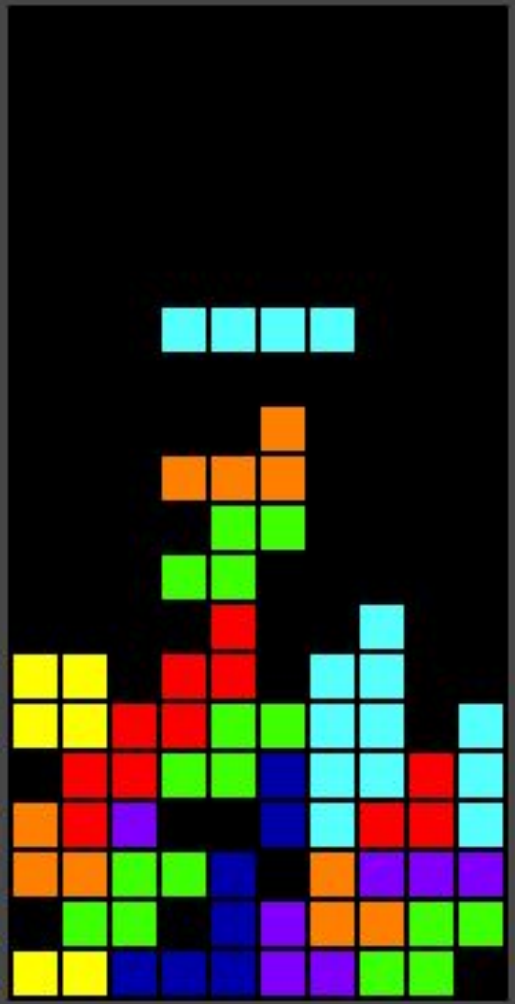


Tetris in xv6

Luke Jaffe

Problem

- Display graphics in 320x200 VGA with xv6
- Use keyboard interrupts to update shared memory
- Combine these features to create Tetris

