Thinking Outside the Box

Who the H*ll Are You?

Name:	Camden Pettijohn
Species:	Gamer (subspecies: crankin’ 90’s)
Roles:	GDAC co-director, SING lab intern
Skills:	programming, drawing, spending so much time making goofy inventions

You may also recognize me as...

Back to the Subject...

"To think outside the box" means to think:

- differently
- unconventionally
- or from a new perspective

This sentiment is visible in game development!

Game Boy

Released: April 21, 1989 (Japan)

RAM size: 8 KiB or 8192 bytes

RAM size for PC gaming today:
16 GB or 17,179,869,184 bytes

Game Boy Games

Race Drivin'

Released on March 12, 1993 (North America)
By Atari Games and Argonaut Software

Race Drivin’

What did it do that was so impressive?

- Despite the limited amount of RAM, it manages to display pseudo-3D graphics
- The road, the ramps, the cars, and many filled polygons - very intensive!
- Not the best framerate... but whatever

One Step Further

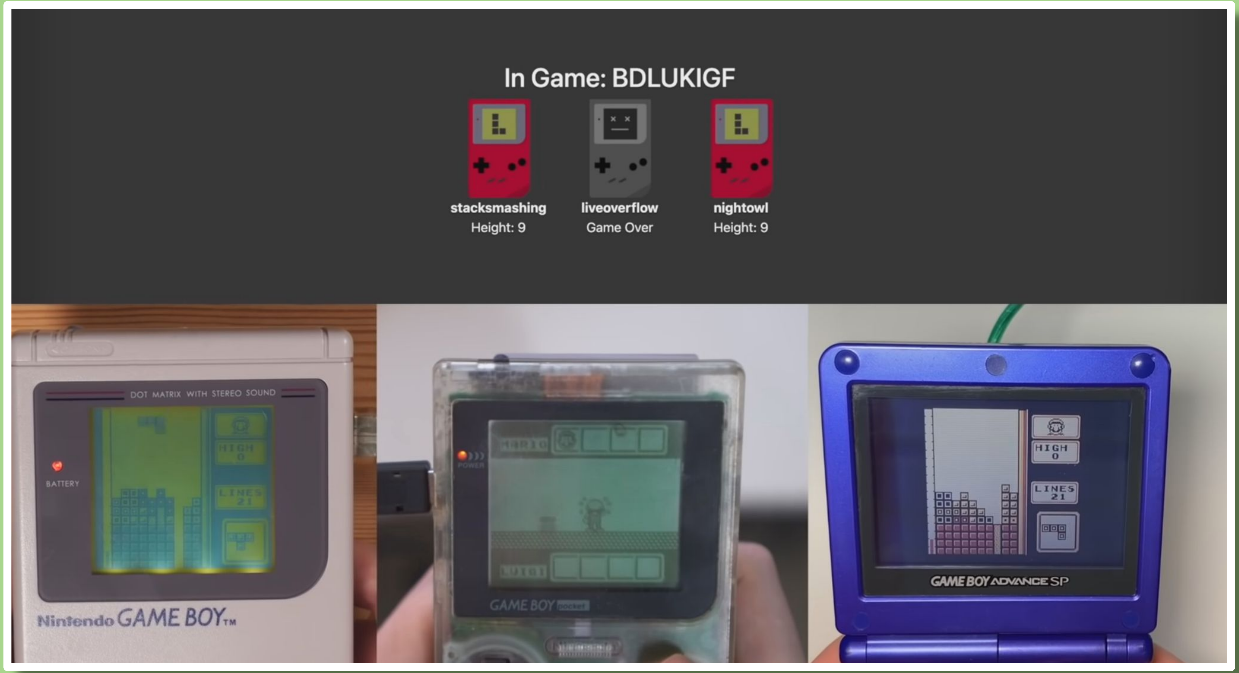
Hence, you could say that the developers of **Race Drivin'** "thought outside the box" to create pseudo-3D graphics

But, perhaps we take it one step further...

