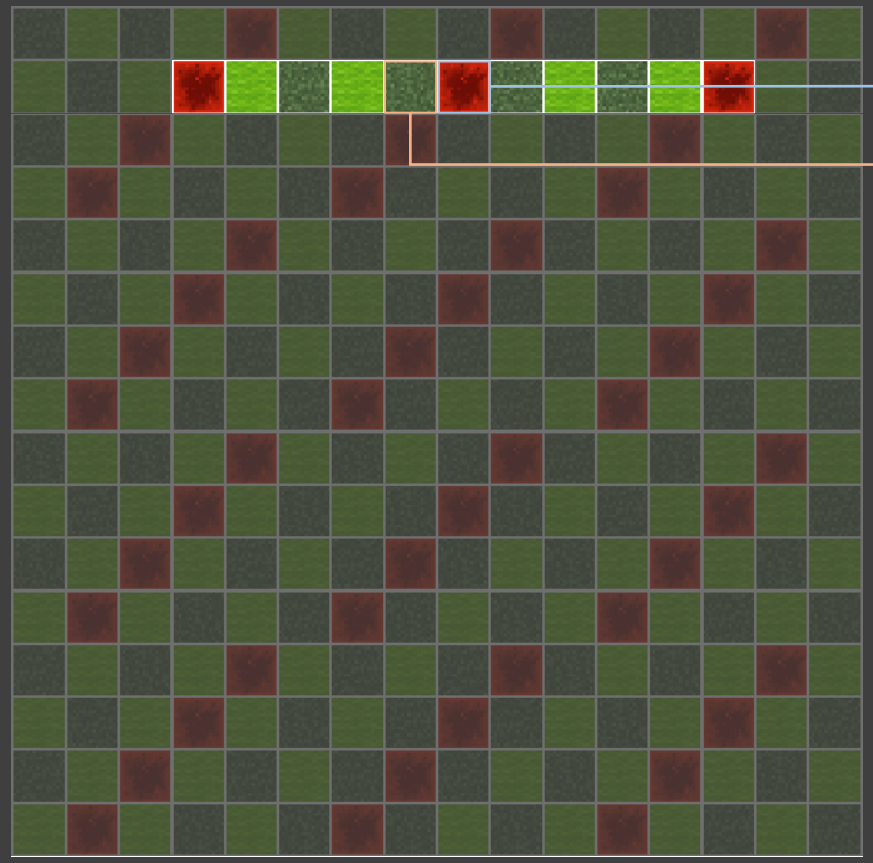
```
  long[] longArray = {
0 0000001000000001000110101000000010000000010001101010000000100000
1 0100011010100000001000000001000110101000000010000000010001101010
2 0000100000000100011010100000001000000001000110101000000010000000
3 0001101010000000100000000100011010100000001000000001000110101000
4 0010000000100011010100000001000000001000110101000000010000000000
5 0110101000000010000000010001101010000000100000000100011010100000
  ...
```

Word Tearing



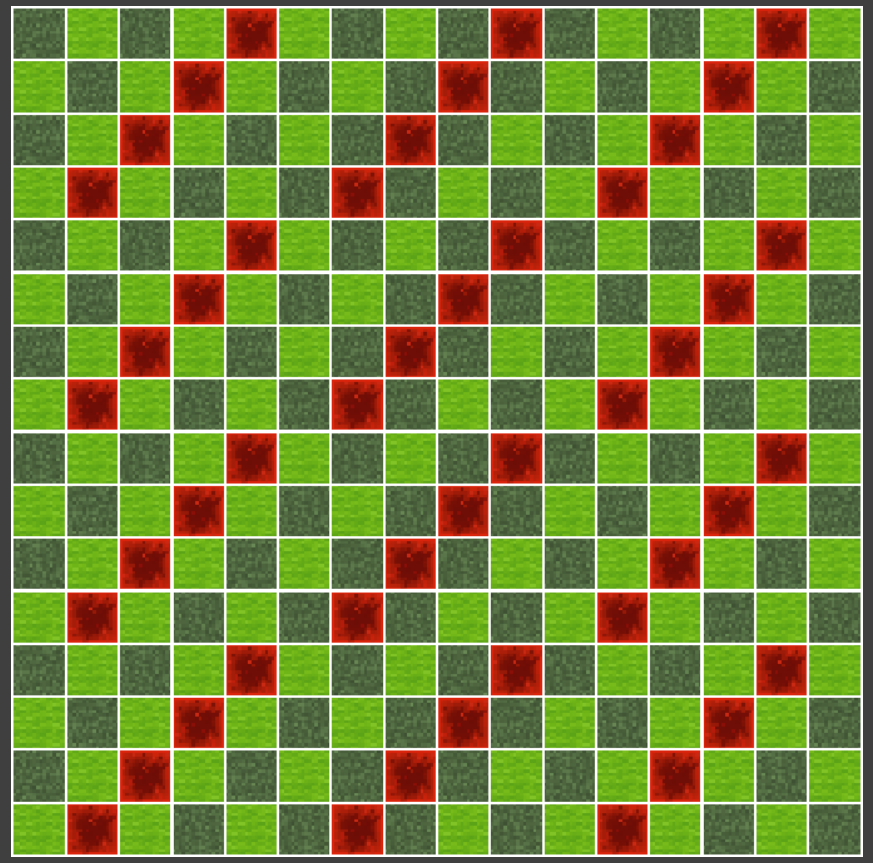
```
   long[] longArray = {
0  1001100000000001000110101000000010000000010001101010000000100000
1  0011000000000000001000000001000110101000000010000000010001101010
2  0110000000000100011010100000001000000001000110101000000010000001
3  1100000000000000100000000100011010100000001000000001000110101010
4  1000000000010001101010000000100000000100011010100000001000000100
5  0000000000000010000000010001101010000000100000000100011010101001
   ...
```