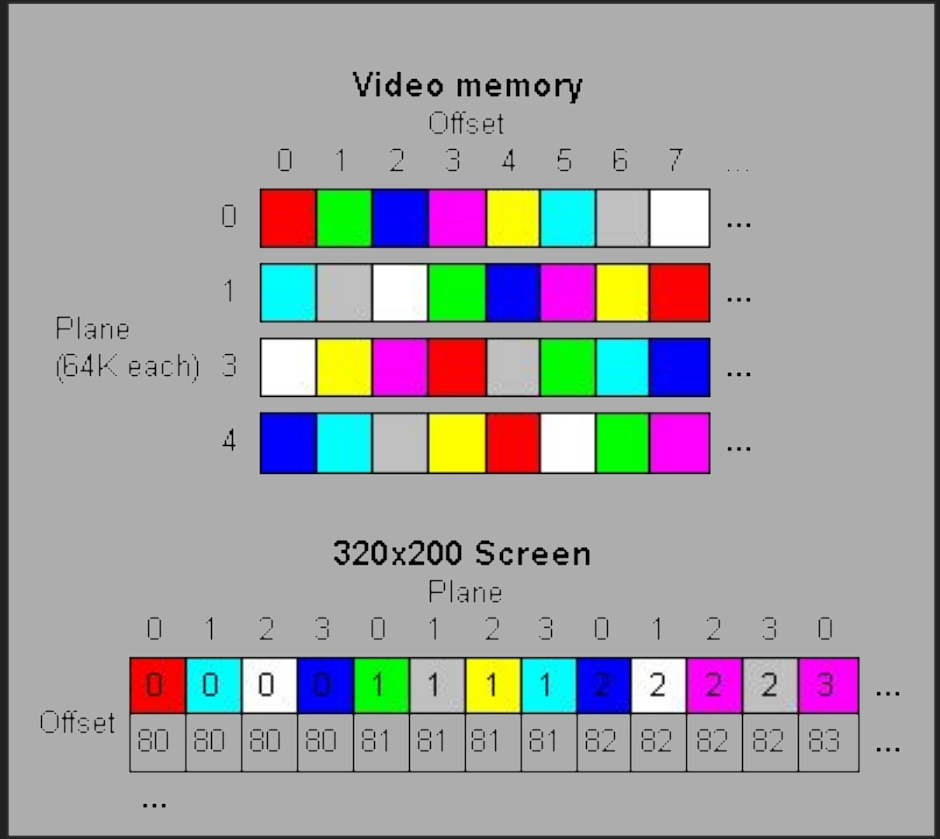
VGA Display

- 320x200 256-color VGA display
- BIOS interrupt 0x10
  - Mode 0x13 displays VGA
  - Mode 0x03 displays text
- Disable interrupts
- Switch to real mode (address anything)

Default Color Palette

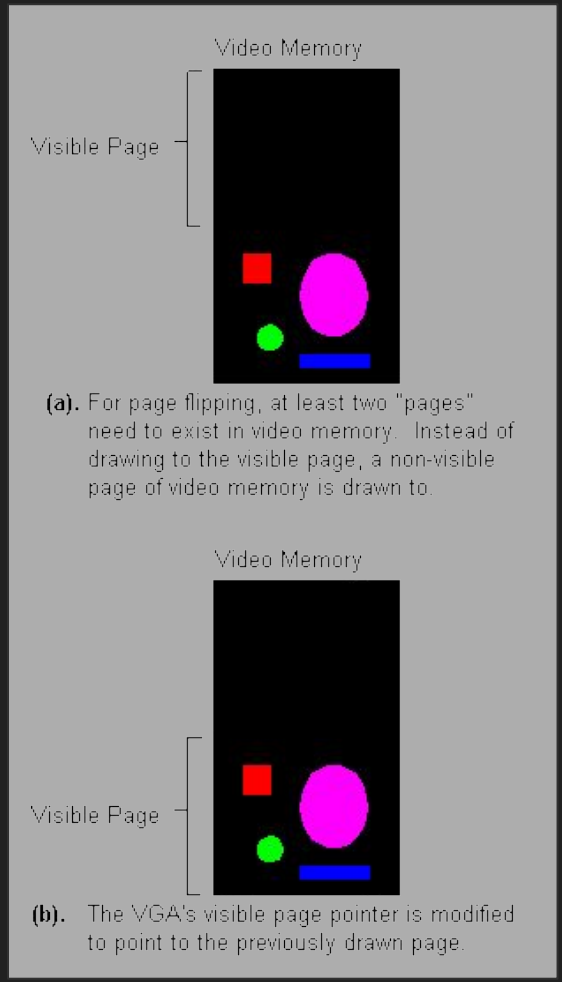
VGA Optimizations

- Double buffering
  - Game updates double buffer
  - memcpy buffer to VGA
- Unchained mode
  - Use all 256k of memory
  - Harder to address pixels



Page Flipping

- Use 2 64k pages
- Write to non-visible page
- Switch the displayed page



Keyboard Interrupts

- Call wakeup in console interrupt
- Shared keystate
    - Dangerous in general, but not for this application
- “Bottom half” handles wakeup

Tetris

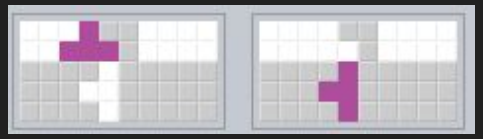
- 7 tetrominoes with 4 rotations
- Pieces spawn at top and fall
- Lock once they touch bottom or block
- Full rows are cleared and give points

Implementation

- tetris.c contains user space program
- game loop is in sys_tetris syscall
- logic code in systetris.h and systetris.c
- display code in display.h display.c
- keyhandling in console.c, proc.h, proc.c

Special Rules

- Lock delay (Infinity)
- Wall kicks
- T-spins and teleporting



T-spin

J-twist

Teleport

Summary

- We can draw to VGA in 2 modes
- We can handle keyboard interrupts outside interrupt context
- We can combine these features to make a working game

Demo