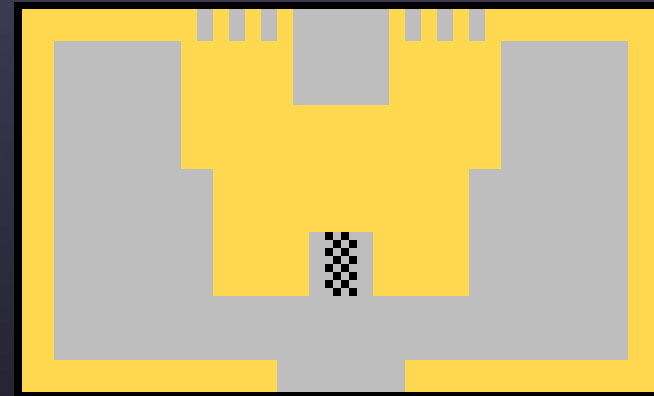


---

# Design of the First Action-Adventure Video Game: Adventure for the Atari 2600

Warren Robinett



University of North Carolina at Chapel Hill  
Department of Computer Science

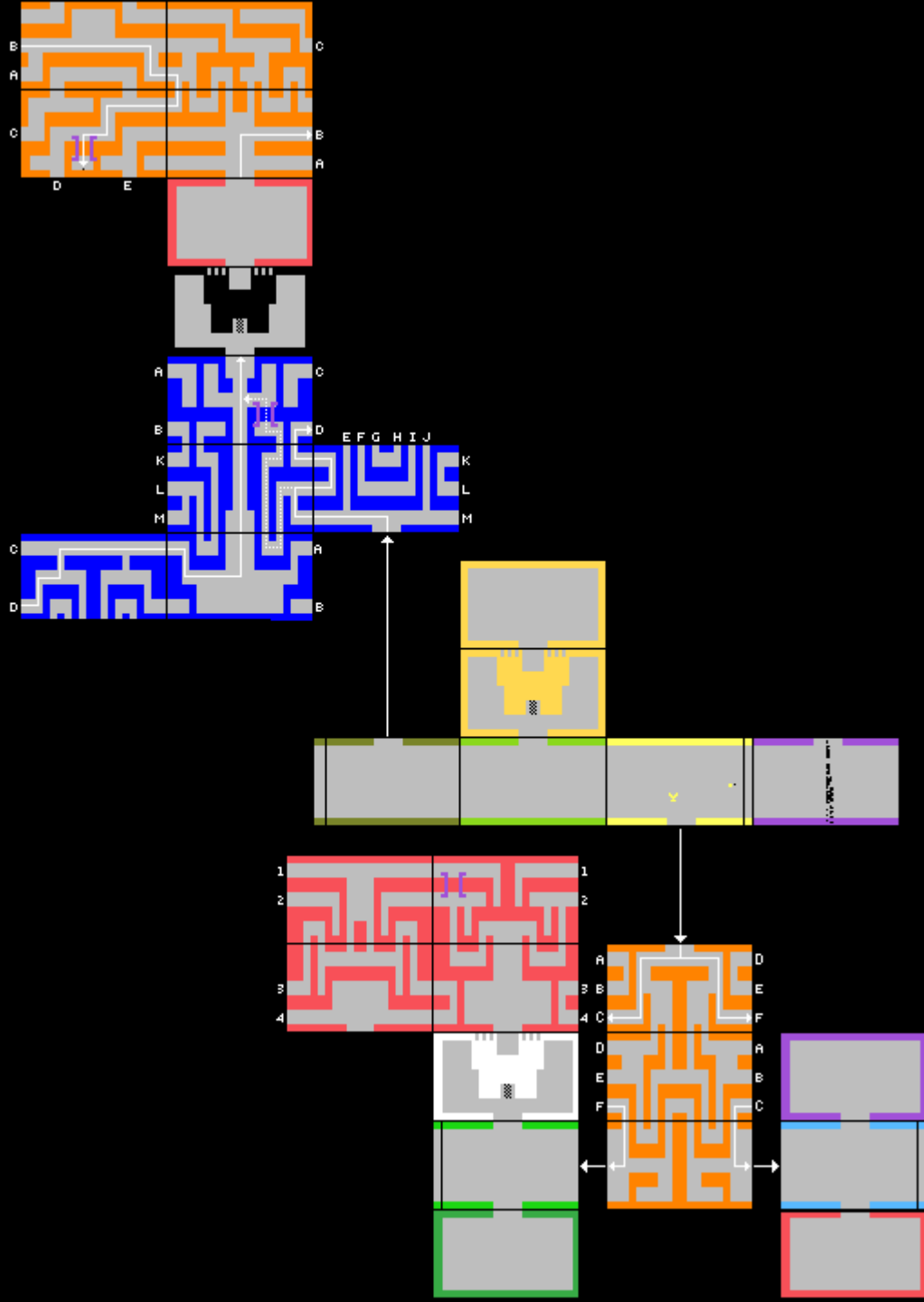
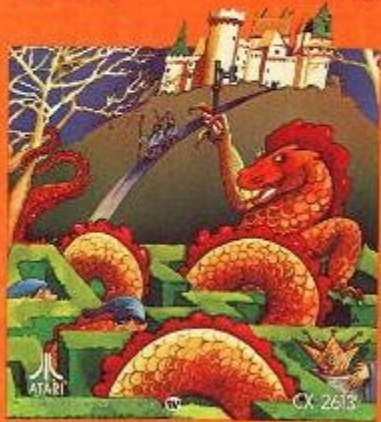
# ADVENTURE

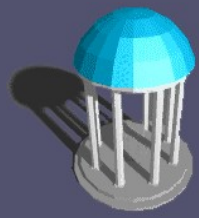
VIDEO COMPUTER SYSTEM™

GAME PROGRAM™

3 VIDEO GAMES

THREE SKILL LEVELS  
ONE PLAYER

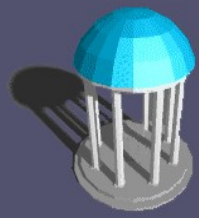




# Overview of Talk

---

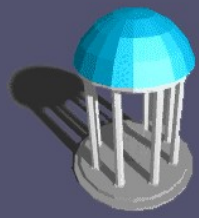
- Early History of Atari
- Architecture of the Atari 2600 Video Game
- Programming the Atari 2600:
  - How Video Games Were Made in the Late 1970's
- Design of Adventure



# History of Atari

---

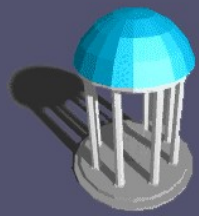
- Atari invented the video game
  - Pong, 1971, coin-op
- Atari 2600 hooked up to home TV, 1977
  - Games on ROM cartridges
  - 18 million consoles sold, 500 game carts developed
- Creation of game genres
  - shooters, driving, sports, adventure, side-scroller



