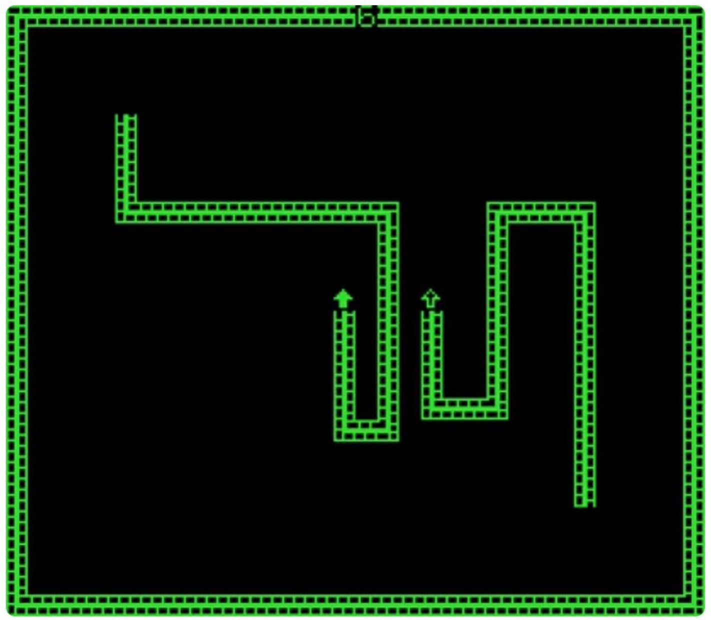
Snake Game

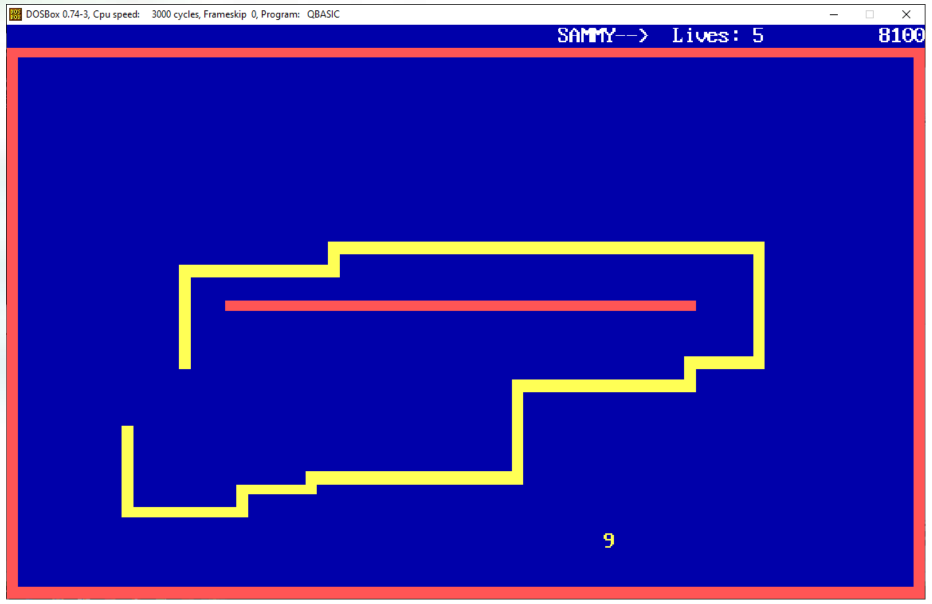
Knox Game Design

February 2026

History

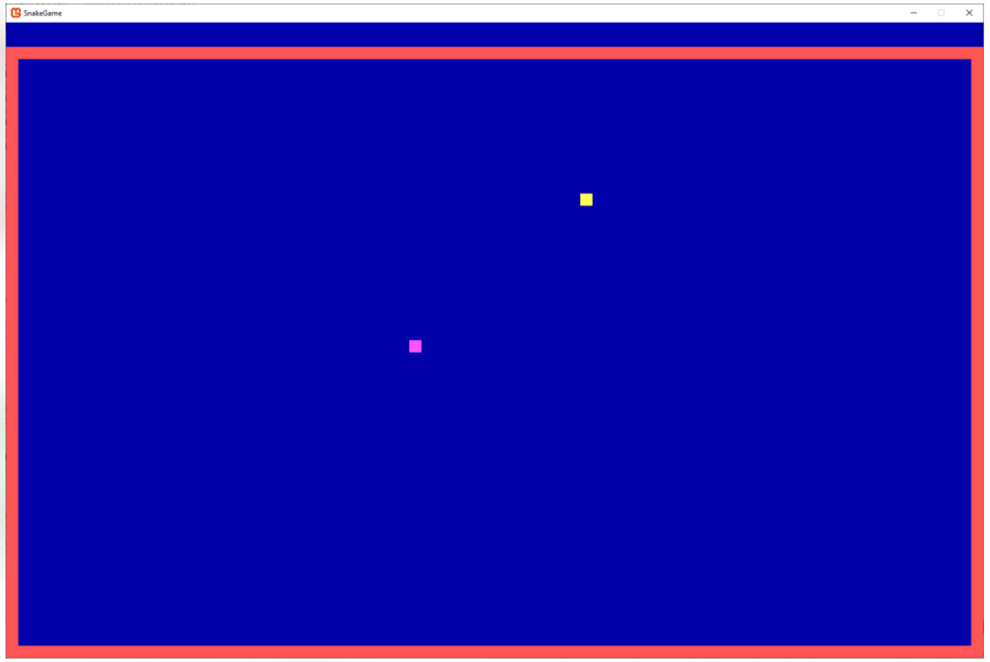
- 1976 arcade game Blockade – Lane Hauck at Gremlin
- Snake game “Nibbles” came with DOS 5.0 in QBasic
- Rick Raddatz at Microsoft 1990
- https://archive.org/details/NibblesQbasic




DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: QBASIC
SAMMY--> Lives: 5
8100
9