Word Tearing: Ejemplo

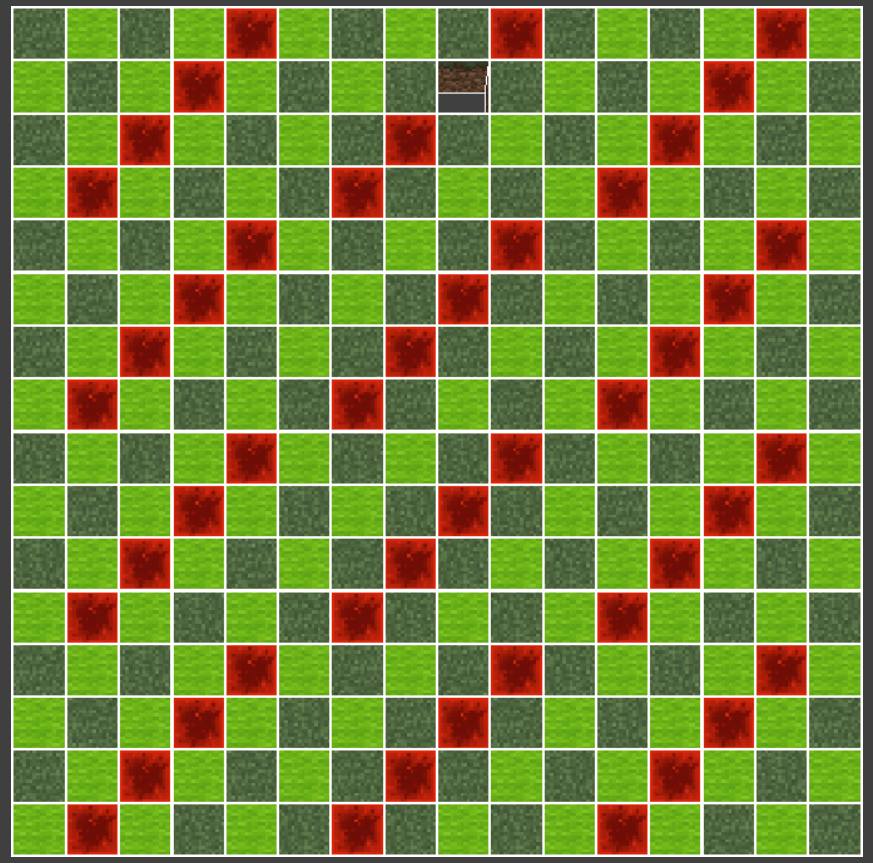


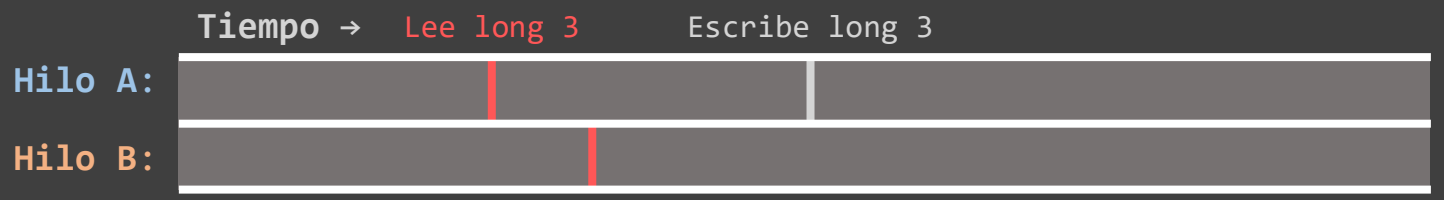

```
   long[] longArray = {
0  1001100000000001000110101000000010000000010001101010000000100000
1  0011000000000000001000000001000110101000000010000000010001101010
2  0110000000000100011010100000001000000001000110101000000010000001
3  1100000000000000010000000100011010100000001000000001000110101010
4  1000000000010001101010000000100000000100011010100000001000000100
5  0000000000000010000000010001101010000000100000000100011010101001
   ...
```