# Atari 2600 Specs

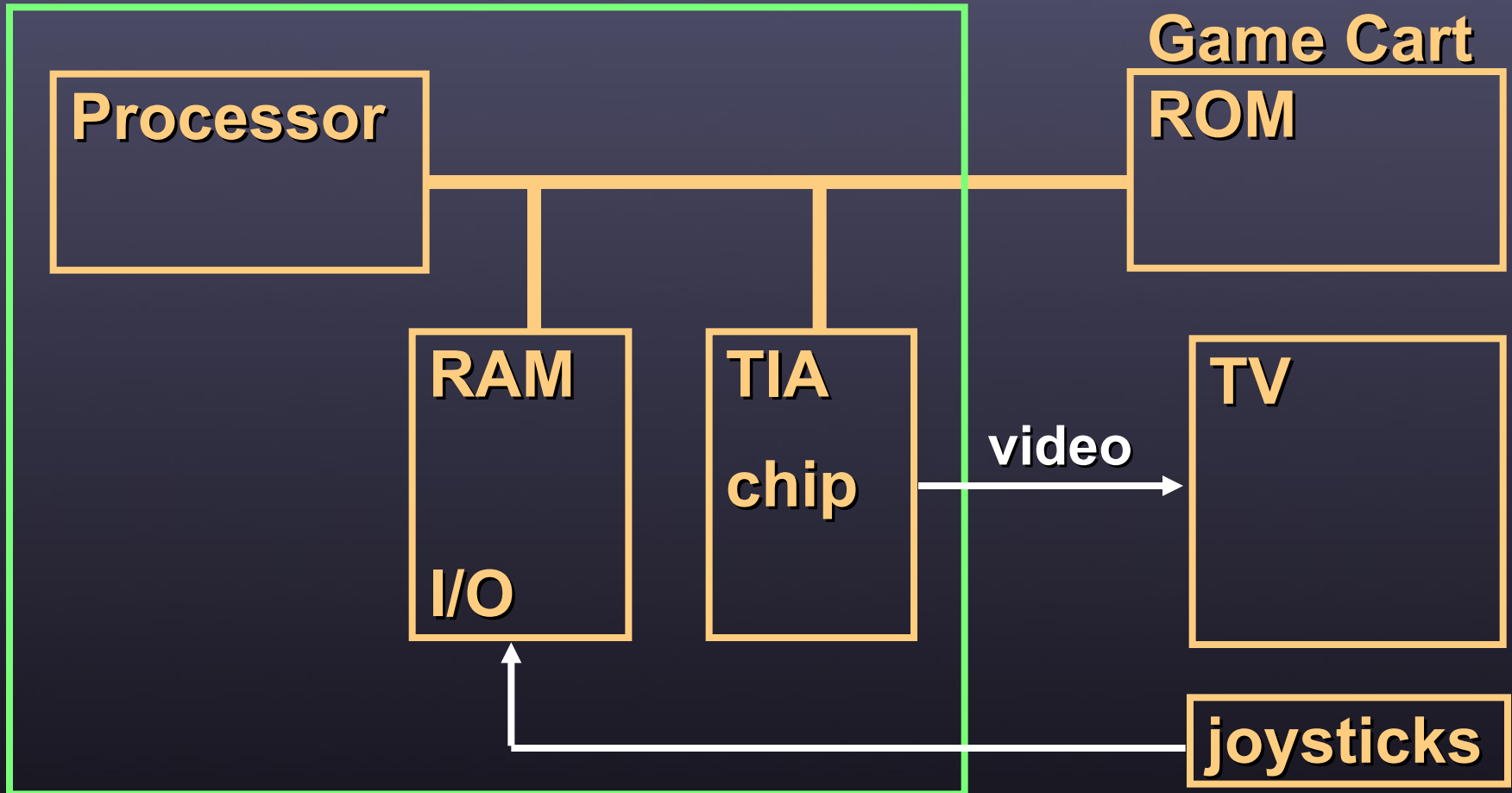
---

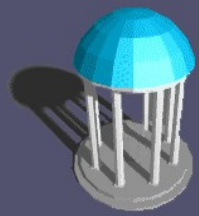
- 4K bytes of ROM
  - for game program and data
- 128 bytes of RAM
  - for game variables and stack
- 1.2 MHz 8-bit processor (6502)
- Custom chip (TIA)
  - generated video signal to drive TV
  - “Television Interface Adaptor” -- codename Stella



# Atari 2600 System Diagram

## Atari 2600 Console

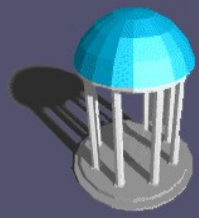




# Atari 2600 -- The Bad News

---

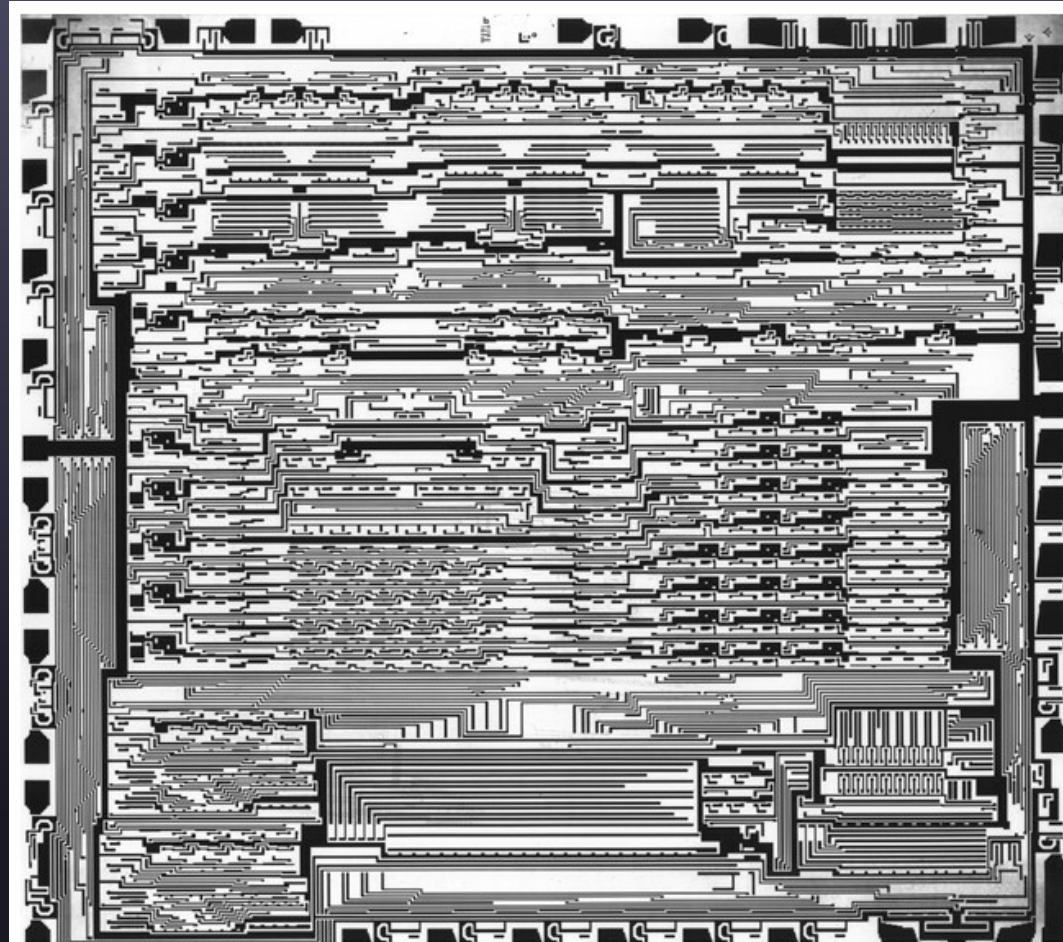
- No frame buffer
- No cache
- No floating point math
- No multiply or divide instructions
  - Adding two 8-bit integers: 10  $\mu$ sec
- Must generate a new video frame 60 times per second
  - Using 5000 machine instructions per frame

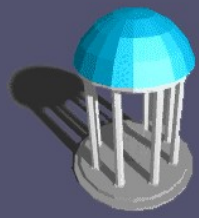


# Atari 2600 -- The Good News

---

- TIA chip -- hardware support for video games
  - Sprites
  - Collision detection
  - Audio synthesis
  - Interval timer