There has been **29 years of new technology!**

(Some metagaming?)

Gameboy Example

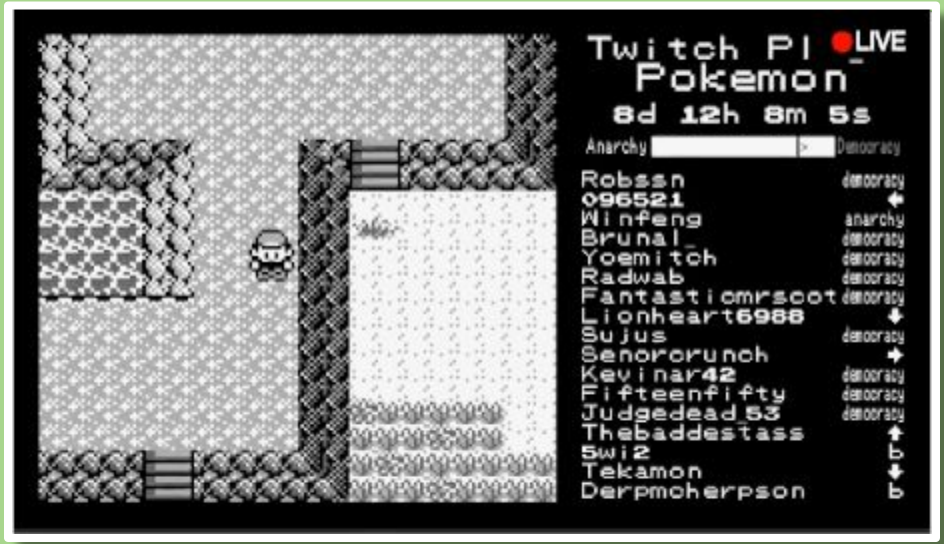


Game Boy Example

Game Boy Example

What Have I Done?

I created something similar
to **Twitch Plays Pokémon**



Robssn	democracy
096521	←
Winfeng	anarchy
Brunal_	democracy
Yoemitch	democracy
Radwab	democracy
Fantasticmrscoot	democracy
Lionheart6988	↓
Sujus	democracy
Senororunch	→
Kevinar42	democracy
Fifteenfifty	democracy
Judgedead_53	democracy
Thebaddestass	↑
Swi2	b
Tekamon	↓
Derpmoherpson	b

SNES-bot

How SNES-bot Works

How it works:

- **Python Discord bot**
  (to convert Discord commands and reactions
  into PS4 controller signals)
- **Various Adapters**
  (to convert PS4 controller signals
  into Super Nintendo controller signals)
- **Super Nintendo**
- **Super Game Boy** (to play your Game Boy games on)
- **Capture Card** (to stream to the Discord users)

```python
async def controller_input(button):
    global sock, address, counter, vertical, horizontal
    if counter >= 5: return
    counter += 1
    packet = bytearray([0x01] * 2 + [button] + [0x00] * 3 + [0xff])
    if button in [0x83, 0x85, 0x86, 0x84]:
        vertical_unique = 0
        horizontal_unique = 0
        if button in [0x83, 0x85]:
            vertical += 1
            vertical_unique = vertical
        else:
            horizontal += 1
            horizontal_unique = horizontal
        await asyncio.sleep(0.005)
        sock.sendto(packet, address)
        timer = 0
        while timer < wait:
            if vertical_unique and vertical_unique != vertical: break
            elif horizontal_unique and horizontal_unique != horizontal: break
            await asyncio.sleep(0.01)
            timer += 0.01
    else:
        sock.sendto(packet, address)
        await asyncio.sleep(0.05)
    counter -= 1
    packet[-1] = 0x00
    sock.sendto(packet, address)
```

```
async def controller_input(button):
    global sock, address, counter, vertical, horizontal
    if counter >= 5: return
    counter += 1
    packet = bytearray([0x01] * 2 + [button] + [0x00] * 3 + [0xff])
    if button in [0x83, 0x85, 0x86, 0x84]:
        vertical_unique = 0
        horizontal_unique = 0
        if button in [0x83, 0x85]:
            vertical += 1
            [illegible]er[illegible]al unique = vertical
        els[illegible]
            [illegible]izo[illegible]l [illegible] 1
            [illegible]riz[illegible]al[illegible]i[illegible] = [illegible]izont[illegible]
        awa[illegible] asy[illegible]o.[illegible]ee[illegible].0[illegible]
        sock.sendto(packet, [illegible]ss)
        timer = 0
        while timer < wait:
            if vertical_unique and vertical_unique != vertical: break
            elif horizontal_unique and horizontal_unique != horizontal: break
            await asyncio.sleep(0.01)
            timer += 0.01
    else:
        sock.sendto(packet, address)
        await asyncio.sleep(0.05)
    counter -= 1
    packet[-1] = 0x00
    sock.sendto(packet, address)
```