Word Tearing: Ejemplo



Word Tearing: Ejemplo

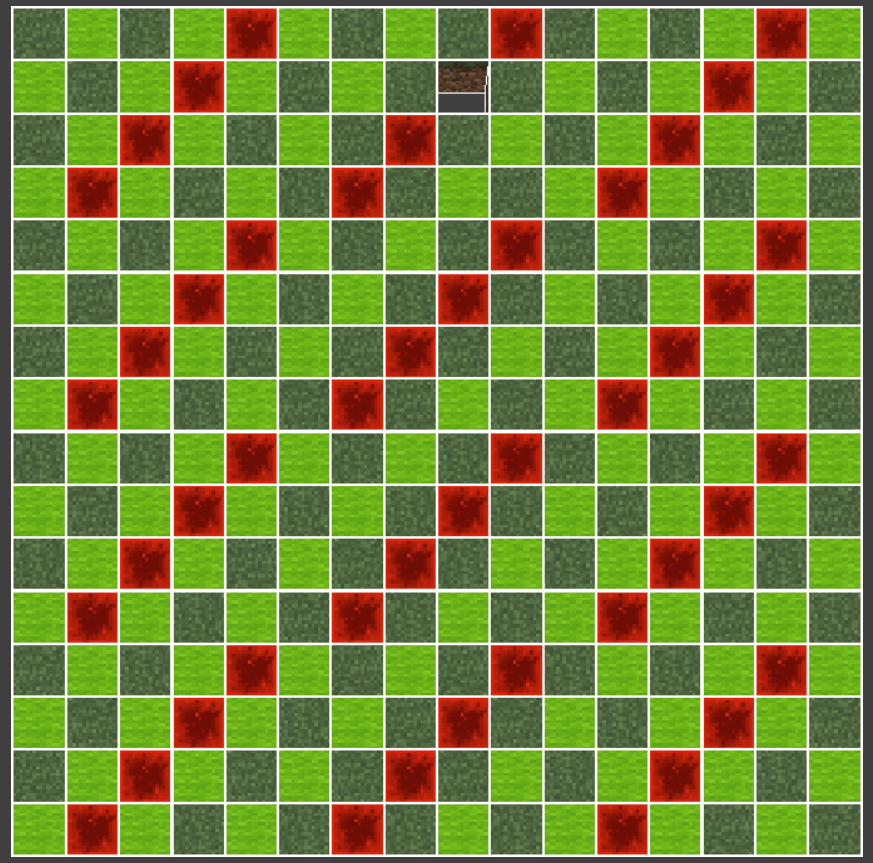




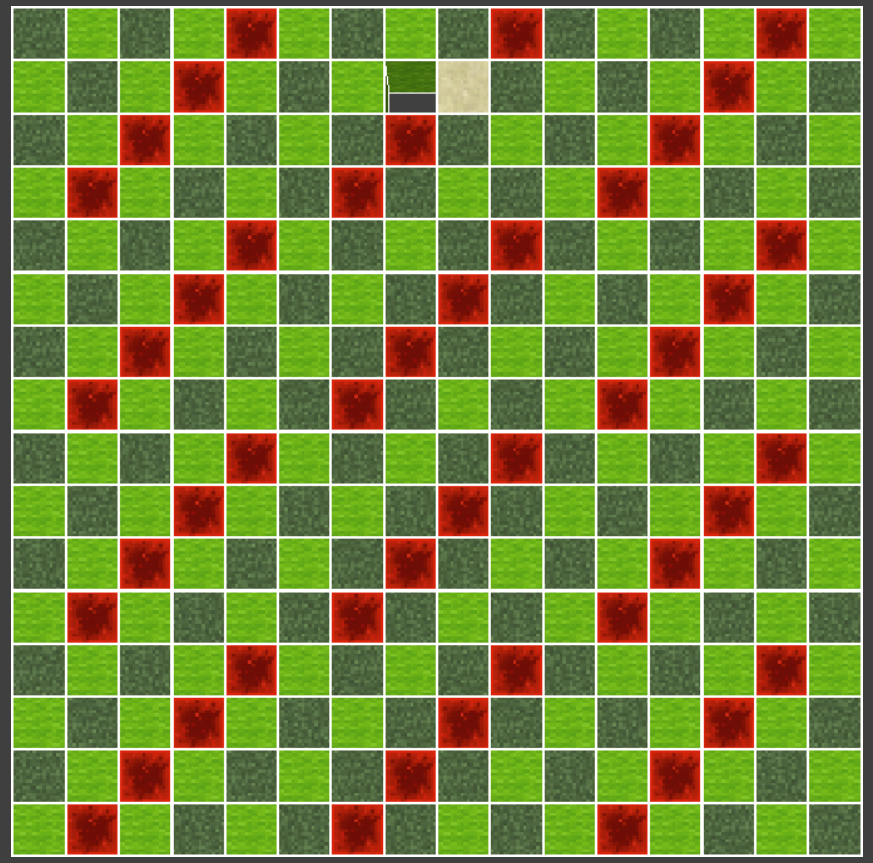
1100000000000000100000000100011010100000001000000000100011010101010

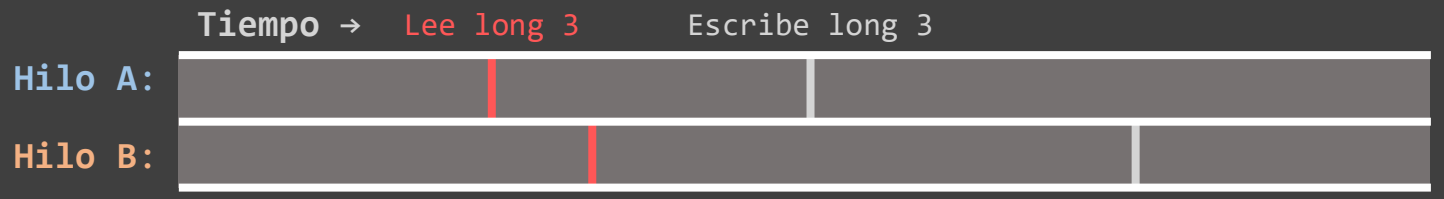
```
long[] longArray = {
0  1001100000000001000110101000000010000000010001101010000000100000
1  0011000000000000001000000001000110101000000010000000010001101010
2  0110000000000100011010100000001000000001000110101000000010000001
3  0000000000000000100000000100011010100000001000000000100011010101010
4  1000000000010001101010000000010000000001000110101000000010000000000
5  0000000000000001000000000100011010100000001000000000100011010101001
   ...
```