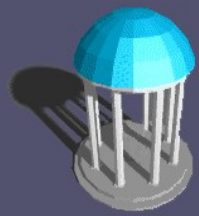




# Memory Map

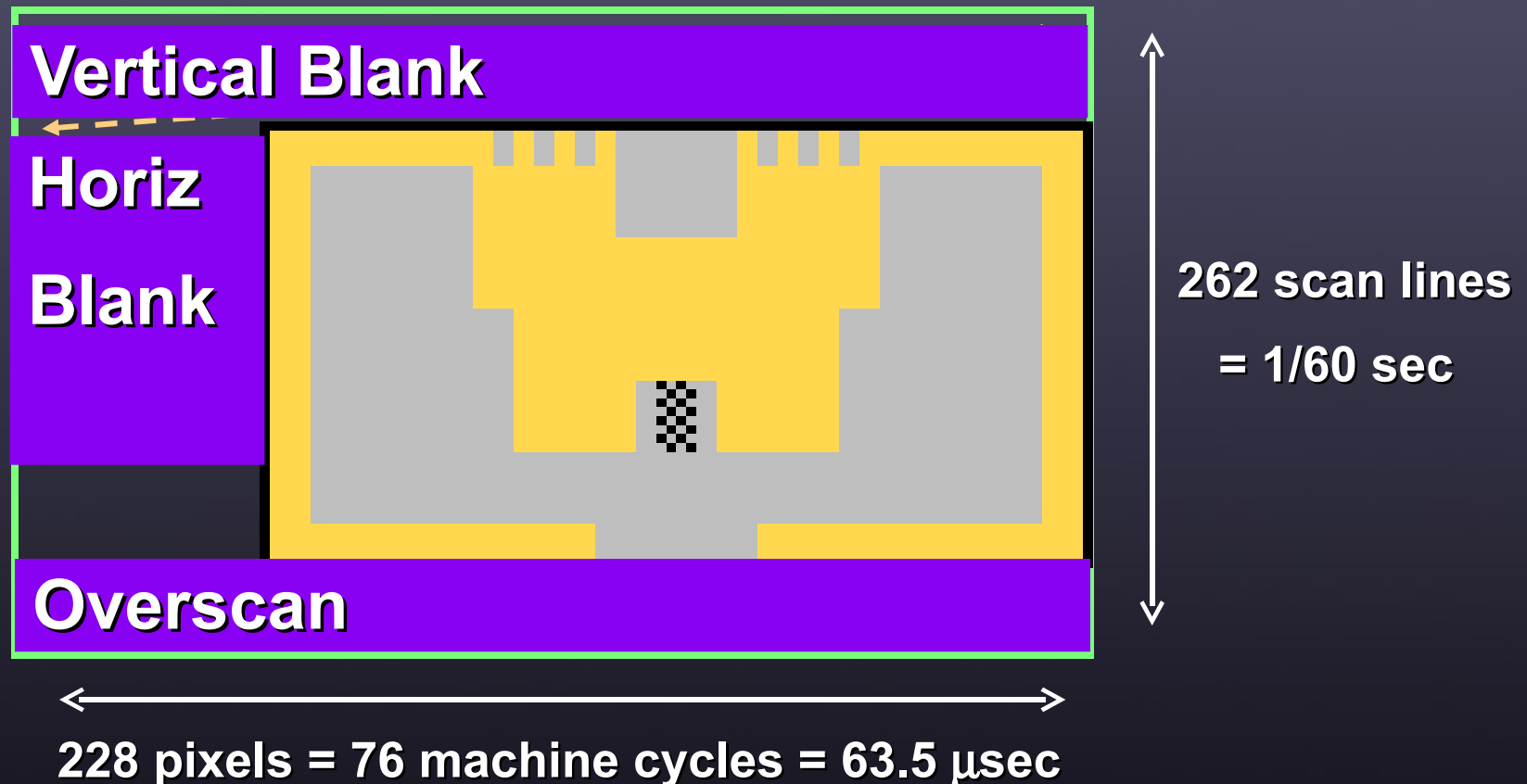
---

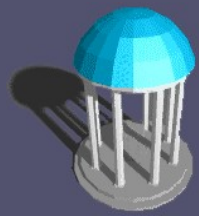
- 00 - 2C: TIA chip registers
- 80 - FF: RAM
- 280 - 281: Joystick port
- 282 - 283: Console switches
- 284 - 297: Interval timer
- F000 - FFFF: Game cartridge ROM



# Timing of Video Frame

## Video Frame

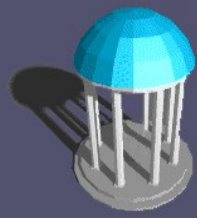




# Frame Timing

---

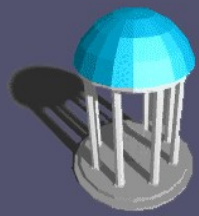
- Kernel (display loop synced with scan beam)
  - 192 scan lines Display
- Overscan: 37 scan lines Game play
- Vertical Sync: 3 scan lines
- Vertical Blank: 30 scan lines Game play



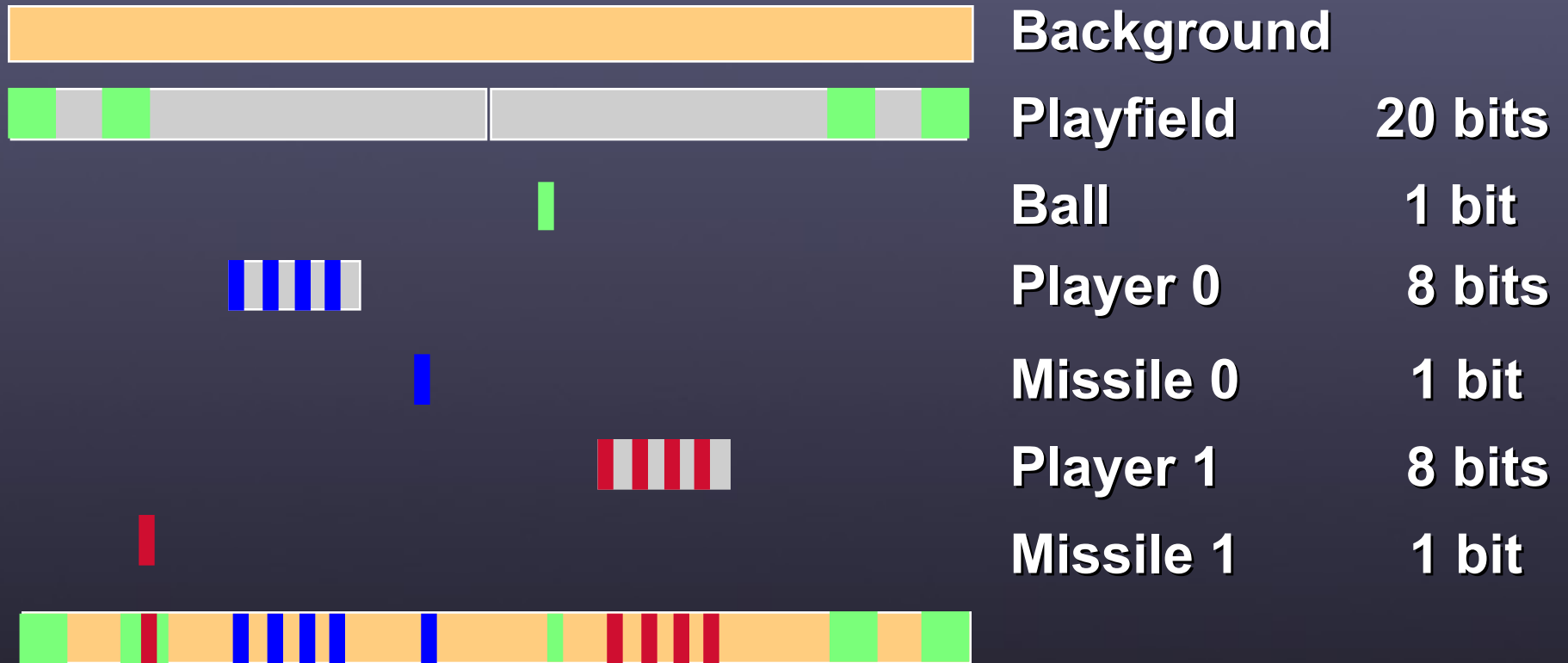
## Q: No Frame Buffer? How do you display anything?

