Gaming Through History

Tolulope Oshinowo, Dwaha Daud

https://github.com/toluooshy/GamingThroughHistory

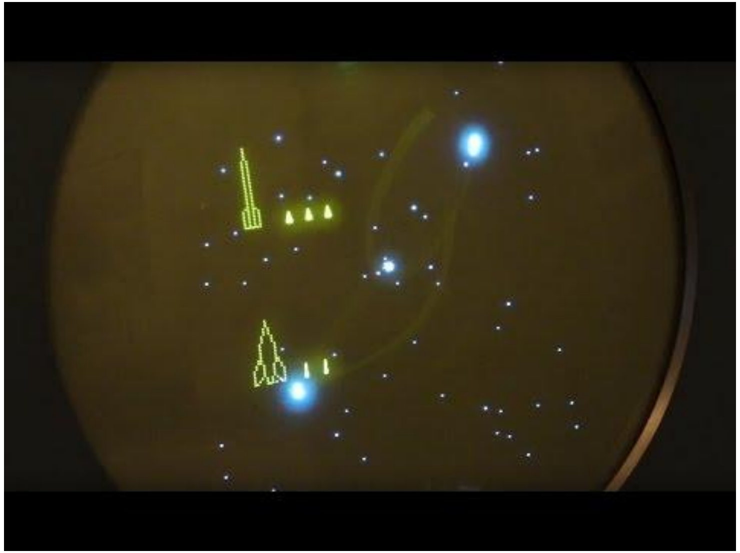
Project Overview

- Goal: to develop an understanding of how game development has evolved over time
- Use a single game, Pong, as a point of comparison for each time period
- Learned about different development models and improvements in design capabilities
- Theme: where possible, we designed the games as the famous “El Clásico” matchup between FC Barcelona and Real Madrid

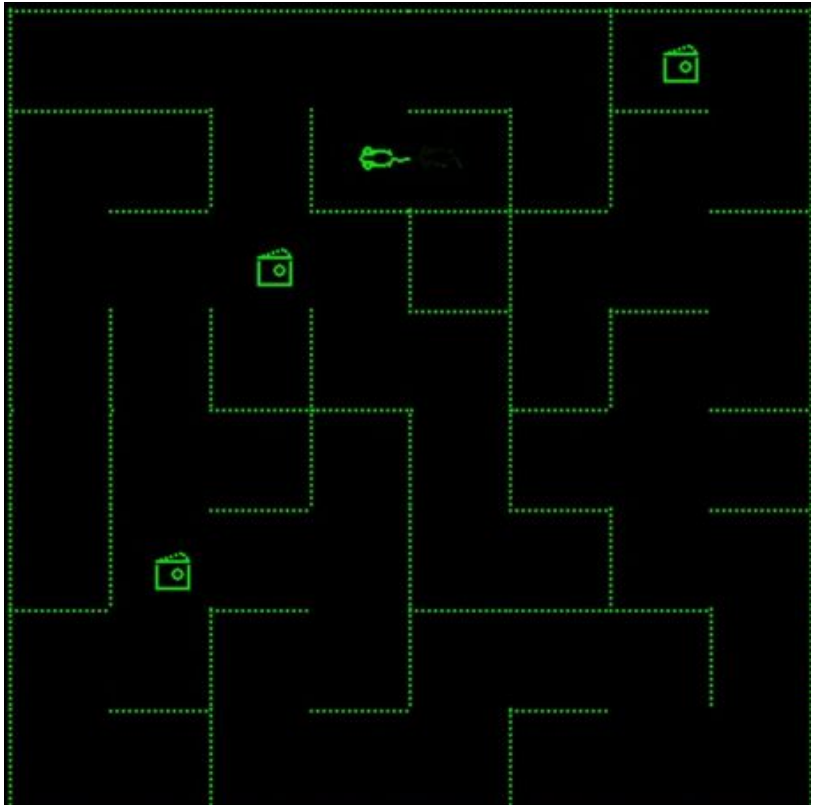
Programmed Data Processor-1 (PDP-1)

- The first computer in the PDP series by Digital Equipment Corporation. Produced in 1959
- Cost: $120,000. Weight: 730 kg
- Architecture:
  - 2700 transistors
  - Main memory: 18-bit word size, 4096 words (9 KB)
  - I/O: Punched tape reader and writer
- Graphics:
  - Cathode Ray Tube, point-plotting display
  - 1024 x 1024 addressable locations
  - CPU can send 20,000 points per second to the display
  - Can interact with the display via a light pen!

