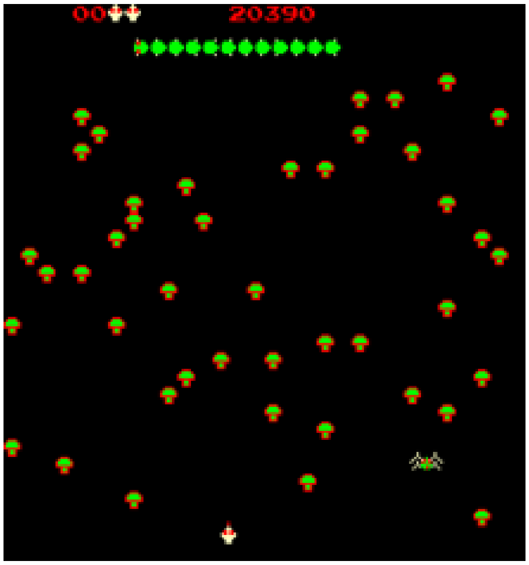
# HW05:

# Centipede - Mode 0 Game



### Provided Files

- Makefile
- print.h/print.c
- gba.h/gba.c
- mode0.h/mode0.c
- sprites.h/sprites.c

### Files to Edit/Add

- .vscode
    - tasks.json
- Main.c
- game.c/game.h

---

## Instructions

For this homework, you will recreate the classic 1981 arcade game Centipede. To get a better feel for the game, you can see some play-throughs on Youtube here for example. Outside of the code in the files provided in the Mode 0 scaffold, you must write all code yourself. This includes not copying from previous lab code.

# Requirements

## Minimum Game Features

- Player controlled Bug Blaster
    - The bug blaster should be represented by a small (16x16) sprite depicting an insect-like creature.
    - The bug blaster should move around the bottom area of the screen using the D-pad.
    - The bug blaster should be able to fire small darts towards the top of the screen.
- Segmented Centipede
    - The Centipede should start at the top of the screen traveling left or right.
    - The Centipede should be made up of multiple segments, each represented by a separate sprite or tile.
        - The centipede should start at least 10 segments long.
    - The head of the Centipede should be represented by a visually distinct sprite/tile.
    - On colliding with a mushroom or reaching the edge of the screen, the centipede should descend one level and reverse direction.
    - On colliding with a dart, a segment of the centipede should turn into a mushroom.
        - In the original game, if a segment being turned into a mushroom causes two halves of the centipede to become disconnected, this creates two centipedes which now travel independently. You can choose to implement this for extra credit.
        - If you decide not implement the extra credit, the “headless” segment of the centipede should all be turned into mushrooms.
    - Once the centipede reaches the bottom of the screen, the player should be taken to the lose state.
    - If the player defeats the centipede, the player should be taken to the win state.
- Mushroom field
    - In addition to the mushrooms created by destruction of segments, each level should start with a mushroom field.
    - The first level should have at least 32 mushrooms.
    - The second level should have at least 64 mushrooms.
- Score
    - Player should receive 100 points for destroying a Centipede head.

  - Player should receive 10 points for destroying a Centipede segment.
  - Player should receive 1 point for destroying a Mushroom.
  - Score should be visible on the top of the screen in both levels.
  - Score should not be reset between levels.
  - Score should be able to display at least 5 digits.
- State machine and multiple levels
  - Must have at least a START, GAME, PAUSE, WIN, and LOSE state
  - The game should begin in the START state, where players can proceed to the first level.
  - After completing the first level, players should be taken to the second level.
  - The player should be able to pause and return to the level they were previously playing without resetting the level.
  - The player should be able to reach a WIN state after completing both levels.
  - The player should be taken to the lose state if they fail during a level.
  - The player should be able to return to the START state from the WIN, PAUSE, or LOSE state.

## Code / Files

Your *code* must have the following:

- Be entirely written in Mode 0, leveraging sprite and tile transparency.
- Good organization.
- Meaningful comments (we're getting stricter about this because it's later in the semester).
- A README.md file.
  - An instruction manual that tells a player how to play your game, like you implemented for previous homeworks.
  - Include markdown formatting

## Extra Credit

If you choose not to implement Centipede splitting for extra credit, game mechanics that make your game more like the original could earn up to 15 points of extra credit. If you decide to do this, *please outline what you implemented in your README file*.

Some examples of this include implementing Fleas, Spiders, or Scorpions, adding a lives system, or anything else included in the original Centipede that we don't already require. For more ideas, feel free to check out the Centipede (video game) Wikipedia page.

## Tips:

- Start early. Never underestimate how long it takes to make a game!
- The simplest way to draw Mushrooms and Centipede segments is to have tiles for these, and represent them by modifying the tilemap.
- The simplest way to implement an extra level is to use an extra state in your state machine to handle multiple levels.

## Submission Instructions:

Ensure that **cleaning** and building/running your project still gives the expected results. **Please reference the last page of previous assignments for instructions on how to perform a "clean" command.**

Zip up your entire project folder, including all source files, the Makefile, and everything produced during compilation **(including the .gba file)**. Submit this zip on Canvas. Name your submission HW05_LastnameFirstname, for example:

“HW05_FerstAlvin.zip”

It is your responsibility to ensure that all the appropriate files have been submitted, and that your submitted zip can be opened and everything cleans, builds, and runs as expected.