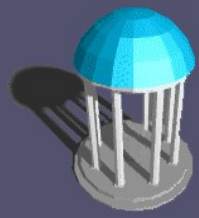
---

- A: There is a buffer that can hold one scan-line
  - The processor keeps changing what's in it as the scan beam moves down the screen.
- Unless the processor changes something, it generates the same line again and again
  - Vertical stripes when the processor crashes.



# TIA chip's Scan-line Buffer: 6 "bit-planes" + background





# Things sprites could do

---

- Sprite (P0)

- Move horizontally



- Change graphics



- Change color

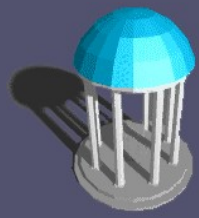


- Change width



- Replicate

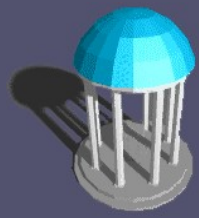




# Kernel Programming

---

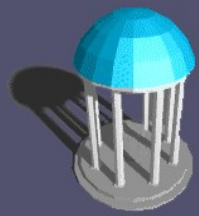
- Kernel loop synchronized with scan beam.
  - 76 machine cycles per scan-line.
- Two-line kernels were usual.
  - $2 \times 76 = 152$  cycles for the loop
- If it took one too many cycles, timing failed and the TV screen rolled.



# Cycle Counting

---

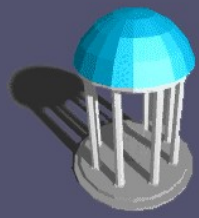
- 3 LDA LINE\_COUNT
- 2 SEC
- 3 SBC P1\_Y\_COORD
- 2 BPL \*+10
- 3 LDY P1\_INDEX
- 4 LDA (P1\_PTR),Y
- 3 STA GRP1
- 2 BEQ \*+2
- 5 INC P1\_INDEX
  
- 27 cycles total



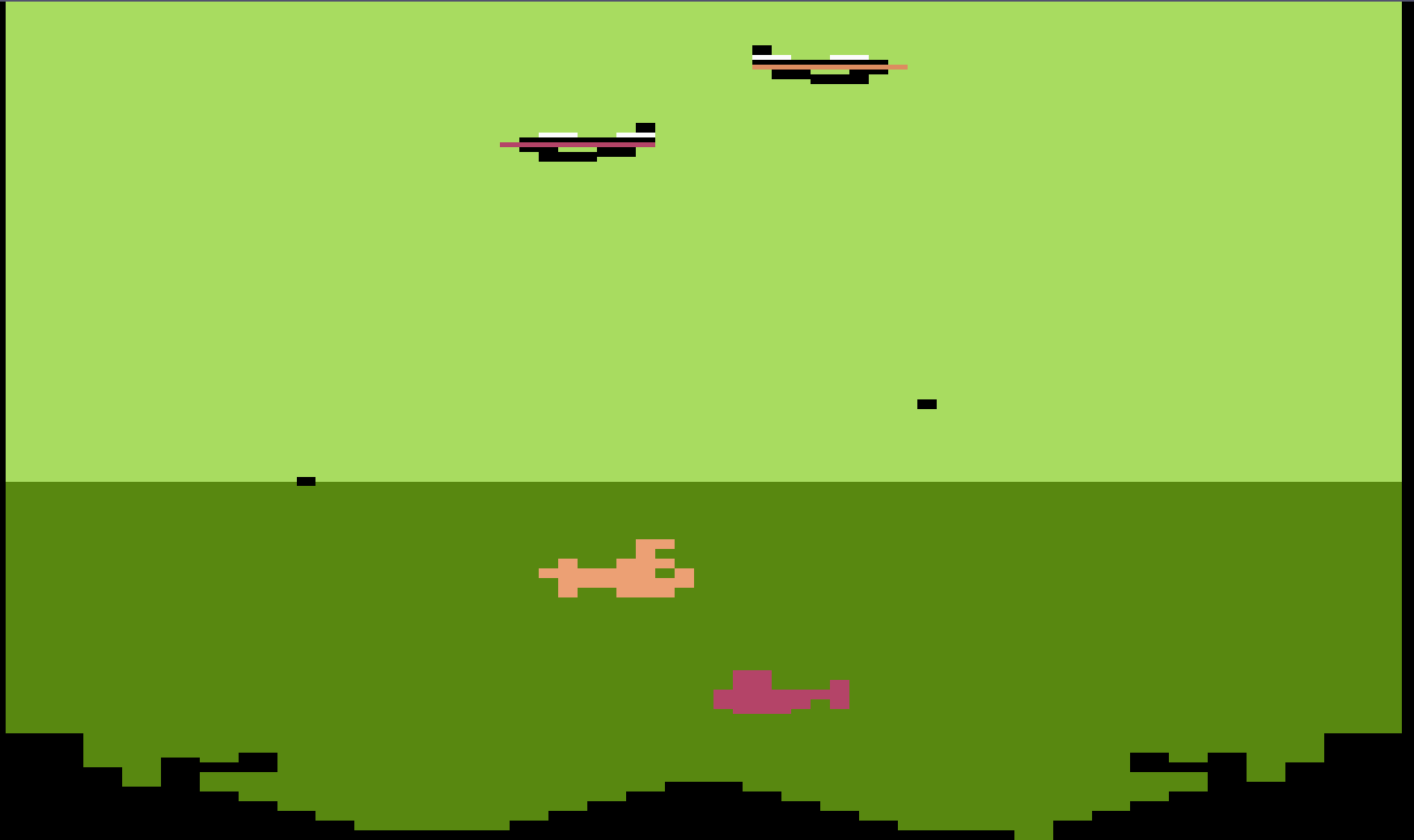
# Screen Shots

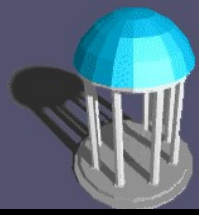
---

- You could do a lot with this hardware.
- Every cart had a different kernel.