Example PDP-1 Games



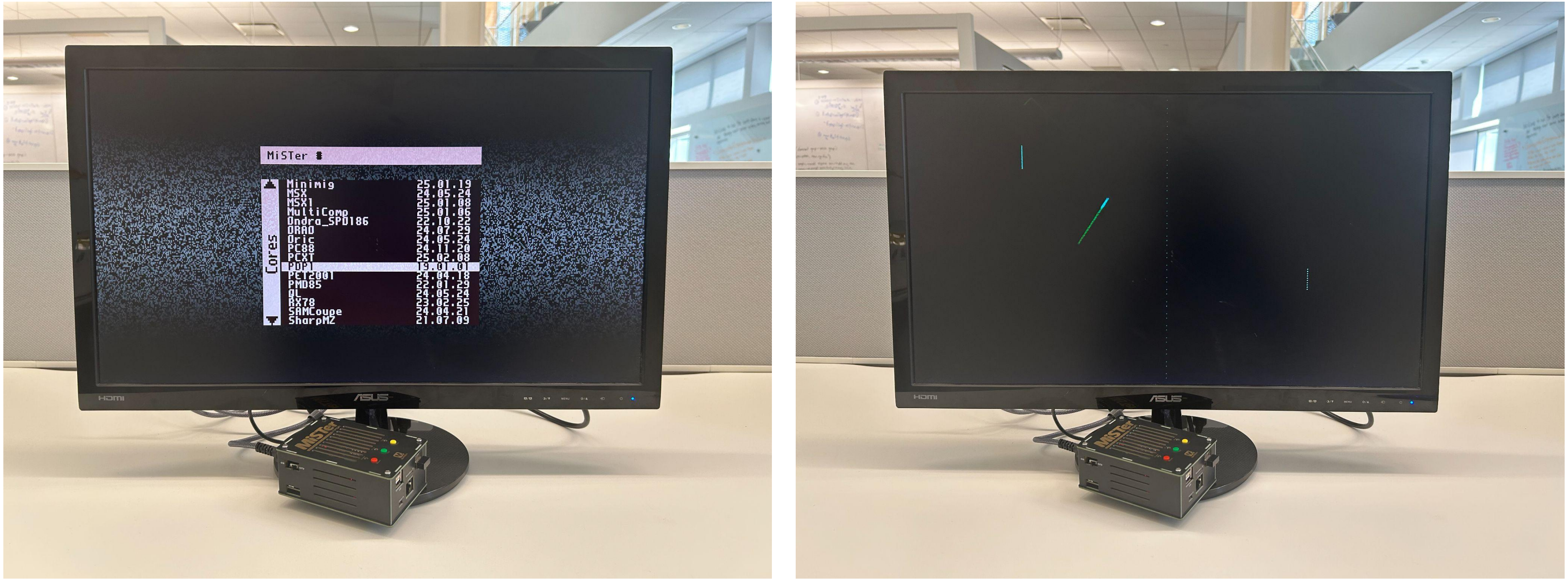
Spacewar (1962)



Mouse in a maze (1959)

El Clásico 1962 (PDP-1)

- Used the Mister FPGA to emulate a PDP-1 version of Pong
  - Mister is an open source project that aims to recreate classic computers using FPGA hardware
  - Mister provides hardware emulation to allows game images to run as they would have on past hardware
- Edited an existing implementation of Pong
  - Limited scope for graphics changes
- Gained an understanding of the programming model
  - Main game loop calls functions to manage game state (e.g check for collisions) and draw objects
  - Special instruction to read I/O from controller and update memory location with key-press state. Game loop functions read this location and move paddles