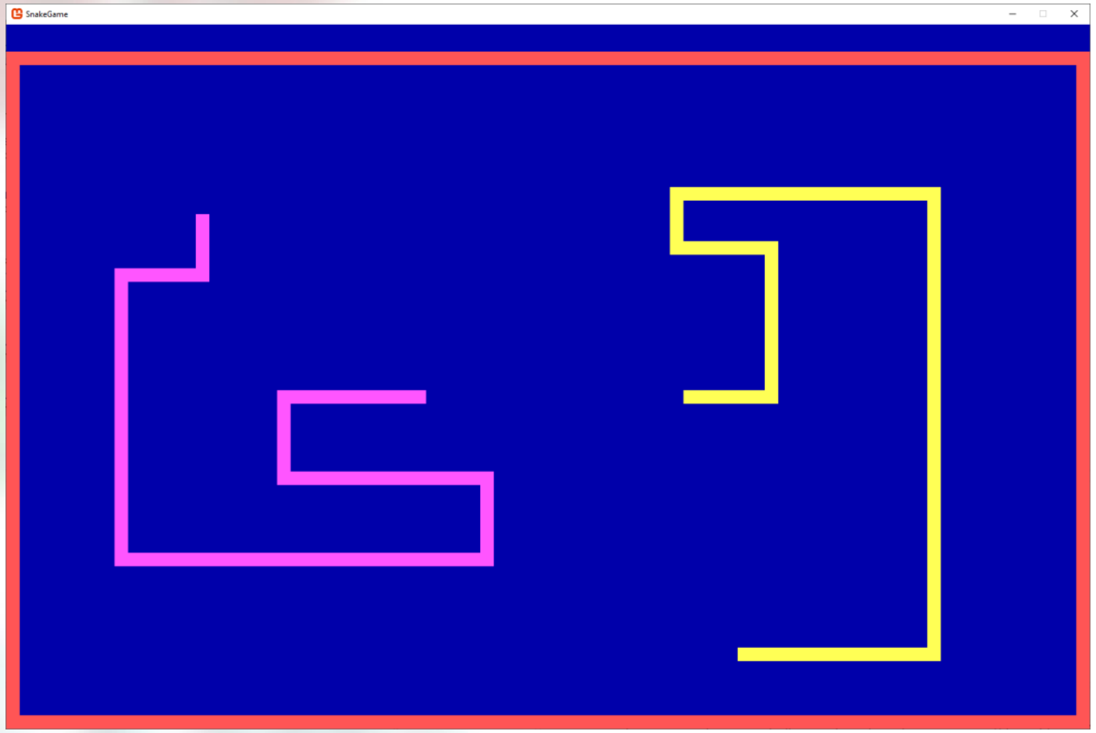
Step 0 – Moving blocks

- Get snake "head" moving around the room
  - Head – moves based on player input
    - Don't allow player to move back the direction they came
- Create room array 80x50
- Create walls around the room
  - Use byte array
    - 0 == empty
    - 1 == snake 1 body
    - 2 == snake 2 body
    - 3 == arena wall
- Detect collision between head and walls
  - Set isAlive boolean to false
- Will implement two snakes/players



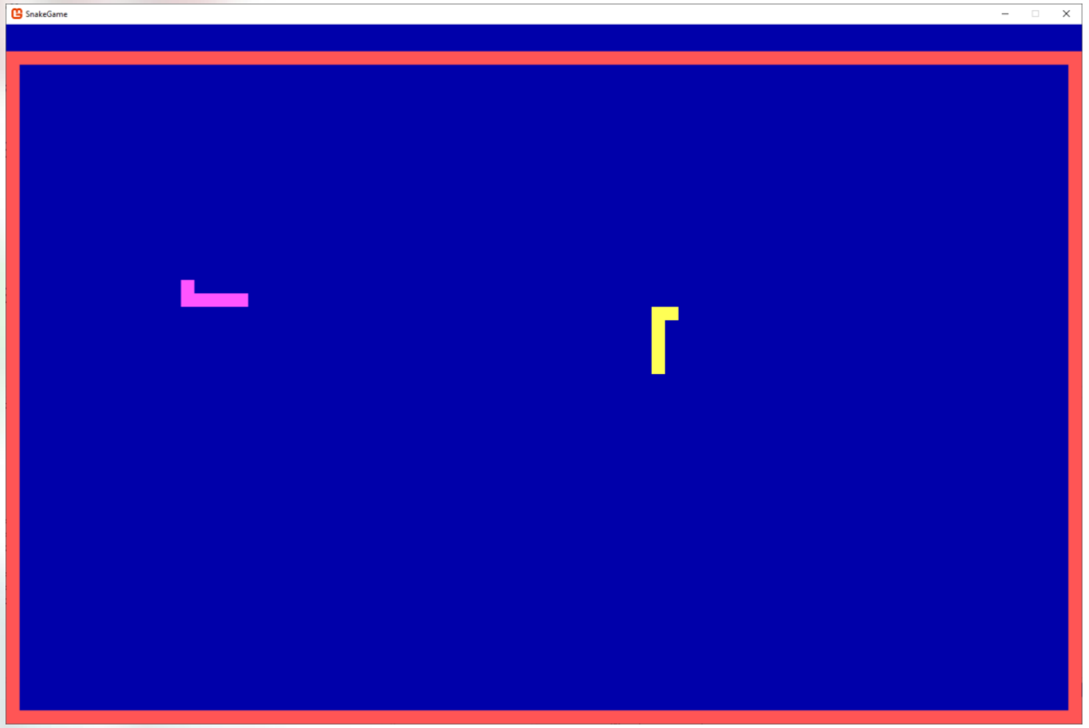
Step 1 – Add snake body

- Add snake “body” to Arena array when head moves
- Update draw method to display snake body cells
- Collision code should automatically work for collision with other snake body
- Will be infinitely long until tail is added