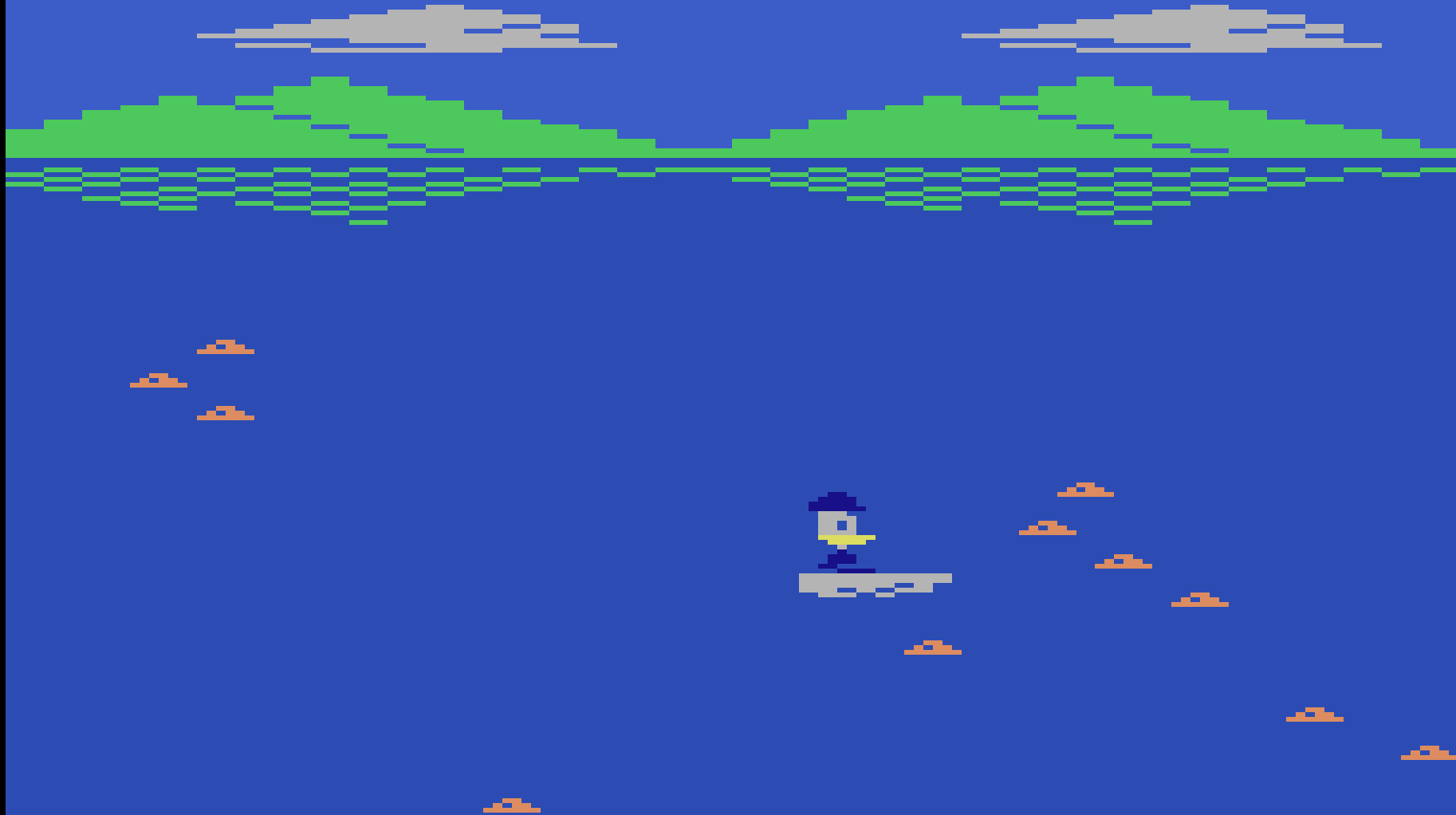
# Screen Shots

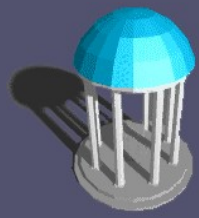




# Screen Shots

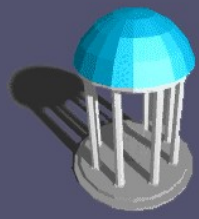
232360



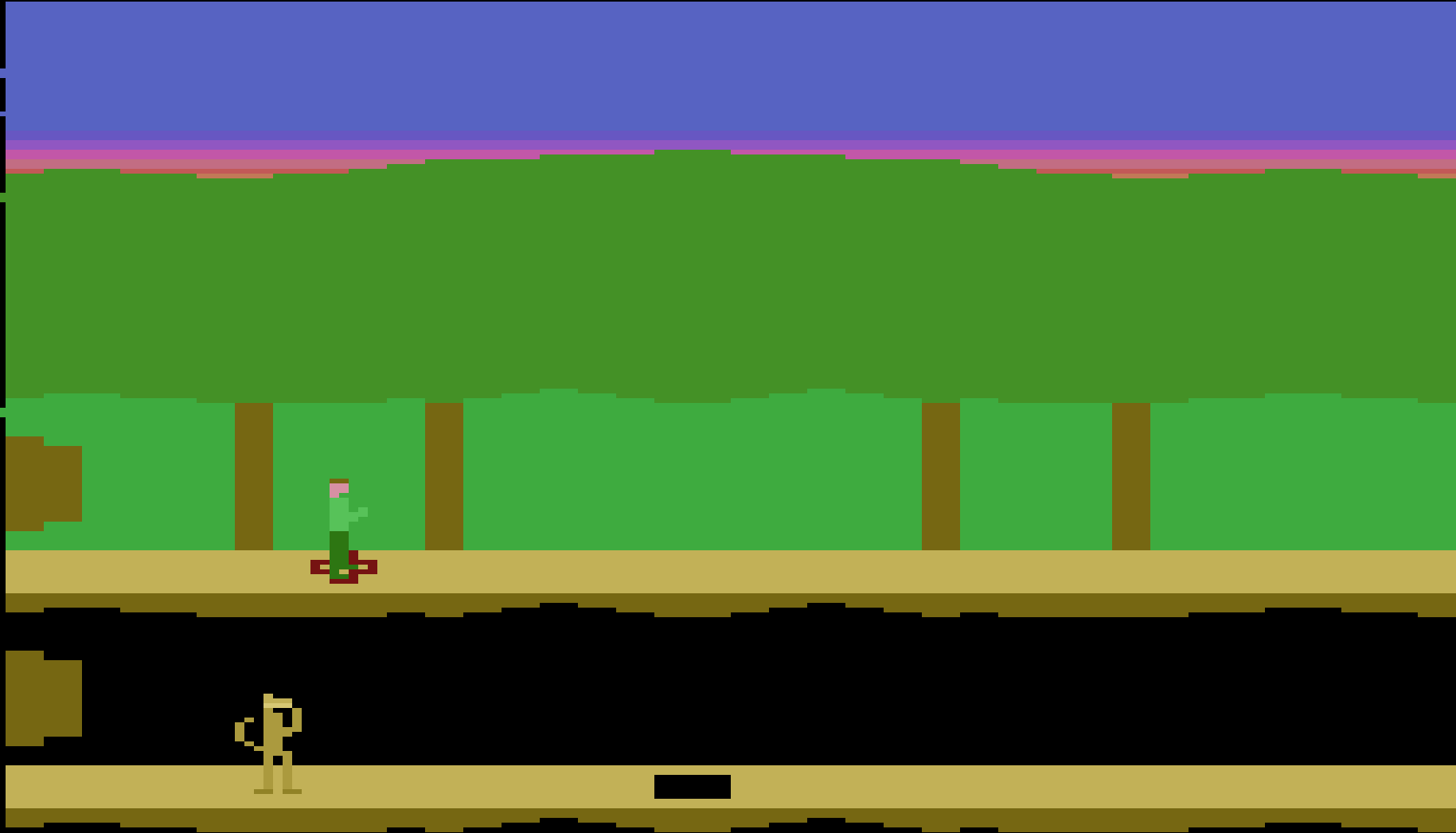


# Screen Shots

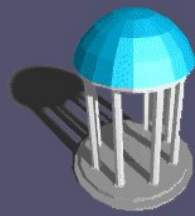




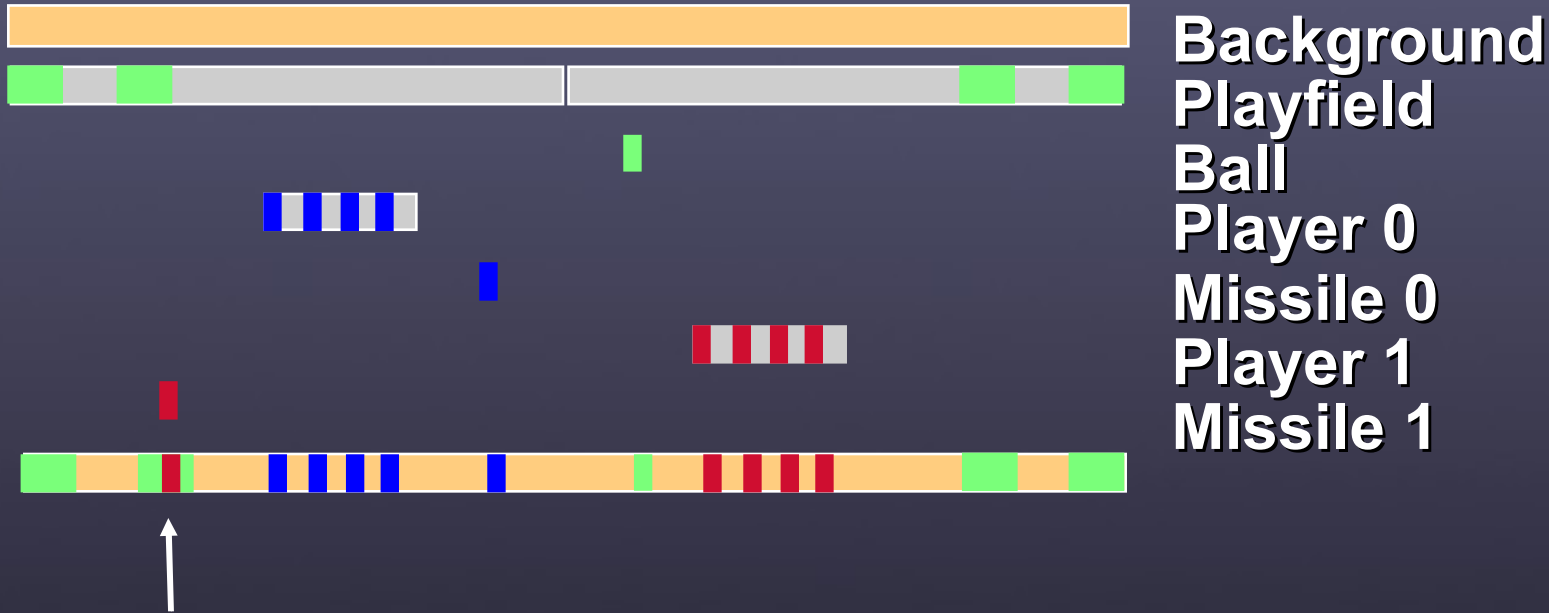
# Screen Shots



Copyright 1983

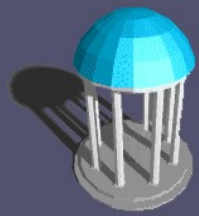


# Collision Detection: Detects Video Overlap



Collision when pair of "On" pixels overlap

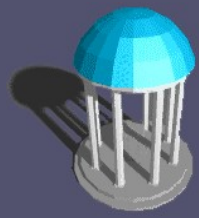
- 6 graphic objects (5 sprites + PF)
  - 15 possible pairs for collisions
  - 15 1-bit collision registers



# Sound Generators

---

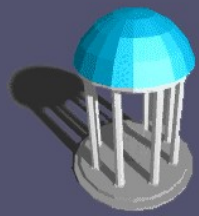
- 2 independent sound generators
  - 1-bit wide, 30 kHz sound signal
  - 4 bit volume control (16 volume levels)
- Generates psuedo-random sequences
  - 4-bit sound type register (16 sound timbres)
  - 5-bit frequency register (32 pitches)
- Processor could change sound parameters



# Design of Atari 2600 Adventure

---

- Directly inspired by the original text adventure by Willie Crowther and Don Woods.
- Main problems:
  - How to do in a 4K video game program what took hundreds of K on a mainframe computer?
  - How to adapt the adventure game idea to the video game medium?
  - My boss told me it was impossible, and not to work on it.



# Crowther & Woods Text Adventure

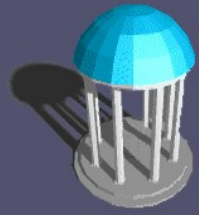
---

Welcome to Adventure!! Would you like instructions?

**YES**

Somewhere nearby is Collosal Cave, where others have found fortunes in treasure and gold, though it is rumored that some who enter are never seen again. Magic is said to work in the cave. I will be your eyes and hands. Direct me with commands of 1 or 2 words.

You are standing at the end of a road before a small brick building. Around you is a forest. A small stream flows out of the building and down a gully.



# Picking up Objects

---

## ENTER BUILDING

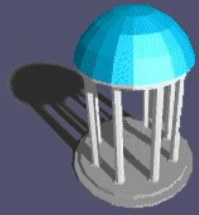
You are inside a building, a well house for a large spring.  
There are some keys on the ground here.  
There is a shiny brass lamp nearby.

## TAKE KEYS

OK