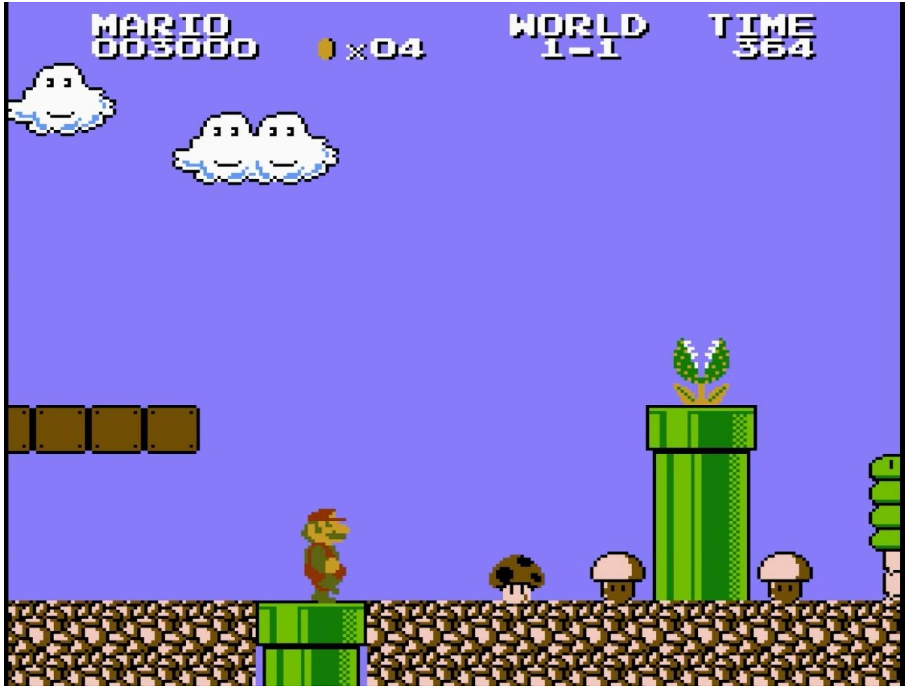
```
MiSTer
Cores
Minimig          25.01.19
MSX              24.05.24
MSX1             25.01.08
MultiComp        25.01.06
Ondra_SPO186     22.10.22
ORAO             24.07.29
Oric             24.05.24
PC88             24.11.20
PCXT             25.02.08
PDP1             19.01.01
PET2001          24.04.18
PMD85            22.01.29
QL               24.05.24
RX78             23.02.25
SAMCoupe         24.04.21
SharpMZ          21.07.09
```

Gameplay from “El Clásico 1962”

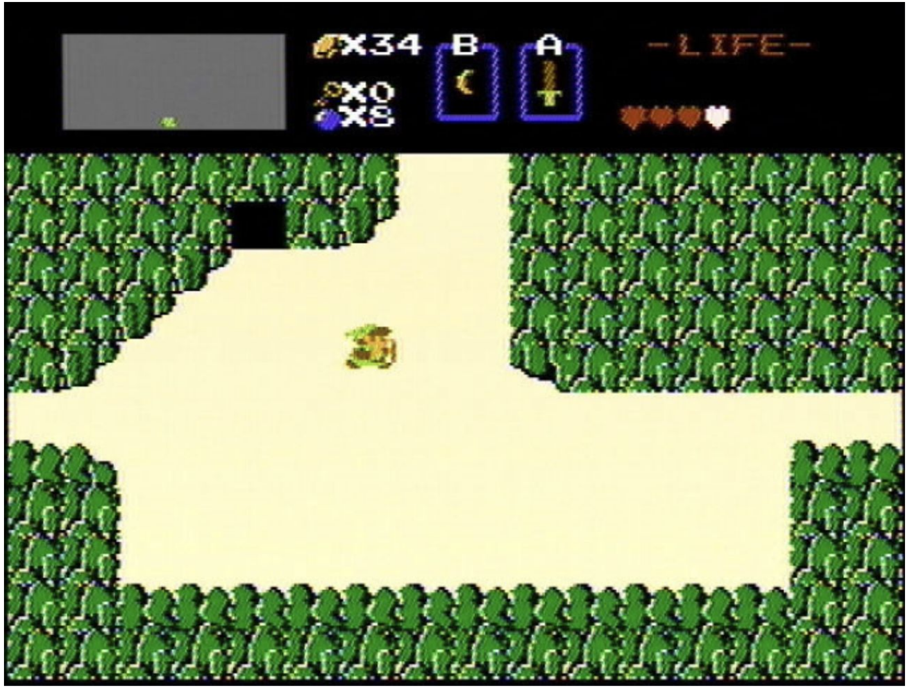
Nintendo Entertainment System (NES)