Word Tearing: Ejemplo



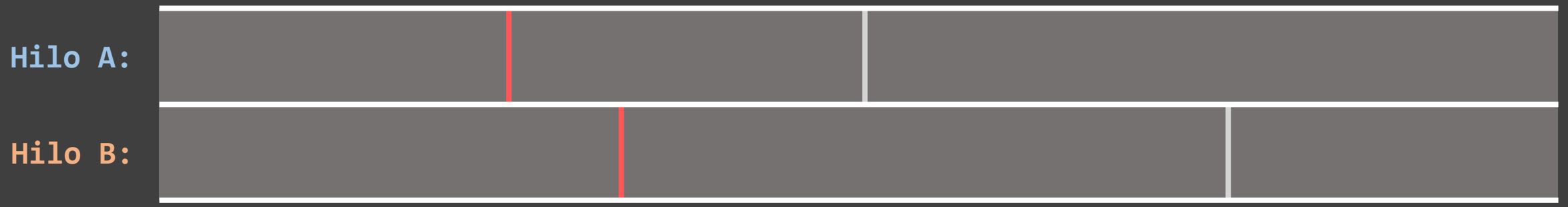
Word Tearing: Ejemplo



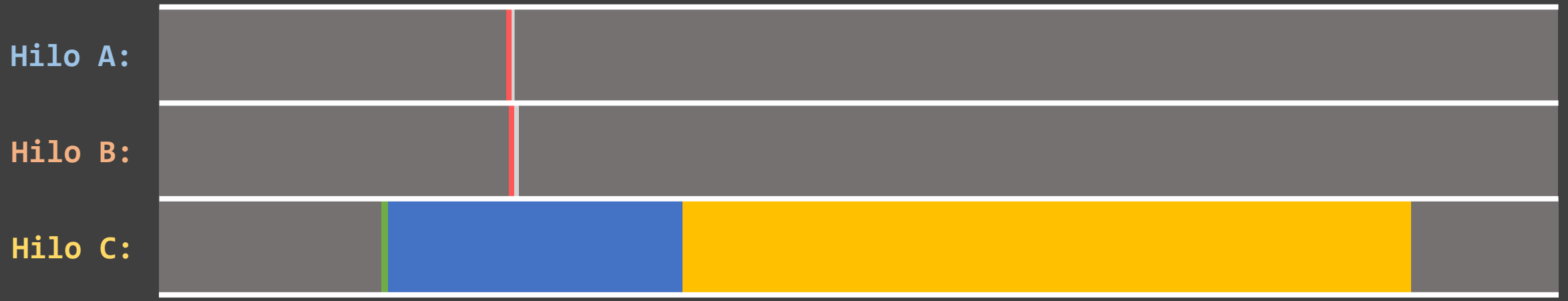



```
   long[] longArray = {
0  1001100000000001000110101000000010000000010001101010000000100000
1  0011000000000000001000000001000110101000000010000000010001101010
2  0110000000000100011010100000001000000001000110101000000010000001
3  1100000000000000000000000100011010100000001000000001000110101010
4  1000000000010001101010000000100000000100011010100000001000000000
5  0000000000000010000000010001101010000000100000000100011010101001
   ...
```

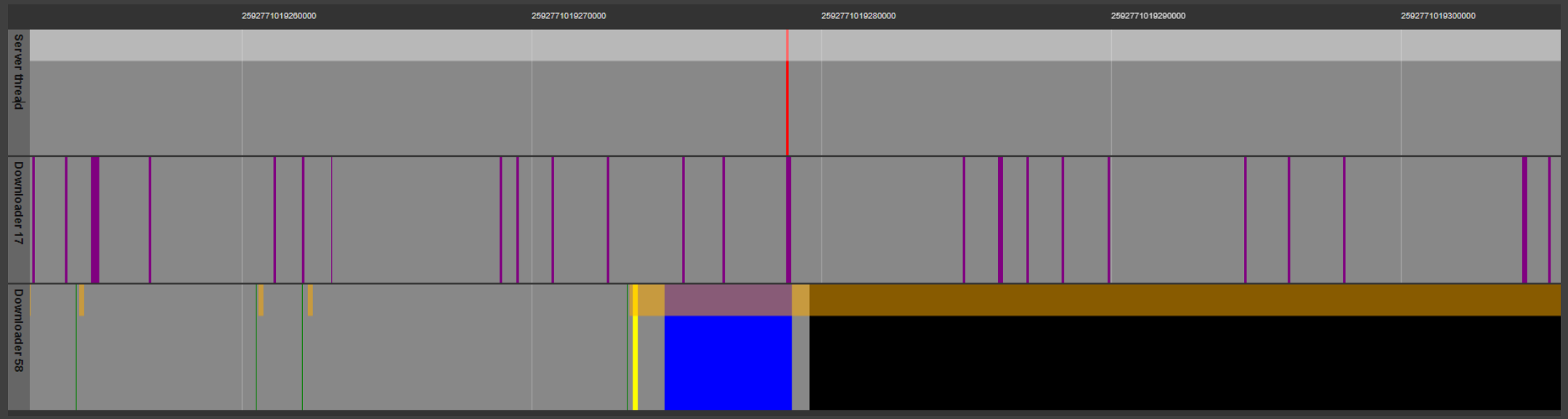
Generic Method



Generic Method



Generic Method



Desarrollo

- Word Tearing: Revisión
- Tile Entity Crash
- PlayerChunkMap Crash

Word Tearing: Revisión


```
this.longArray[startLong] =
  this.longArray[startLong]
  & ~(this.maxValue << relativeBit)
  | (block & this.maxValue) << relativeBit
```

Word Tearing: Revisión


```
this.longArray[startLong] =
  this.longArray[startLong]
  & ~(this.maxValue << relativeBit)
  | (block & this.maxValue) << relativeBit
```

```
; r8 = this.maxValue << relativeBit
mov     r8, r9
shl     r8, cl
; r8 = this.longArray[startLong] & ~r8
andn    r8, r8, qword ptr [rax + rsi * 8 + 18h]
; r9 = (block & this.maxValue) << relative_bit
and     rbx, r9
mov     r9, rbx
shl     r9, cl
; this.longArray[start_long] = r8 | r9
or      r8, r9
mov     qword ptr [rax + rsi * 8 + 18h], r8
```

Tile Entity Crash


```
updateColorAsync(world, position):
  Thread.run:
    while position.y >= 0:
      block = world.getBlock(position)
      if block is BEACON:
        world.addSynchronizedTask(lambda:
          block.updateColor(position))
      position = position.down()
```

Tile Entity Crash

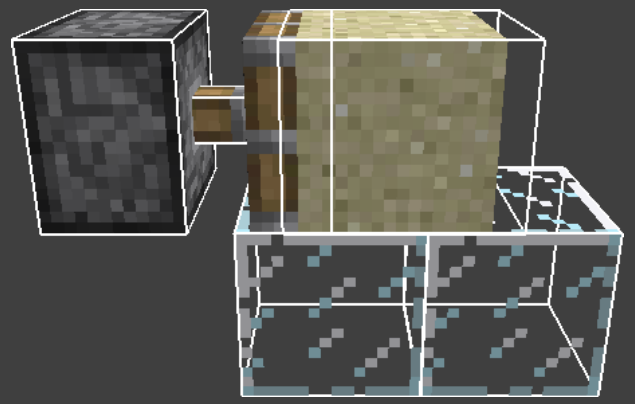

```
updateColorAsync(world, position):
  Thread.run:
    try:
      while position.y >= 0:
        block = world.getBlock(position)
        if block is BEACON:
          world.addSynchronizedTask(lambda:
            block.updateColor(position))
        position = position.down()
    catch exception:
      print(exception)
```

```
Caused by: NullPointerException
at World.getPendingTileEntity (line 2413)
at World.getTileEntity (line 2398)
at BlockPistonMoving.neighborChanged (line 172)
at World.notifyBlock (line 582)
at World.notifyNeighbors (line 550)
...
```