YES, YOU'RE RIGHT

```
async def controller_input(button):
    global sock, address, counter, vertical, horizontal
    if counter >= 5: return
    counter += 1
    packet = bytearray([0x01] * 2 + [button] + [0x00] * 3 + [0xff])
    if button in [0x83, 0x85, 0x86, 0x84]:
        vertical_unique = 0
        horizontal_uniq[illegible]
        if butt[illegible] [0x[illegible] 0x[illegible]
            ver[illegible] += [illegible]
            ver[illegible]uniq[illegible] = [illegible]tica[illegible]
        els[illegible]
            [illegible]
            [illegible]al_uni[illegible] = [illegible]izont[illegible]
        awa[illegible] asy[illegible].[illegible]ee[illegible].0[illegible]
        sock.sendto(packet, [illegible])
        timer = 0[illegible]
        [illegible]tim[illegible] < w[illegible]:
            if ve[illegible]cal[illegible]iq[illegible]nd [illegible]tica[illegible]nique != [illegible]ic[illegible] br[illegible]
            elif h[illegible]ontal_u[illegible]ue [illegible] hor[illegible]tal_unique[illegible]orizontal: break
            [illegible]t [illegible]cio.sle[illegible]
            t[illegible]r [illegible].01
    else:
        sock.sendto(packet, address)
        await asyncio.sleep(0.05)
    counter -= 1
    packet[-1] = 0x00
    sock.sendto(packet, address)
```

But What About Pico-8?

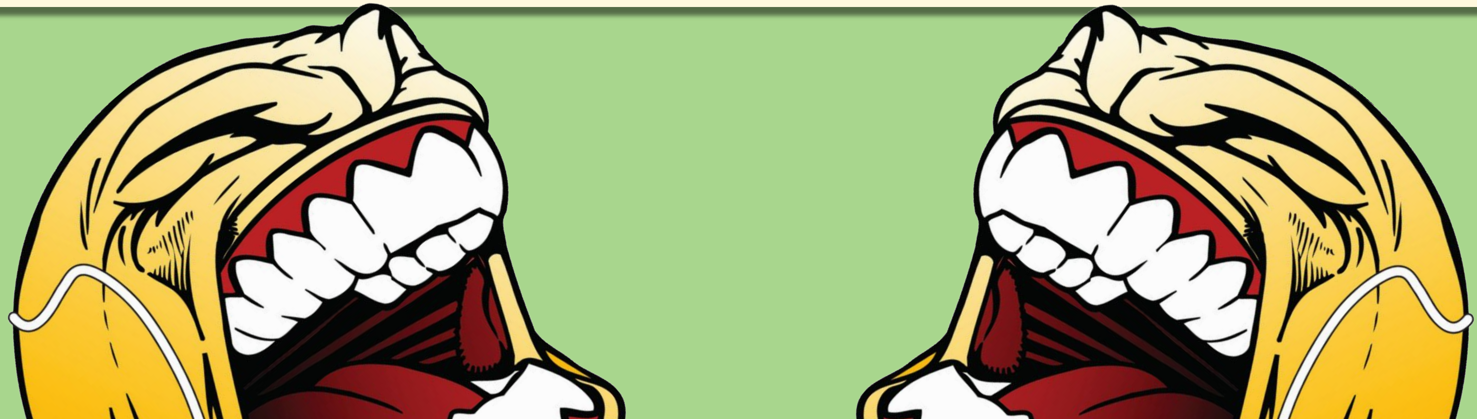
But What About Pico-8?

