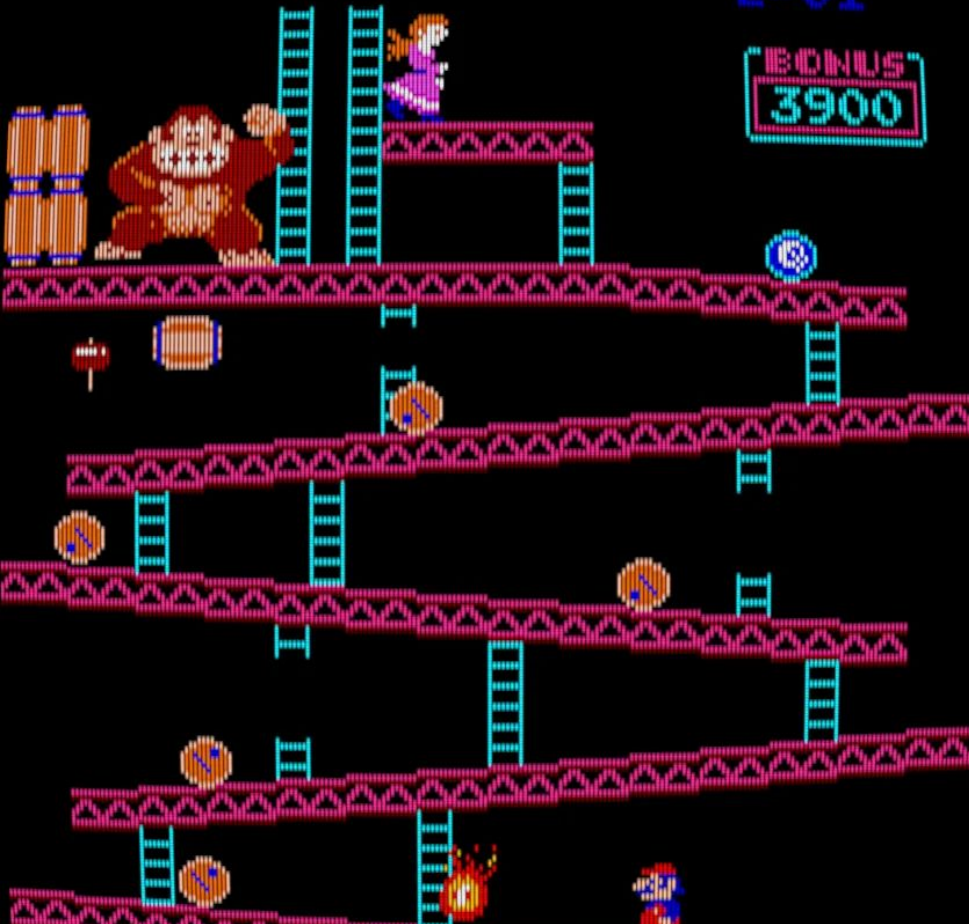


1UP  
000900

HIGH SCORE  
007650

L=01

BONUS  
3900



# Donkey Kong

Sean Stothers (sps2308),  
Ania Róża Krzyżańska (ark2219),  
Ines Khouider (ik2512)