Tile Entity Crash

Tile Entity Crash

Tile Entity Crash



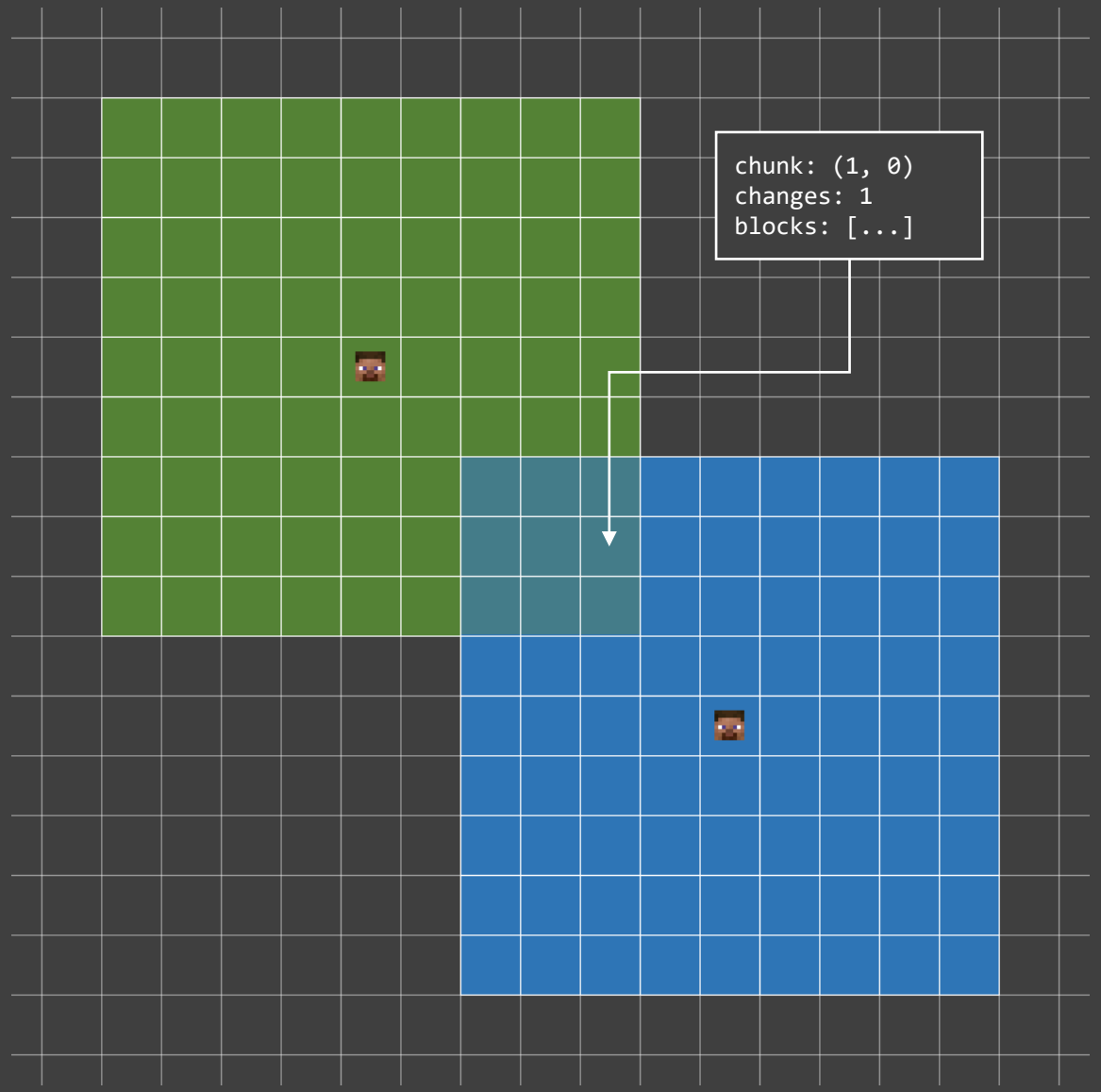
```
# BlockPistonMoving
neighborChanged(block, world, position):
  world.getTileEntity(position)

# World
getTileEntity(position):
  tileEntity = getChunk(position)
               .getTileEntity(position)

  if tileEntity is null:
    tileEntity = getPendingTileEntity(position)

  return tileEntity
```

Tile Entity Crash


```
getPendingTileEntity(position):
  i = 0
  while i < pendingTileEntities.size():
    tileEntity = pendingTileEntities.get(i)
    if tileEntity.position == position:
      return tileEntity
    i += 1
  return null
```