- Home video game console produced by Nintendo. First released in Japan in 1983
- Cost: $199
- Architecture
  - Custom CPU based on 8-bit 6502
  - Picture Processing Unit (PPU) chip handles graphics
  - I/O: Cartridges for game distribution. Controllers to read player input.
    - Cartridges include program ROM (game instructions for CPU to execute) and character ROM (graphics data for PPU to display)
- Graphics
  - PPU renders 2D graphics. Sprites and Backgrounds
  - Background 256 x 240 pixel display, organized into 8x8 bit “tiles” (32x30 tiles)
  - Srpties 256 bytes Object Attribute Memory in PPU, storing data for up to 64 sprites (4 bytes per sprite)
  - Graphics data (i.e tiles) and metadata (i.e tile coloring) pulled from game cartridge

Example NES games



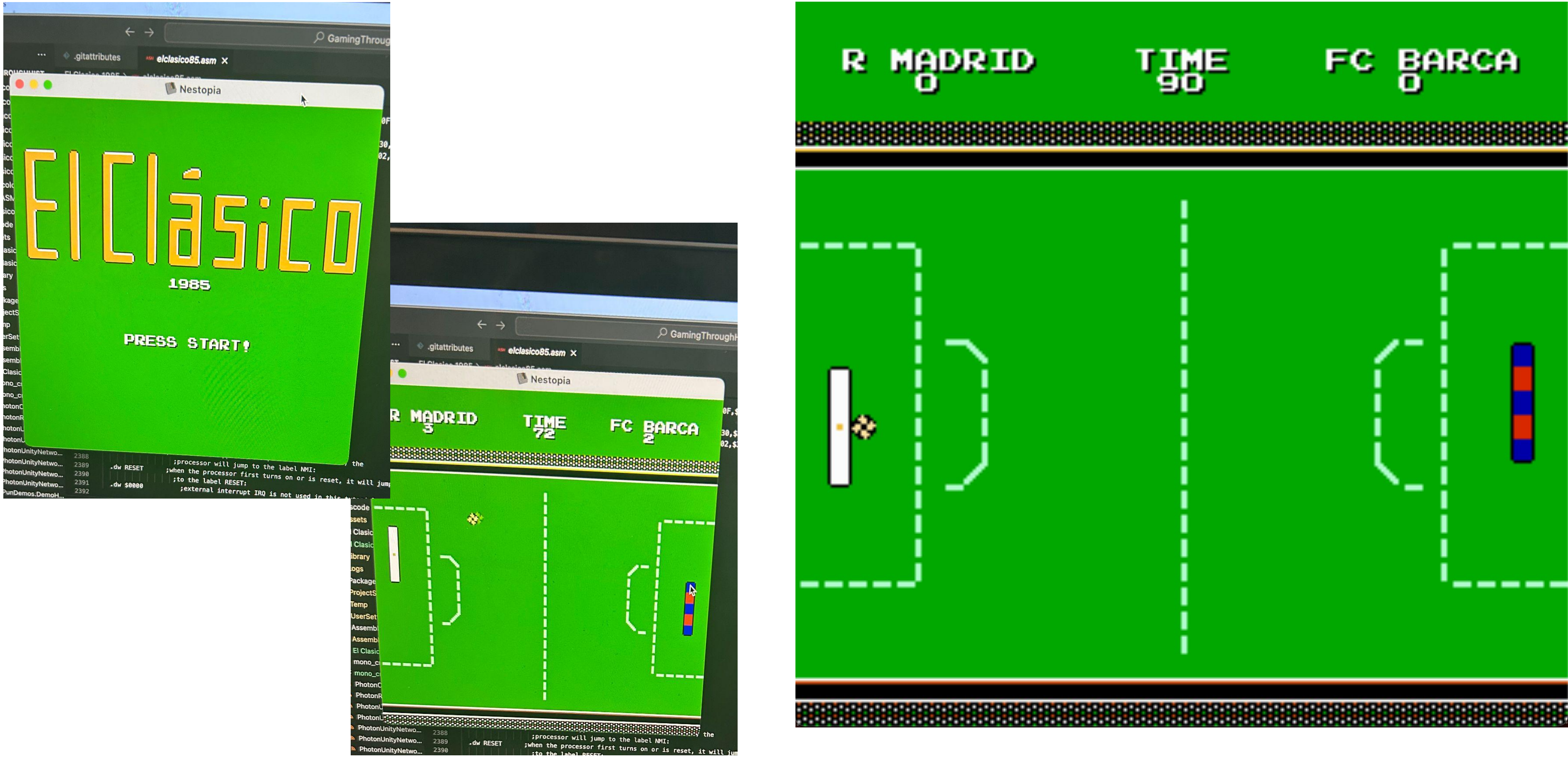
Super Mario Bros. (1985)