## TAKE LAMP

OK



# Carrying Objects

---

## LEAVE BUILDING

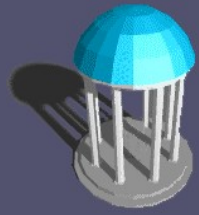
You're in front of building.

## INVENTORY

You are currently holding the following:

Set of keys

Brass lantern



# Moving from Place to Place

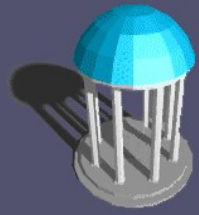
---

## GO SOUTH

You are in a valley in the forest beside a stream tumbling along a rocky bed.

## GO SOUTH

At your feet, all the water splashes into a 2-inch slit in the rock. Downstream the streambed is bare rock.



# Obstacles

---

## GO SOUTH

You are in a 20-foot depression floored with bare dirt. Set into the dirt is a strong steel grate mounted in concrete. A dry streambed leads into the depression.

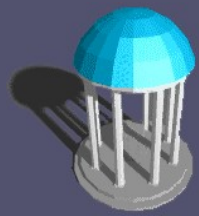
The grate is locked.

## GO DOWN

You can't go through a locked steel grate!

You're outside grate.

The grate is locked.



# Using Objects

---

## USE KEYS

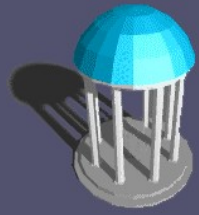
The grate is now unlocked.

## GO DOWN

You are in a small chamber beneath a 3x3 steel grate to the surface. A low crawl over cobbles leads inward to the west. The grate is open.

## GO WEST

You are crawling over cobbles in a low passage. There is a dim light at the east end of the passage. There is a small wicker cage discarded nearby.



# Another Obstacle

---

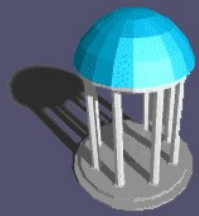
## GO WEST

It is now pitch dark. If you proceed you will likely fall into a pit.

## LIGHT LAMP

You are in a debris room filled with stuff washed in from the surface. A low wide passage with cobbles becomes plugged with mud and debris here, but an awkward canyon leads upward and west. A note on the wall says "MAGIC WORD XYZZY".

A three-foot black rod with a rusty star on an end lies nearby.



# Creatures

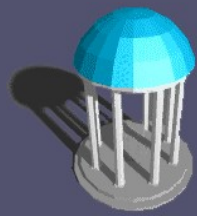
---

## GO WEST

You are in a splendid chamber thirty feet high. The walls are frozen rivers of orange stone. An awkward canyon and a good passage exit from east and west sides of the chamber.

There is a threatening little dwarf in the room with you!  
One sharp nasty knife is thrown at you!

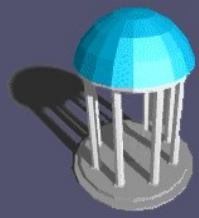
It misses!