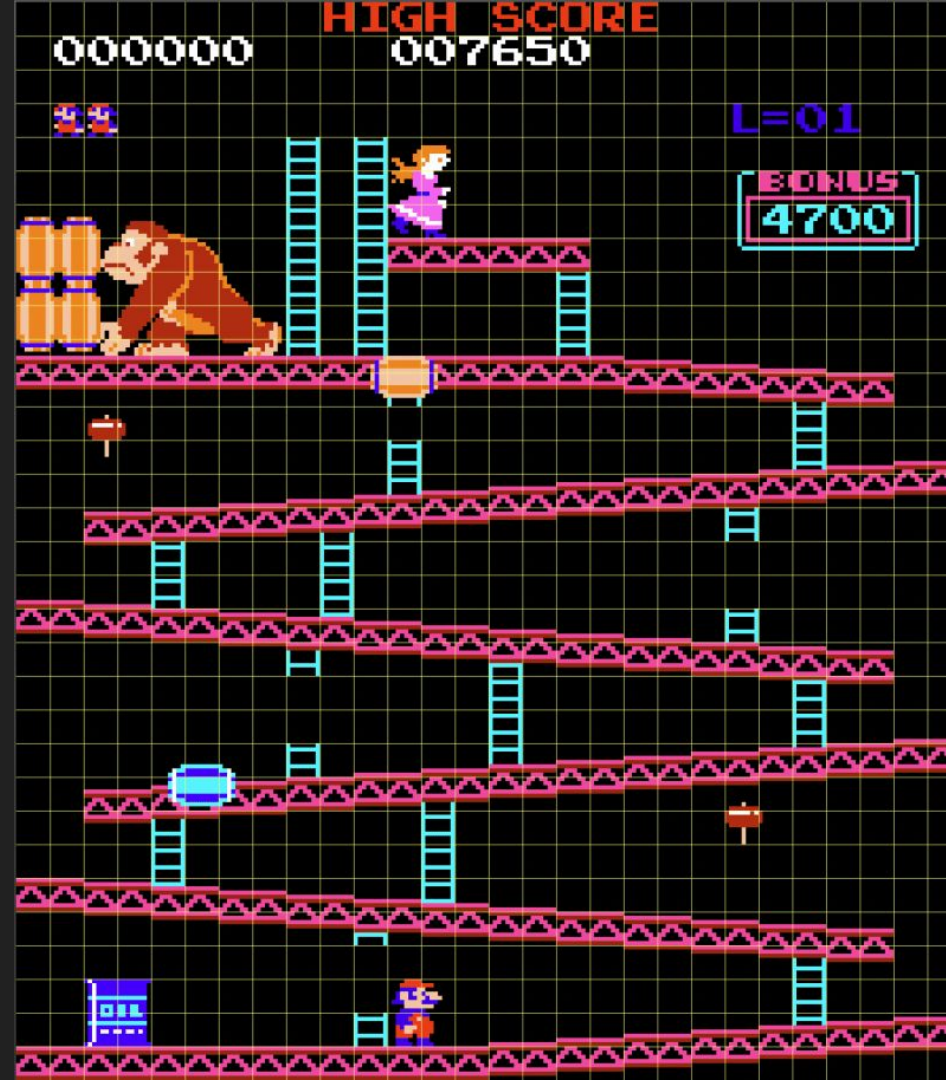


# Goal

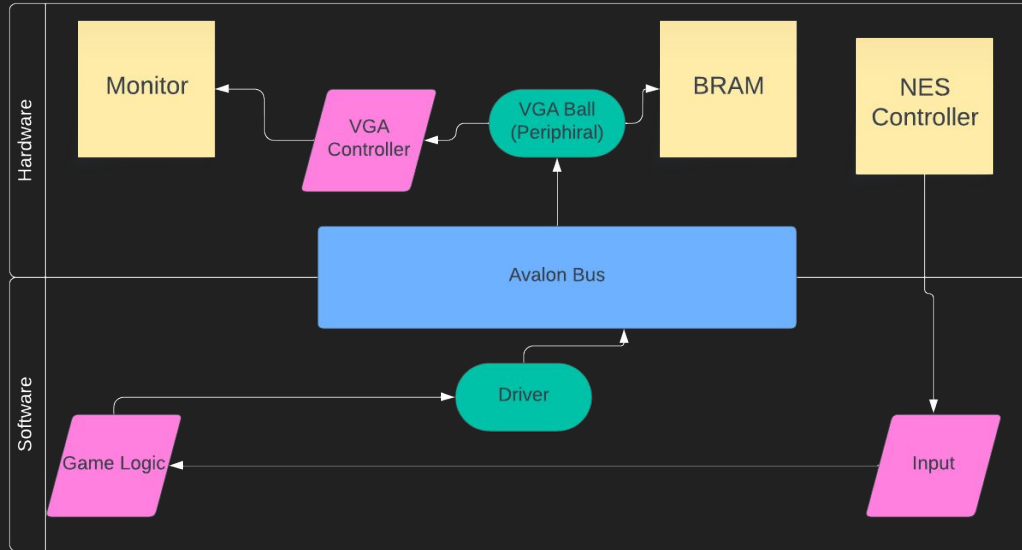
Make the arcade game Donkey Kong

Mario jumps over barrels

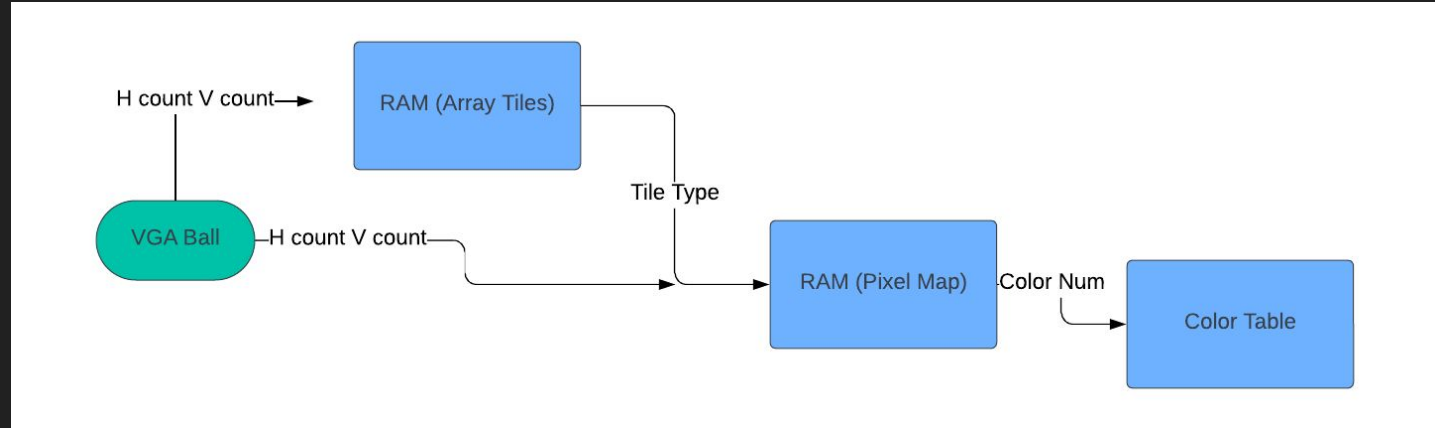
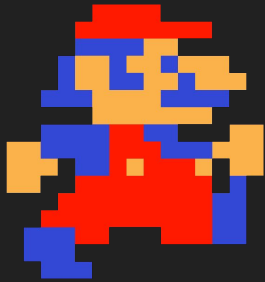