~ yes :D ~

Pico-8 LAN Multiplayer

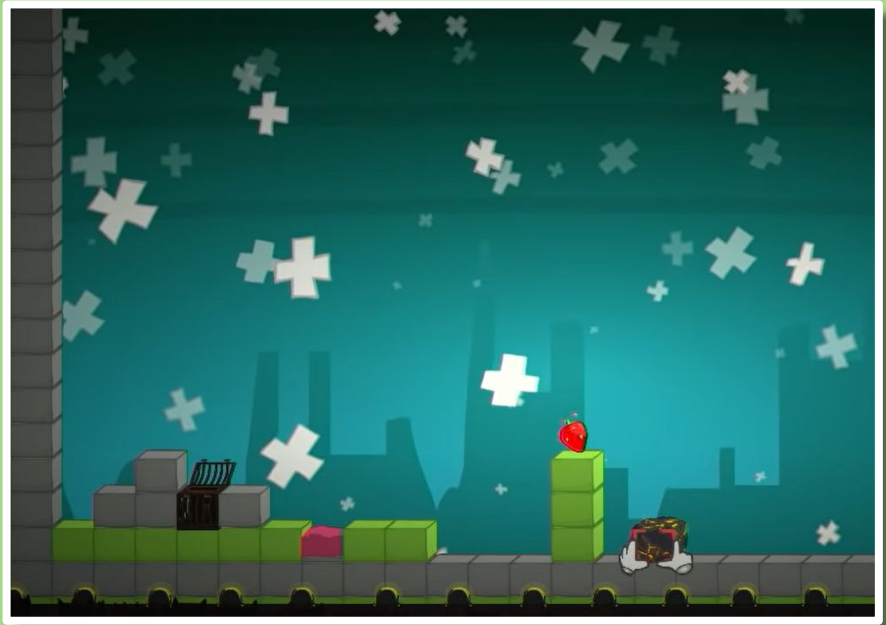
Pico-8 Discord bot

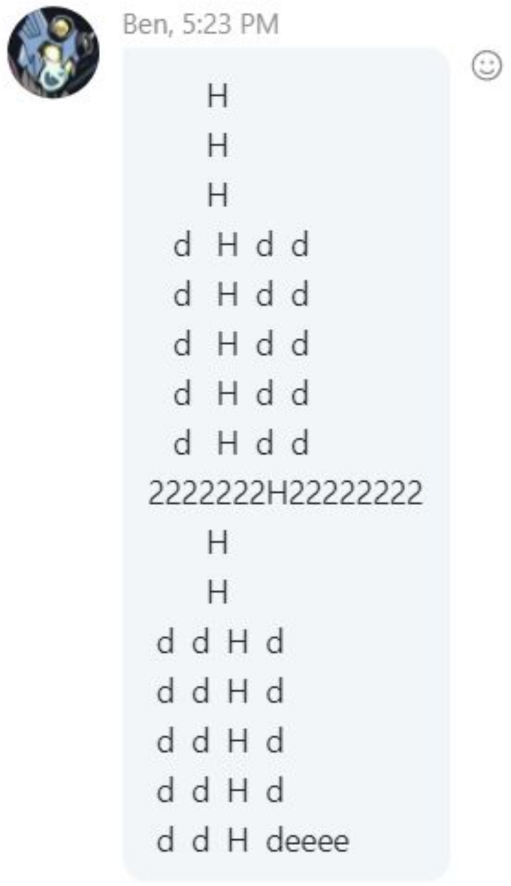
Tool being used: Pico-8’s **serial** function for piping