# How to Adapt Adventure Game Idea to Video Game Format?

---

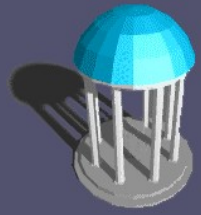
- How to display rooms?
  - One room on screen at a time, using PF graphics.
- How to display objects?
  - P0 and P1 sprites.
- How to display your own location?
  - Use the Ball sprite. Joystick moves you.
- How to move from room to room?
  - Move off the edge of screen to adjacent room.



# How to Adapt Adventure Game Idea to Video Game Format?

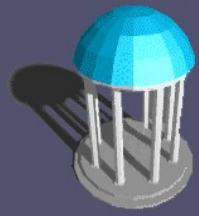
---

- How to pick up objects?
  - Video overlap of Self and Object sprites.
- How to drop objects?
  - Can carry only one object. Button on joystick drops it.
- How to use objects?
  - Video overlap of two objects.
- How to do creatures?
  - Object with subroutine to make it do things.

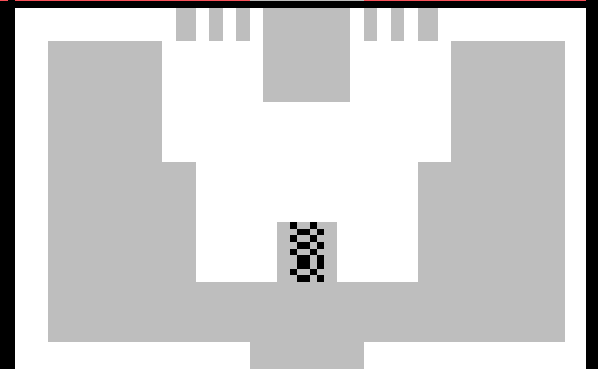
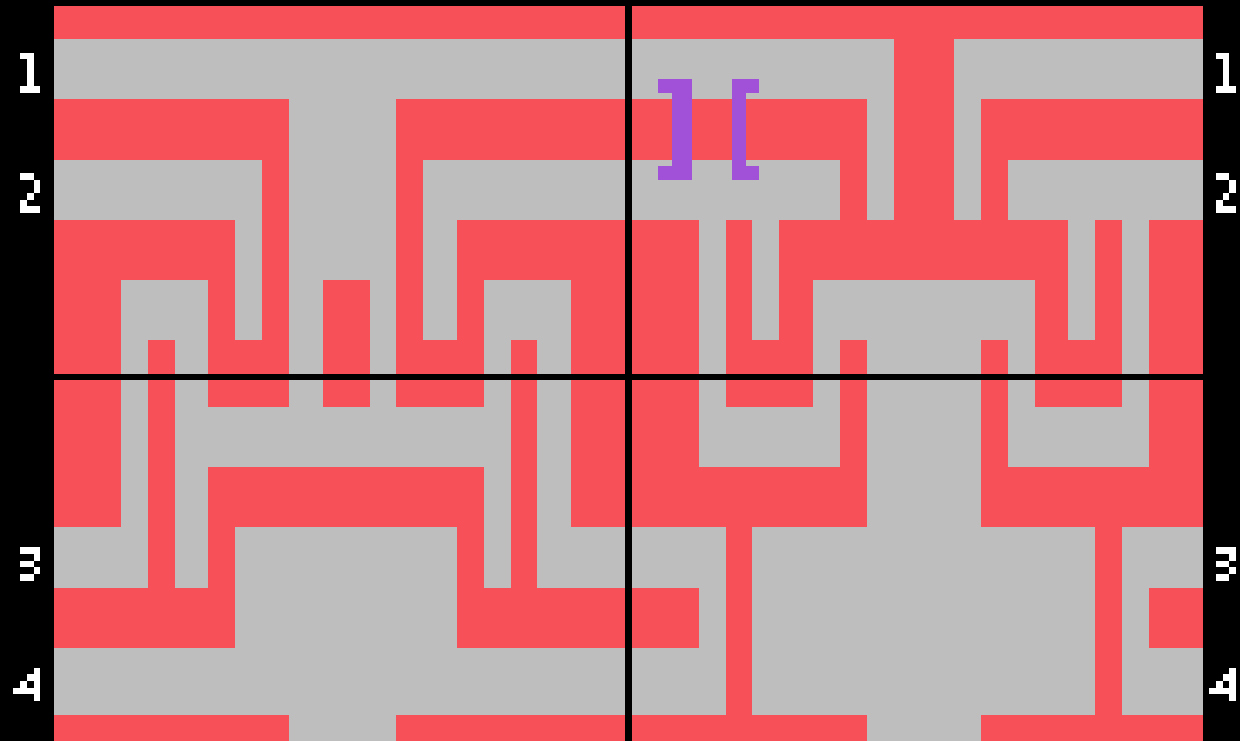


# Demo Adventure

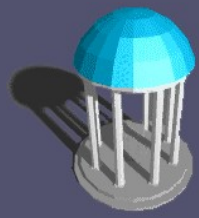
---



# Network of 30 Rooms



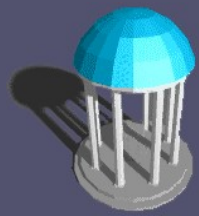




# Objects

---

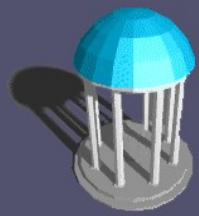
- 3 Keys: opened castle doors.
- Sword: killed dragons.
- Chalice: won the game.
- Magnet: attracts other objects.
- Bridge: crosses maze walls.
- Dot: gets you into secret room.



