• **Project Overview**

• **Goal:** Recreate the classic 1980’s arcade game *Donkey Kong*

• **Project Components**
  ▫ **Game Logic**
    • Game FSM
    • DK, Mario, Barrel, Collision Detector Logic
  ▫ **Display Logic**
    • DK, Mario, Barrel, Princess, Background Modules
    • ZBT RAM Implementation of a Frame Buffer

• **Possible Further Explorations**
  ▫ Video Detection Driven Motion
History

- Developed by Nintendo
- Designed by the legendary Shigeru Miyamoto
- Released in America in 1981
- Instant hit
- Considered by many to be one of the greatest games of all time
- Spawned spin-off games and became entrenched in American popular culture


Gameplay

- The user controls Mario with keyboard controls (the up, down, right, left arrows, and the space bar)
  - If time permits, we will implement motion control
- Mario starts at the lowest platform
- Pauline is trapped at the highest platform
- Mario (Jumpman) has to climb up a series of platforms to save Pauline, while Donkey Kong (who is at the top platform) throws barrels to impede Mario’s progression
- Mario can jump over barrels and climb ladders to get to the next highest platform
- Once Mario loses all of his lives, the game is over
- When Mario reaches the Princess, the player wins the level


http://www.classicgaming.cc
Block Diagram - Game Logic

Block Diagram for Game Logic

Block Diagram Description:
- Score Counter:
  - Score[7:0]
  - Reset score
  - Increment score
- Life Counter:
  - Life[2:0]
  - Reset score
  - Decrease life
- Game Finite State Machine:
- Barrel_Coord. RAM (10x21)
- Collision Detector:
  - Collision barrel
  - Collision princess
- Donkey Kong Logic:
  - 1/4hz_clock
  - Restart
- Clock Divider:
  - Dk_frame[1:0]
  - Restart
- Barrel Logic:
  - New_barrel
  - barrel_x[10:0]
  - barrel_y[9:0]
  - barrel_num[3:0]
- Mario Logic:
  - mario_x[10:0]
  - mario_y[9:0]
- Video Logic:
  - Pixel count
  - Title screen, Initial_game_screen, Restart

Connections:
- Start
- Reset
- Title screen, Initial_game_screen, Restart
Game Logic- Details

- Collision Detector
  - Determines whether Mario collides with a barrel, jumps over a barrel, or reaches Pauline

- Donkey Kong Logic
  - Uses the ¼ Hz clock to determine when to try to throw another barrel (if max_barrels is not one)
Game Logic - Details

- **Barrel Logic**
  - Writes/Reads the coordinates of each barrel from a 10x21 RAM
  - Creates a new barrel when told so by Donkey Kong Logic
  - Tells Donkey Kong Logic when there are the maximum number of barrels onscreen
  - Sends the coordinates of each barrel to the Video Logic and the Collision Detector

- **Mario Logic**
  - Takes the control inputs from the user and gives the Collision Detector and Video Logic the coordinates of Mario
  - Also tells the Video Logic which frame of animation to use for Mario
Block Diagram - Display Logic

Background Module
- bg_screen [1:0]
- Life [2:0]
- Score [7:0]

Princess Module
- bg_waddr

Barrel Module
- barrel_x [10:0]
- barrel_y [9:0]

Donkey Kong Module
- Dk_frame [1:0]

Mario Module
- mario_x [10:0]
- mario_y [9:0]
- mario_frame [1:0]

Display Logic
- bg_waddr

BRAM

Display ZBT—Screen Buffer
- radd

XVGA 1024x768 at 60Hz
- R[8:0]
- G[8:0]
- B[8:0]

- v_sync, h_sync, v_count, h_count

To:
- Display ZBT,
- Background,
- Princess,
- Barrel,
- Donkey Kong,
- and Mario Modules
Display Object Specifications

- Donkey Kong
  - 32x64 pixels
  - 3 frames of animation
- Peach
  - 32x64 pixels
  - 2 frames of animation
- Mario
  - 16x32 pixels
  - 4 frames of animation
- Barrels
  - 16x16 pixels
- Background: Platforms, Life & Score Display
  - Spacing between platforms = 64 pixels
  - Each platform height: 32 pixels
  - Each repeating unit: 32x32 pixels
RAM Implementation - Mario Example

- Each of the 4 animation frames stored in a BRAM
- Mario module
  - Inputs: <mario_x>, <mario_y>, <mario_frame>
  - Outputs:
    - <raddr> - which animation frame to load from BRAM
    - <mario_waddr> - where this is placed in the frame buffer
- Frame Buffer – each location maps directly onto a pixel on the monitor
Proposed Timeline

• Game Logic
  ▫ Friday, November 17
    • Finish a basic version of the Mario and Barrel logic
  ▫ Wednesday, November 22
    • Have a completed basic version of all of the main logic modules

• Display Logic
  ▫ Friday, November 17
    • All objects loaded onto ZBT RAM
      • Begin Assembling of Frame Buffer
  ▫ Wednesday, November 22
    • Frame buffer with all 5 objects displayed onto the monitor