BBT-like Level Editor

Tool being used: Pico-8's **stat(4)** to read the clipboard





Ben, 5:23 PM

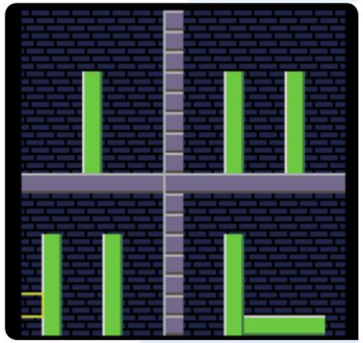
```
     H
     H
     H
  d  H d d
  d  H d d
  d  H d d
  d  H d d
  d  H d d
2222222H22222222
     H
     H
d d H d
d d H d
d d H d
d d H d
d d H deeee
```

Ben, 5:23 PM

```
   H
   H
   H
 d  H d d
 d  H d d
 d  H d d
 d  H d d
 d  H d d
2222222H22222222
   H
   H
d d H d
d d H d
d d H d
d d H d
d d H deeee
```

6:51 PM

Lemme go paste it in



You are no longer my friend

Ben, 7:00 PM

ahahahahaha

>:3

How It All Works

a lot of coding

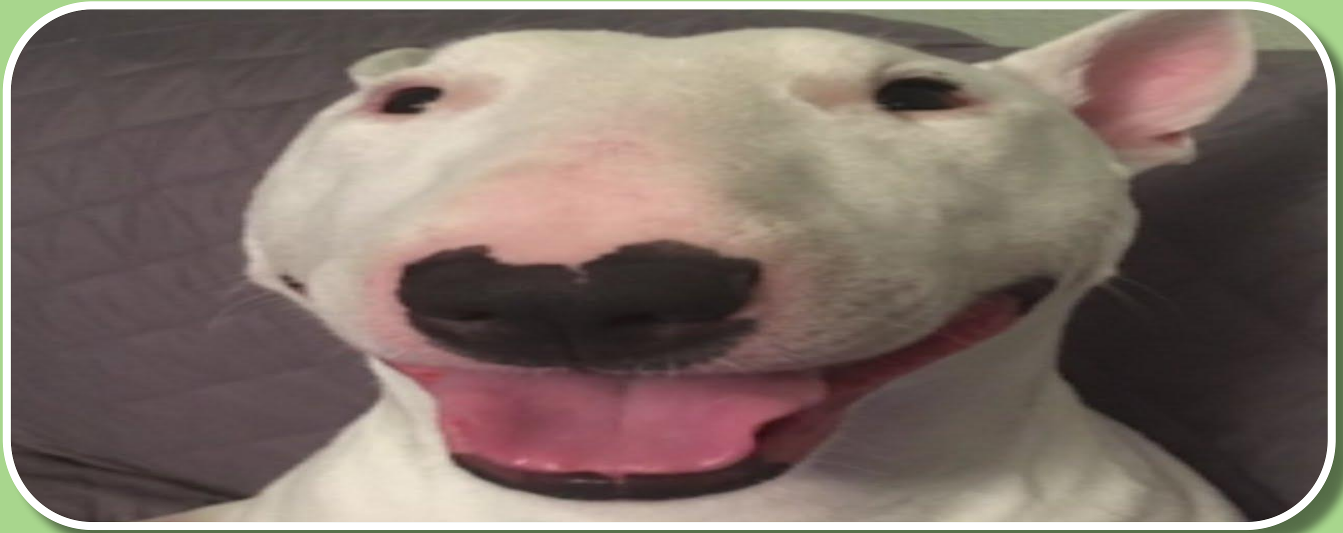
Closing Remarks

You do not need to implement the most advanced system ever conceived to “think outside the box”

All it requires is to view a tool from a different angle to discover more possibilities

And maybe that sparks a new and **great idea**

Thanks for having me!



Nelson picture for you!