Mario dies when a barrel hits him



# System Block Diagram



# Background Tiles



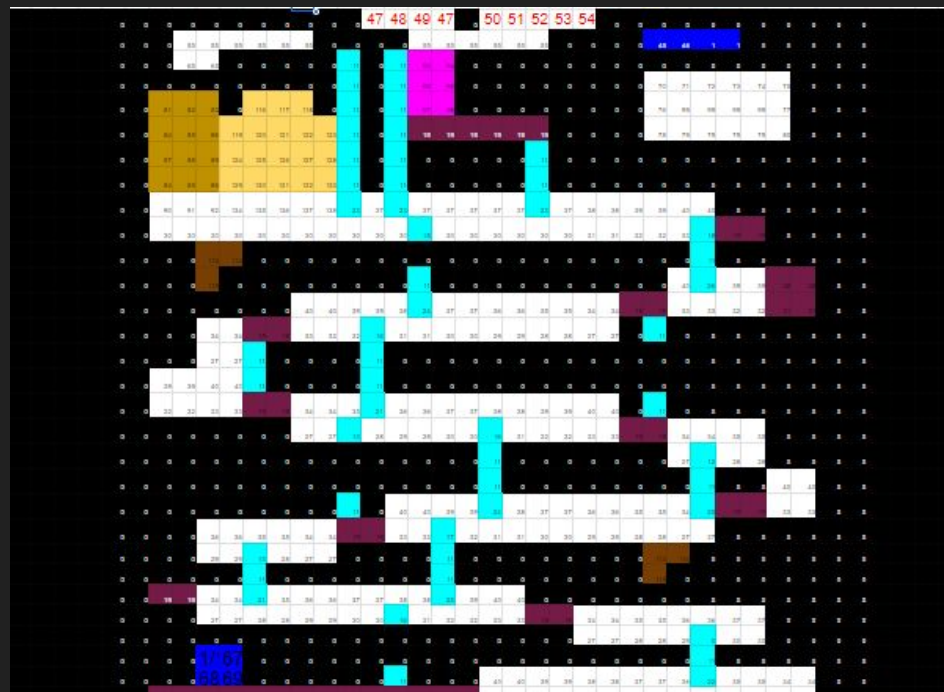
# Color Table And Tile Map

Array of 1024  
different addresses  
(32x32). Each  
address  
corresponds to an  
8x8 tile