The Legend of Zelda (1987)

El Clásico 1985 (NES)

- Started with an existing pong game, we rebuilt it to be customized for our greater project.
- Created a custom .chr file for our custom sprites.
- Adjusted game mechanics to give player different advantages, a feature common to games of the time.
  - The Real Madrid paddle moves faster, the FC Barcelona paddle launches the ball faster.
- Since NES also uses Assembly, the high-level programming workflow was similar to PDP-1’s.
  - Core game loop calling different functions to read input, detect collisions



```
                          the
                ;processor will jump to the label NMI:
  .dw RESET    ;when the processor first turns on or is reset, it will jump
                ;to the label RESET:
  .dw $0000
                  ;external interrupt IRQ is not used in this
```

Gameplay from “El Clásico 1985”

Unity

- Cross-platform game engine first released in 2005, and used to create both 2D and 3D games.
- Cost: Free (for individuals and small teams)
- The Unity Editor provides a visual interface to create game scenes. Users can drag and drop assets into the editor such as 3D models, textures etc.
- Scripting (C#) can be used to more acutely define the behaviour of object in scenes.
- Offers third party networking capabilities to make online multiplayer games.

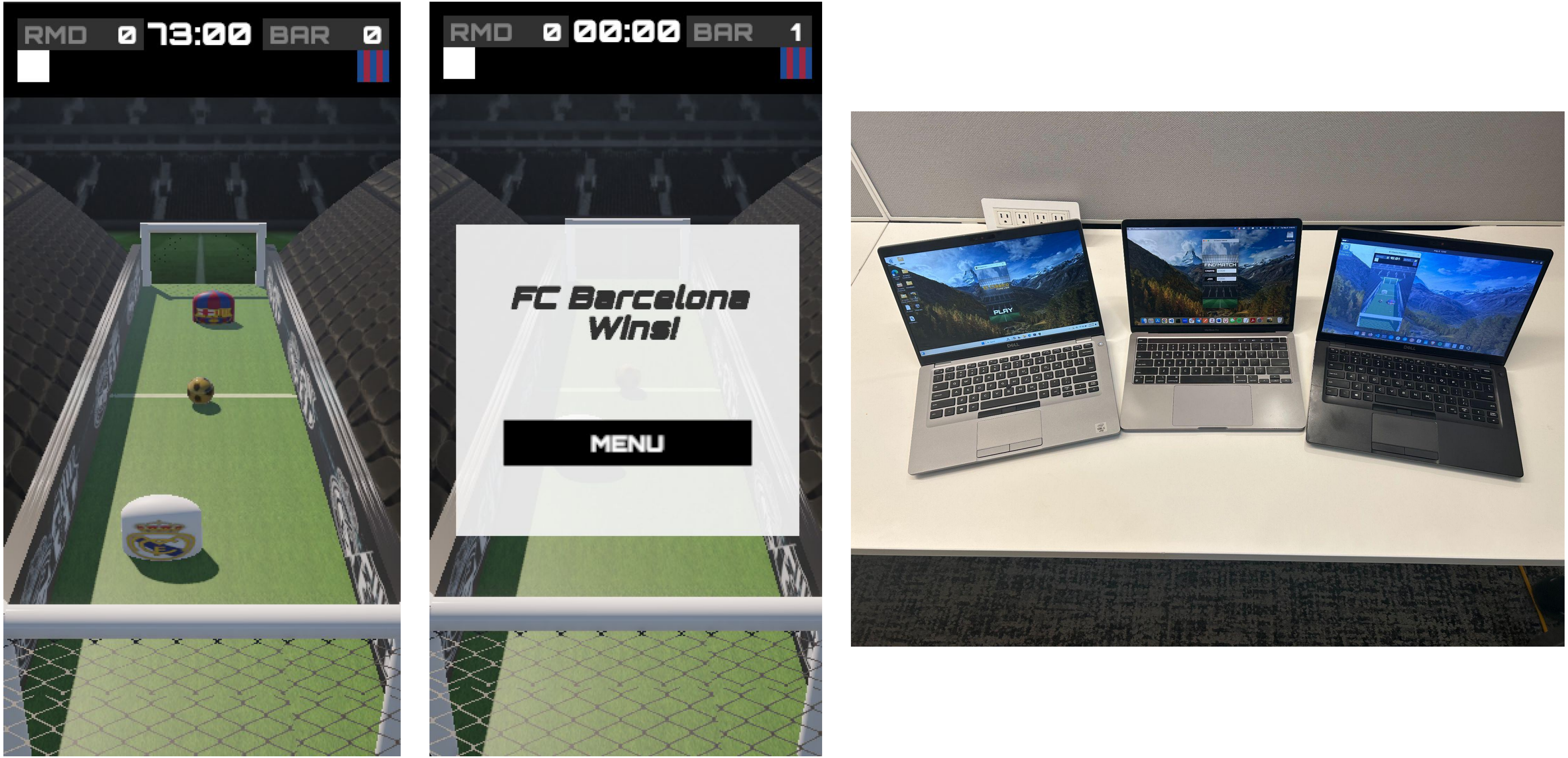
Example Unity games

Pokemon Go (2016)

Among Us (2018)

El Clásico FOREVER (Unity)

- Built Pong game from scratch
- Used 3D assets with customized textures to give the look and feel of a miniature football stadium.
    - “Paddle” skins were designed with logos for each team
- Used Photon Engine for online multiplayer
    - One player creates a game room with a unique code that another player can join. Game is synchronized via a shared server.
- The game runs on desktop, for Windows, Mac, and Linux.



FC Barcelona Wins!

Gameplay from “El Clásico FOREVER”

Comparisons

- Graphics
  - Limited options for graphics on PDP-1 due to point-display
  - NES improved on this with background and sprites. Tiling approach to graphics and coloring due to memory constraints
    - Tiles still require careful positioning and color palette selection to design appearance
  - Unity offers developers a 3D game engine that can be used in tandem with other modeling/graphics software like Blender or Photoshop.
- Coding effort/ease of understanding
  - PDP-1 and NES games required a long ramp up period just to learn the assembly and programming model. The code can also be hard to debug.
  - Unity provides an intuitive object-oriented model, a large Asset Store with useful third party plugins, and built-in physics simulation which simplifies game design.

Thank you!