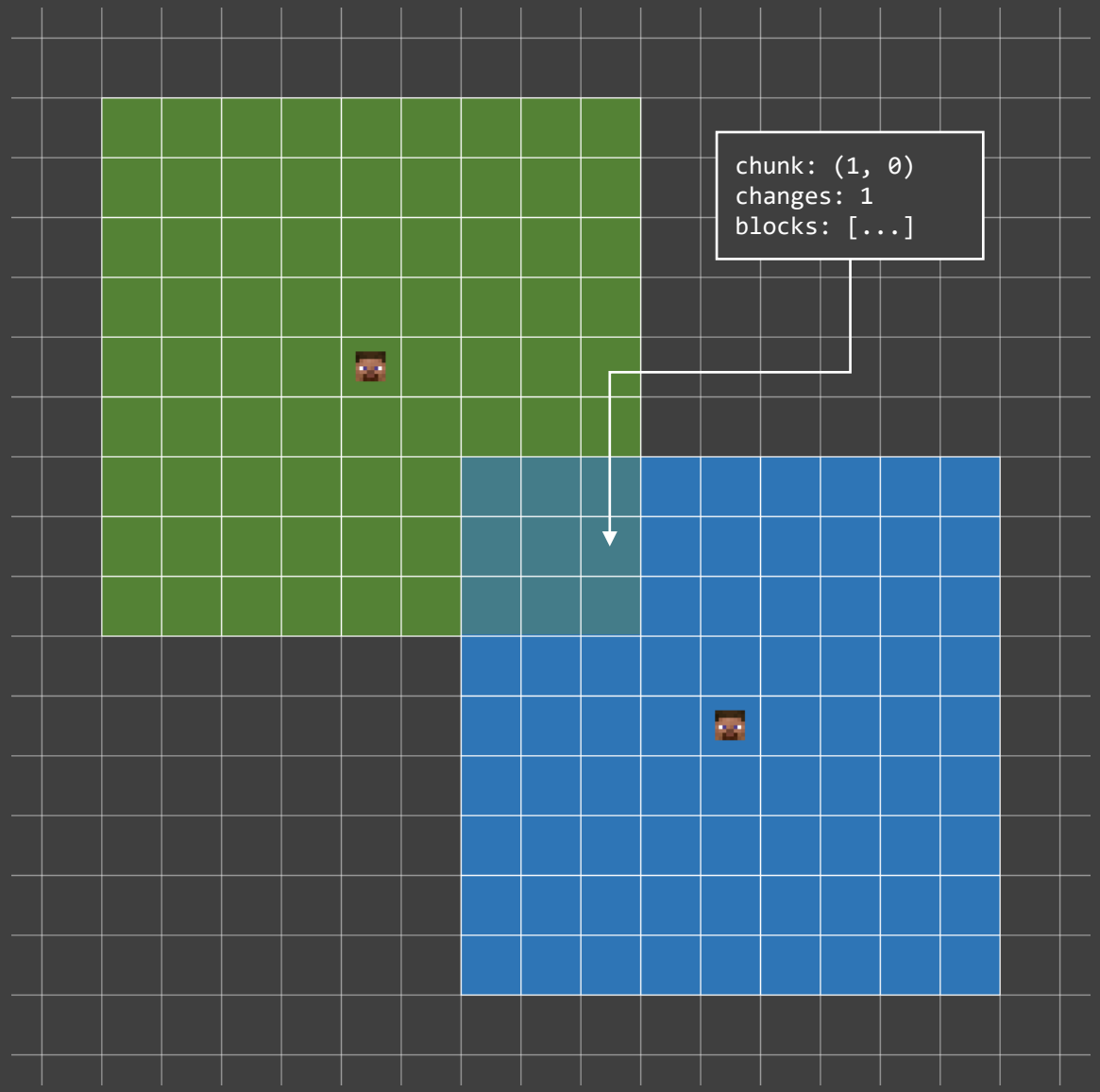
```
# java.util.ArrayList
clear():
  # clear to let GC do its work
  i = 0
  while i < size:
    elements[i] = null
  size = 0
```

PlayerChunkMap Crash




```
# PlayerChunkMap.tick
  ...

  if not entriesToUpdate.isEmpty():
    for entry in entriesToUpdate:
      entry.update()
    entriesToUpdate.clear()

  ...
```

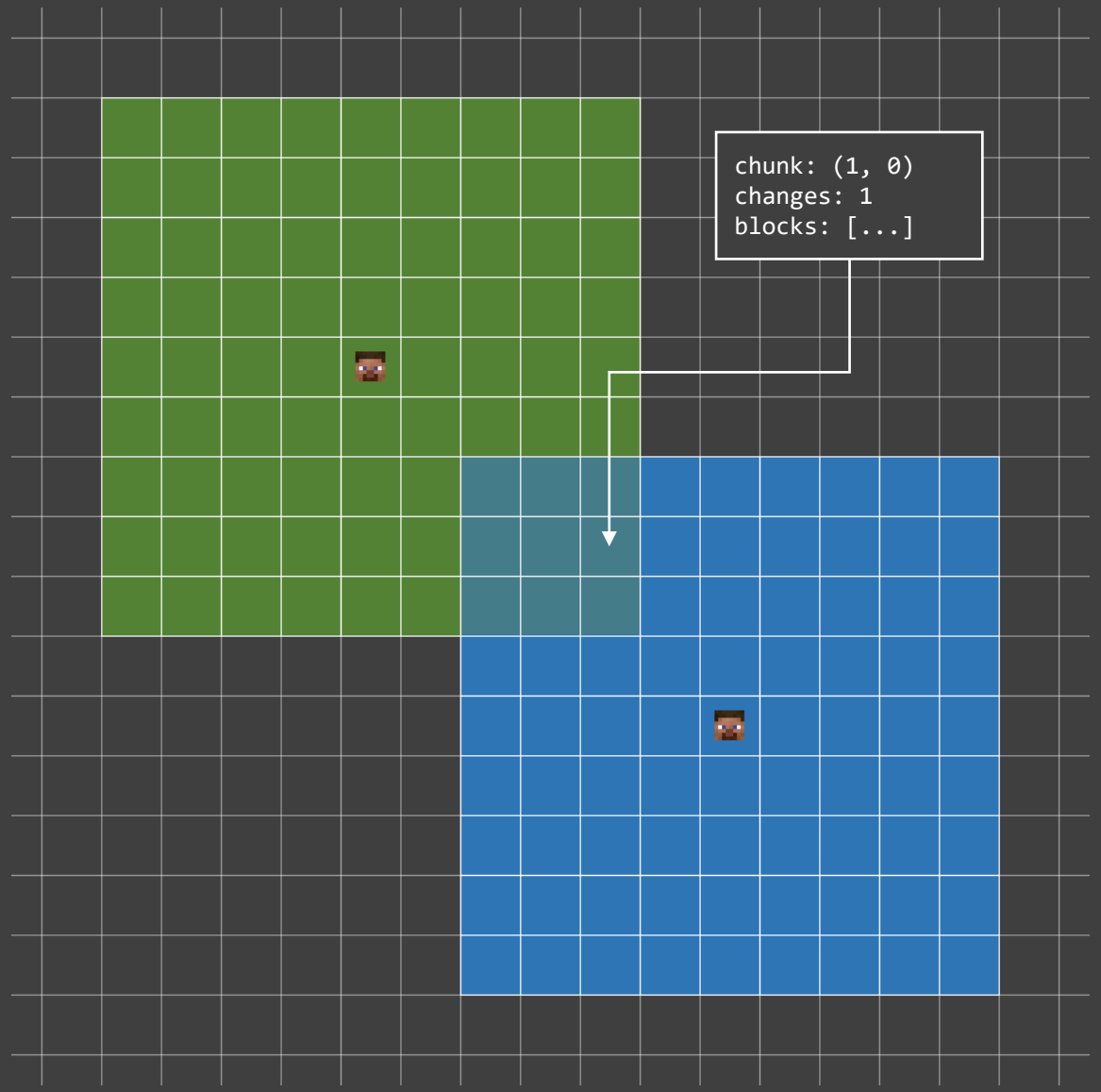
PlayerChunkMap Crash




```
# PlayerChunkMap.tick
  ...

  if not entriesToUpdate.isEmpty():
    for entry in entriesToUpdate:
      entry.update()
    entriesToUpdate.clear()

  ...
```

```
# PlayerChunkMapEntry.blockChanged
  if changes == 0:
    PlayerChunkMap.addEntry(this)
  ...
```