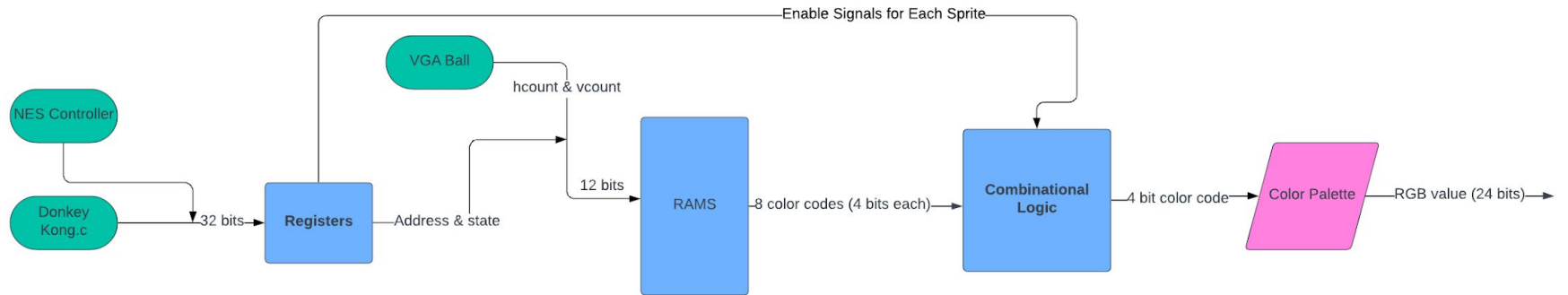


Color Table	
0	#000000
1	#FFFFFF
2	#FF2155
3	#FA0000
4	#00FBFF
5	#970000
6	#FF6800
7	#FFB855
8	#0000F8
9	#B70000
10 (added)	#FA4EF2

