* General Description
  + First video arcade game released in 1978
  + First game with concept of saved high scores
  + Used Black-and-white raster display (grid of pixels)
    - Vector graphics did not exist yet
    - Paints from left to right and top to bottom
    - Monitor rotated counterclockwise 90 degrees
  + Colored transparencies that overlay areas of the monitor to give color
    - Green for player area
    - Red for saucer line
  + Screen memory = One-pixel-per-bit -- eight pixels per byte
  + UI
    - Joystick and shoot button
* General Code Structure (Assembly)
  + One file with lots of jumps
  + As it is assembly, there is a lot of lines of code to do simple tasks such as timers
  + This is the code from the arcade box as well so there is code to handle payment processes, sending to screen, controls, tilt switches, etc that we will not necessarily need to recreate thanks to the power or modern languages and technology
  + There is a chuck of 2K bytes from 0x0C00 to 0x13FF that is diagnostics routines that got dropped in release so this is never actually called during the game processes
  + Code Bug at 0x1504
    - Shoot all the aliens but the one in the upper left. Try and stay to the left side of the screen as much as possible because you want to keep the aliens from shooting as much of your right shield as you can. Wait for this last alien to wiggle down the screen. Move to the far right side of the screen under the right edge of your right shield. The alien will turn green as it crosses the screen going from right to left above your shields. Just as it hits the left side of the screen and turns to eat your shields fire a shot into your shield. The game will think your shot hit the last alien and the next round will begin.
  + Five game objects processed by the interrupt routines (2 routines - used to avoid electron beam of screen that would cause multiple display locations of things)
    - Game Objects
      * 0: The Player at the bottom of the screen
      * 1: The Players shot
      * 2: Alien 1 Shot
      * 3: Alien 2 Shot
      * 4: Alien 3 Shot (also flying saucer)
    - 16 byte data structure
      * First 3 - timers used by system
      * Next 2 bytes - pointer to tasks code
        + Sorta like object-oriented virtual function pointer table
      * Remaining bytes are object-specific data
    - X,Y coordinate
      * Routines will run on one of the 2 interrupts based on the Y coordinate
  + Game Object 0: Move/draw Player
    - Player moved up and down on the left of the screen since the screen is rotated in the box
    - Runs every screen refresh
    - First draws the sprite
    - Player can move left and right up to end of screen
    - If player is blowing up then code flips back and forth between two images for half a second
  + Game Object 1: Move/draw Player’s shot
    - One shot at a time
    - Used Y coordinate
    - Several states: just starting, moving up screen, blowing up from hitting alien, blowing up from hitting something else
    - Shot = 4 pixels, speed = 60Hz\*4 pixels = 240 pixels per second
    - Sprite drawing algorithm checks the bits on the screen as it writes the new bits, if any overlaps then a collision is detected and a flag is set
  + Game Object 2, 3, and 4: Alien Shots
    - 3 types of alien shots in the game each with unique images
    - Obj 2 is Rolling shot - spirals a bit as it falls
    - Obj 3 is Plunger shot - looks like a plunger falling down the screen
    - Obj 4 is Squiggly shot - turning zig-zag line
    - One shot is processed per second
    - Speed = 4\*60/3 = 80 pixels per second
      * Less that 8 aliens then 100 pixels per second
    - Rate between shots gets faster as game is progressed
    - Rolling shot fires when right above the player (tracking shot)
    - Plunger is disabled when only one alien left on screen
    - Other two shots fall based off a pointers into a table of columns that advance predictably
      * Squiggly will fall first in column 11 and then 1, 6, 3
    - Saucer
      * Comes from left if players shot count is odd number
      * Comes from right is players shot count is even number
      * 50-300 score range for shooting it
        + Table of 15 entries of scores (50, 100, 150, 300) with wrap at the end
        + Only 1 300 entry
        + 1D54:10 05 05 10 15 10 10 05 30 10 10 10 05 15 10 05
        + Pattern: Count to 8 and start over, from then on count to 15 and start over. You want the 15th shot to be the one that hits the saucer.
  + Aliens
    - 5 rows of 11 aliens
    - 3 types of aliens
    - Each type has 2 images that flip to make an animation with each step
    - Lower left alien is the reference alien
    - Game keeps up with reference aliens coordinates and all others are drawn relative to it
    - 55 byte table tracking living/dead state
    - Alien moves left or right 2 pixels
    - Reverses direction when hitting edge of screen
    - Draws reference alien after it moves, then draws next alien up to all of max 55 so 55 interrupts resulting in almost 1 second to move the whole thing
    - Last alien moves 2 pixels left but 3 right
    - First wave starts at Y=78, then Y=60,then Y=50, then Y=48 for three rounds, then Y=40 for 3 rounds which is right above the players shields which is as hard as the game gets
* Easter Egg
  + 0x199A to 0x19BB
  + A hidden message that will appear at the top of the screen in the menu that reads “TAITO COP” with no “R” as in “CORP”
  + Sequence to active in menu
    - “Start”, “Fire”, “Left”, “Right” at the same time