PlayerChunkMap Crash




```
# PlayerChunkMap.tick
  if time - lastUpdate > 8000:
    lastUpdate = time
    for entry in entries:
      entry.update()

  if not entriesToUpdate.isEmpty():
    for entry in entriesToUpdate:
      entry.update()
    entriesToUpdate.clear()

  ...
```

```
# PlayerChunkMapEntry.blockChanged
  if changes == 0:
    PlayerChunkMap.addEntry(this)
  ...
```

Conclusiones

- ¡Hemos conseguido nuestro objetivo!
- Aún quedan cosas por mejorar y descubrir de los efectos de este exploit
- Probablemente no vuelva a darse un caso similar
- El mundo de los exploits es muy complejo e interesante de estudiar

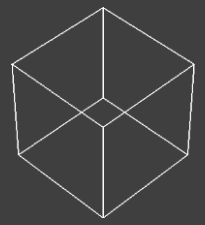
La evolución de los errores de programación en los videojuegos: El curioso caso de Minecraft

Adrián Muelas Gómez

Generic Method: Paleta

Bonus: 1

0000011000000



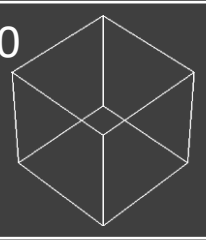
0000000000000

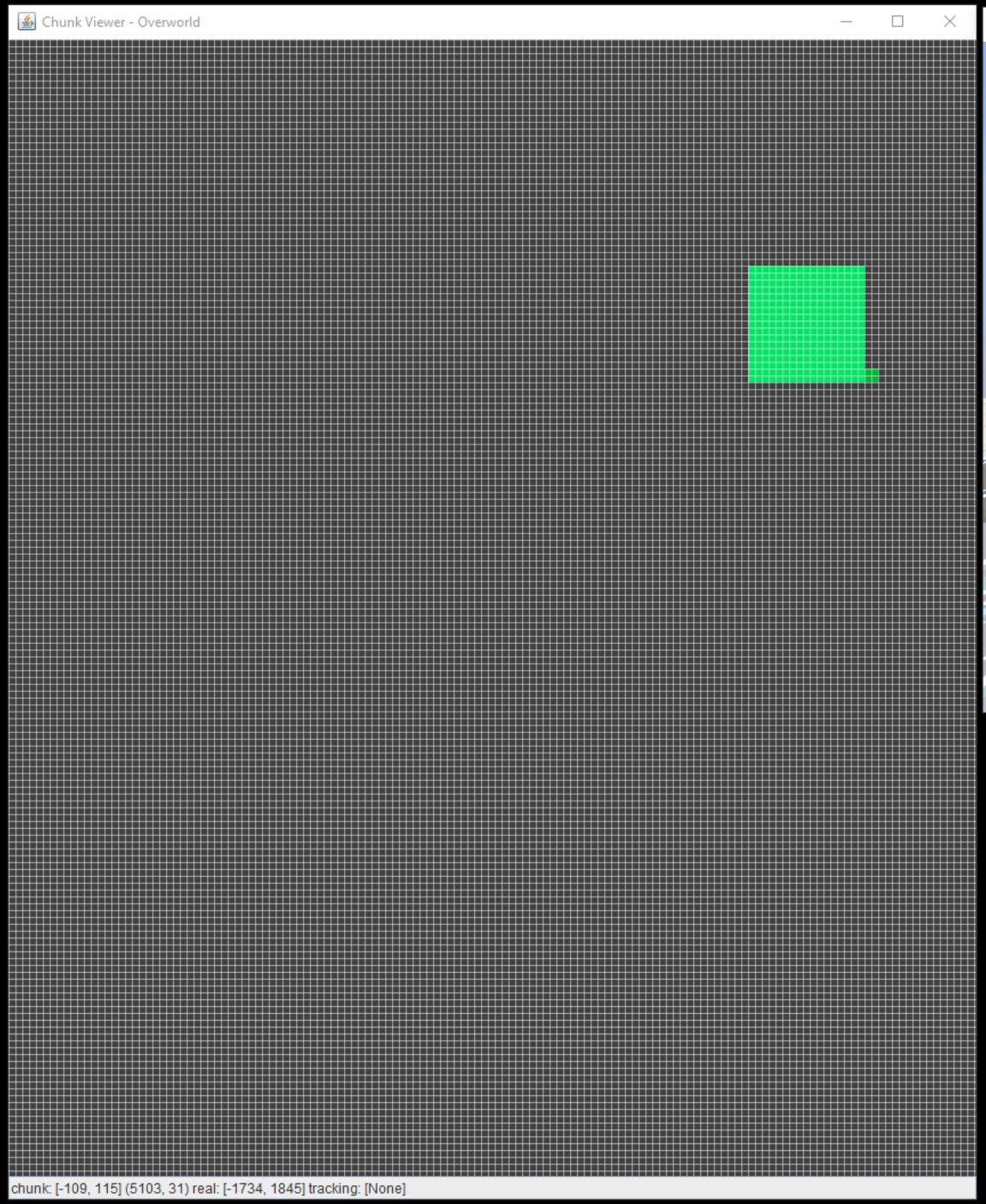
0100010010000

011

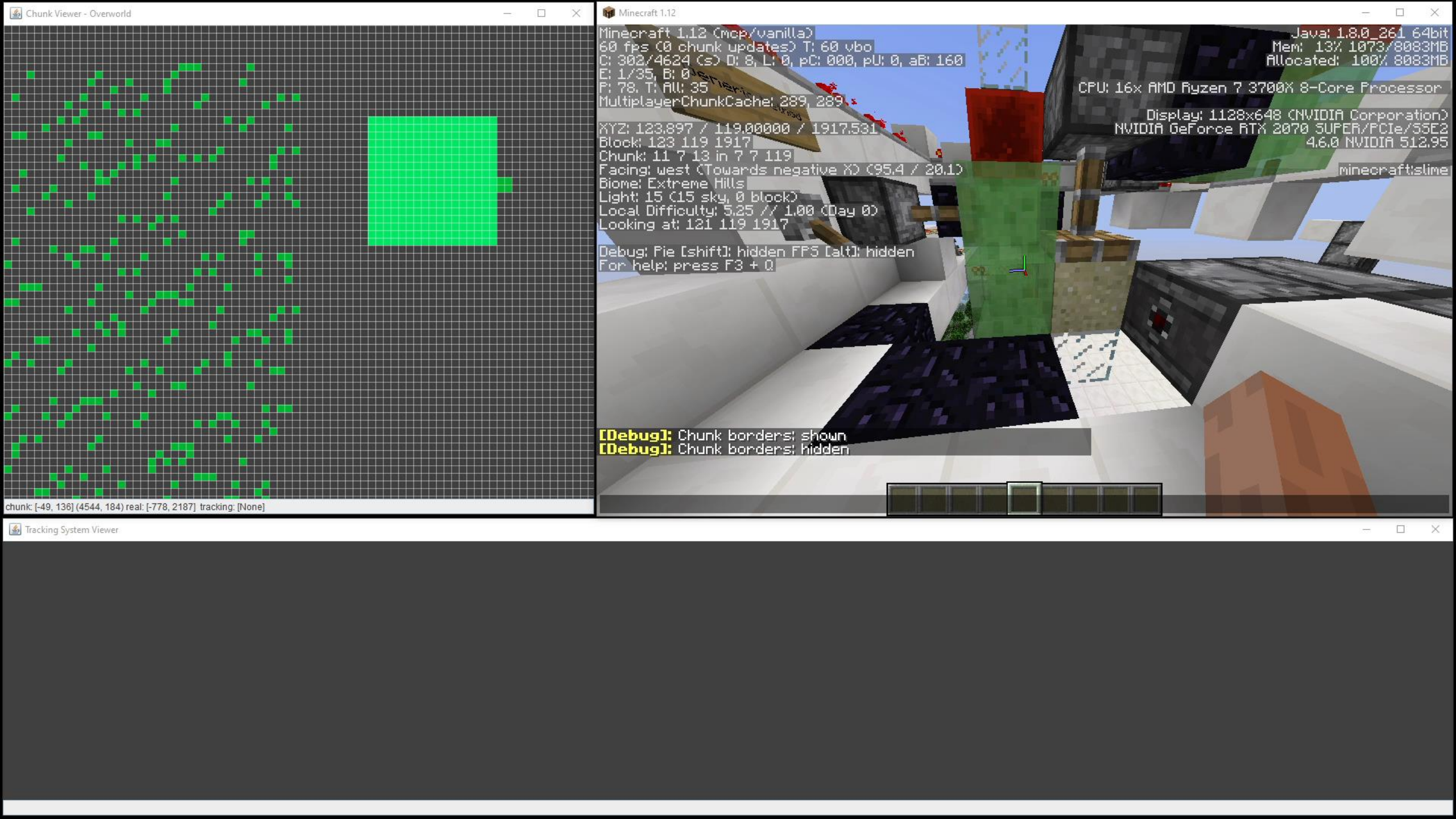
000

010

0	1	2	3	4	5	6	7
							