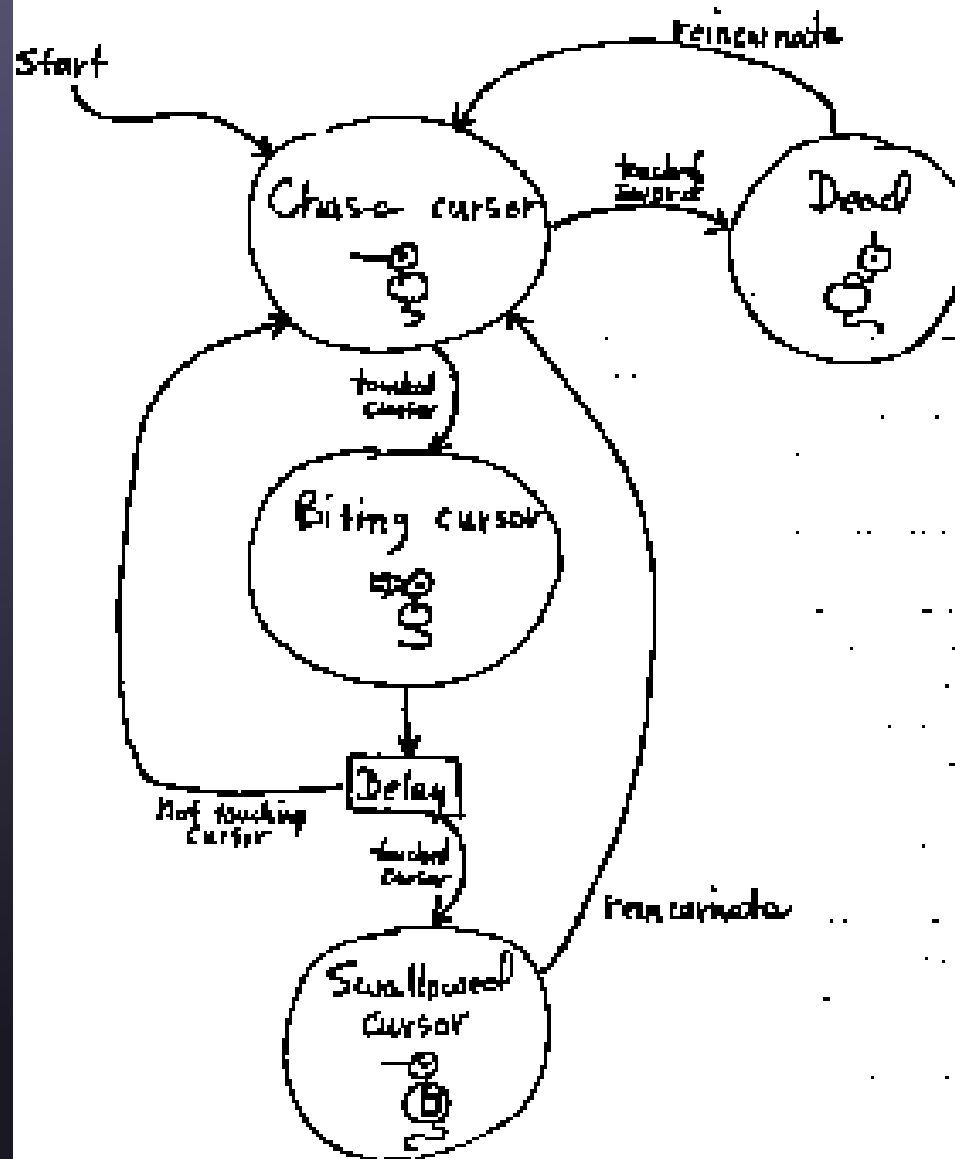
# Creatures

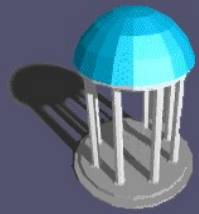
---

- 3 Dragons: chase you and eat you.
- Bat: steals what you're carrying, moves objects around
- Behaviorism
  - Each creature had a prioritized list of objects to go toward or away from. Responded to highest priority object in the same room with it.

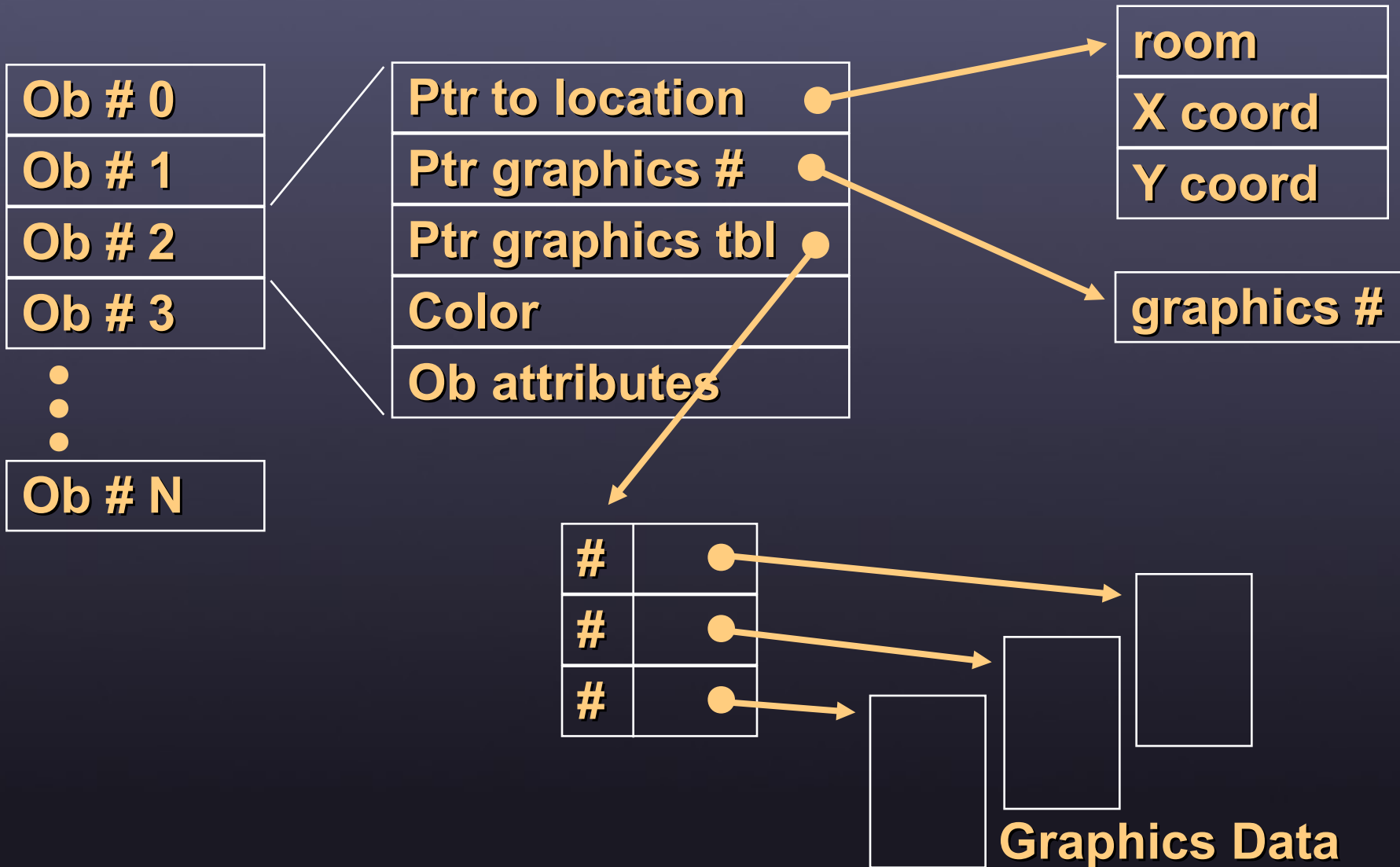


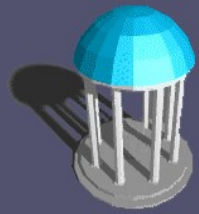
# State Diagram for Dragon





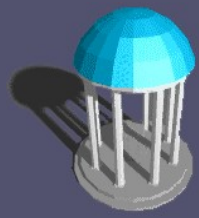
# Object List Data Structure





# Room List Data Structure

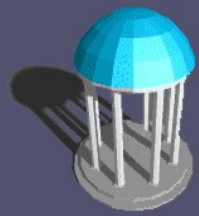




# Adventure RAM USage

---

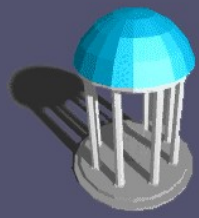
- Kernel variables: 18 bytes
  - World state: 49 bytes
  - Program variables: 34 bytes
  - Reserved for Stack: 14 bytes (max stack depth)
  - Unused 13 bytes
- **TOTAL 128 bytes of RAM**



# Adventure ROM Usage

---

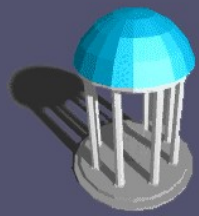
- Kernel and display code: 751 bytes
  - General adventure game code: 1081 bytes
  - Code for creatures and objects: 763 bytes
  - Sound code and data: 221 bytes
  - Data structures and graphics data: 1263 bytes
  - Unused: 17 bytes
- TOTAL ROM 4096 bytes



# Summary

---

- First action-adventure video game.
- Sold 1 million units (\$25 retail).
  - I got paid a salary (\$22K/year).



# Summary


---

- From the web:

## **Q: What are the best games for the 2600?**

Here are the results of the atari 2600 poll I conducted:

Best overall

1. Adventure 
2. (tie) Solaris, Yar's Revenge
4. Asteroids
5. River Raid