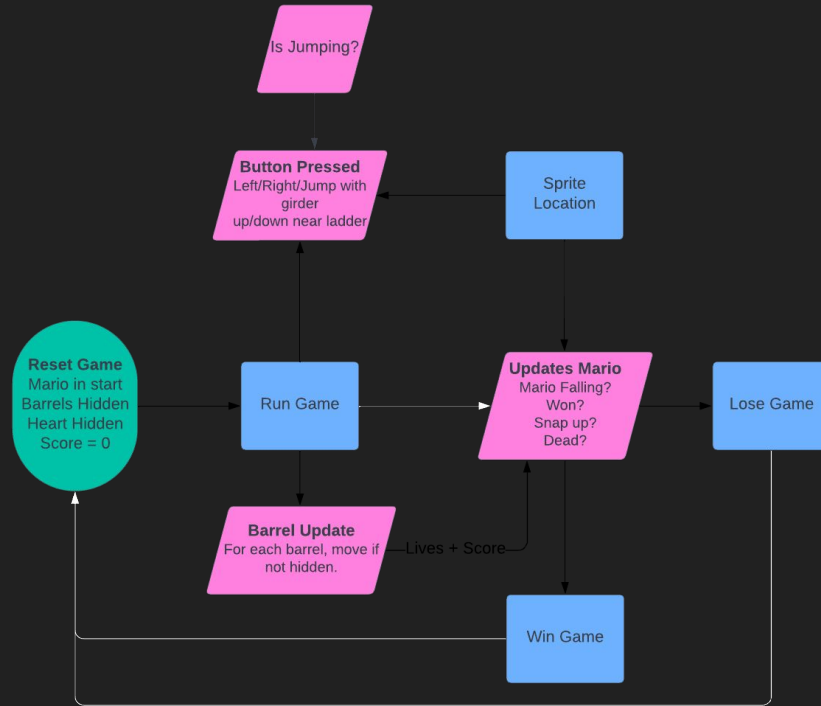
# Pixel Map



# Sprites

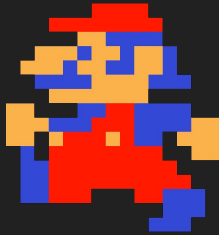


# Game Logic





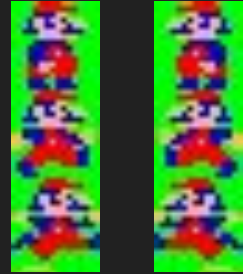
# Resource Budgets



Category	Size (bits)	Number of images	Total Size (bits)
Mario	32 x 32	16	$3 \times 32 \times 32 \times 16 = 49152$
Donkey Kong	50x50	1	$3 \times 50 \times 50 \times 1 = 7500$
Barrel	16x16	2	$3 \times 16 \times 16 \times 2 = 1536$
Ladders	20x30	9	$3 \times 20 \times 30 \times 9 = 16\ 200$
Platform	20x30	15	$3 \times 20 \times 30 \times 15 = 27000$
Hammer	10x10	1	$3 \times 10 \times 10 \times 1 = 300$
Total			101, 688 bits

# Hardware Software Interface

- Game displayed on computer monitor
- We used 10 32-bit registers
  - Registers for tile map and array & mario coordinates
  - Separate register for each barrel
  - Joint register for sprite states
- NES controller protocol:



Direction States

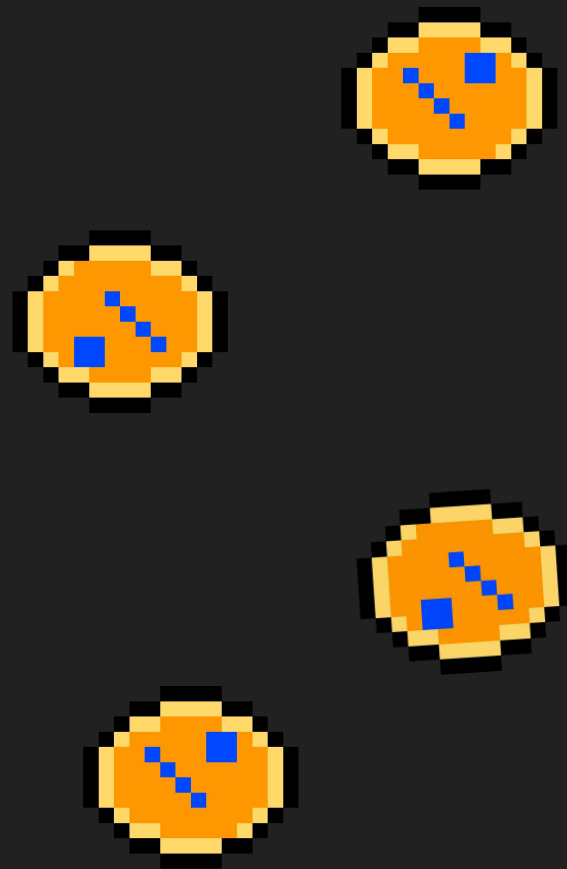
```
Nothing:      01 7f 7f 7f 7f 0f 00 00
Left:         01 7f 7f 00 7f 0f 00 00
Right:        01 7f 7f ff 7f 0f 00 00
```



# Registers

	0	1	2	3	4	5	6	7
0	Sprite X							
1	Sprite X							
2	Sprite Y							
3	Sprite Y							

	0	1	2	3	4	5	6	7			
0	L/R Mario		Which Mario			Which b0		b0 flip?			
1	Which b1		b1 flip?		Which b2		b2 flip?		Which b3		
2	b3 Flip?		Which b4		b4 flip?		which b5		b4 flip?		b0 on
3	b1 on		b2 on		b3 on		b4 on		b5 on		



# Challenges

- Consistent hidden barrels → created bitmask

```
st.state = (sprite.tag << 5) + (sprite.direction<<7) + (st.state & 4294967071); //this is the value of the bitmask with 32 1s and 0s at bits 5-7
```

- Lining up mario to the slanted Girder → modified tilemap



Tiles	
0	Falling
11	Ladder
12-13	Ladder Bottom
14-15	Ladder Top
19	Girder slaight no offset
20	Girder offset -8
40s	Negative slope ladder top
50s	Positive slope ladder top
60s	Negative slope ladder bottoms
70s	Positive slope ladder bottoms
80s	Negative slope
90s	Positive slope

Demonstration!