Chunk Viewer - Overworld
chunk: [-109, 115] (5103, 31) real: [-1734, 1845] tracking: [None]


Minecraft 1.12


Chunk Viewer - Overworld
chunk: [-49, 136] (4544, 184) real: [-778, 2187] tracking: [None]
Minecraft 1.12
Minecraft 1.12 (mcp/vanilla)
60 fps (0 chunk updates) T: 60 vbo
C: 302/4624 (s) D: 8, L: 0, pC: 000, pU: 0, aB: 160
E: 1/35, B: 0
F: 78, T: All: 35
MultiplayerChunkCache: 289, 289
XYZ: 123.897 / 119.00000 / 1917.531
Block: 123 119 1917
Chunk: 11 7 13 in 7 7 119
Facing: west (Towards negative X) (95.4 / 20.1)
Biome: Extreme Hills
Light: 15 (15 sky, 0 block)
Local Difficulty: 5.25 // 1.00 (Day 0)
Looking at: 121 119 1917
Debug: Pie [shift]: hidden FPS [alt]: hidden
For help: press F3 + Q
Java: 1.8.0_261 64bit
Mem: 13% 1073/8083MB
Allocated: 100% 8083MB
CPU: 16x AMD Ryzen 7 3700X 8-Core Processor
Display: 1128x648 (NVIDIA Corporation)
NVIDIA GeForce RTX 2070 SUPER/PCIe/SSE2
4.6.0 NVIDIA 512.95
minecraft:slime
[Debug]: Chunk borders: shown
[Debug]: Chunk borders: hidden
Tracking System Viewer