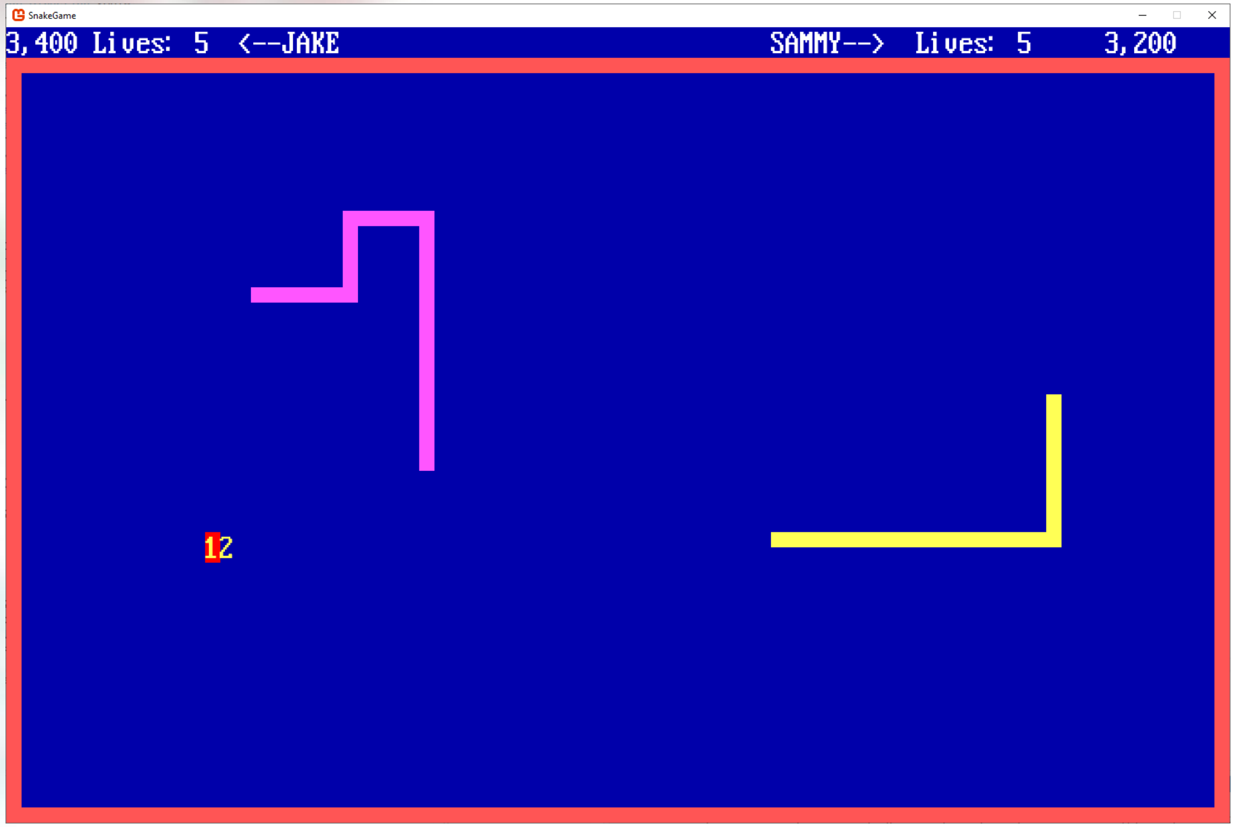
Step 2 – Limit snake length

- Based on snake length property
- Add list of “body” objects (row and column)
  - Required to track which cells need to be deleted from arena
- Delete body objects that exceed the snake length
  - Remove associated cells from arena cell array
  - body cell index 0 will be the “tail” of the snake



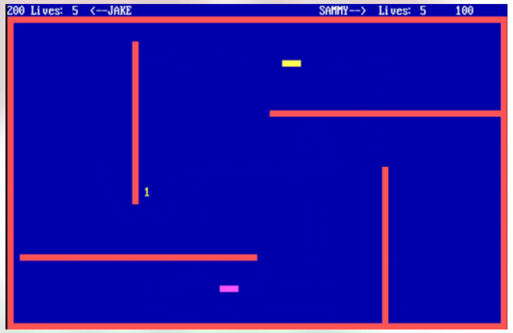
Step 3 – Adding collectibles

- Add collectible on screen to pick up (eat) to extend the length of the snake
    - Collectibles are represented by numerals
- Collectibles should not be placed on top of existing Arena walls or snake body
- When snake head collides with collectible
    - Extend the snake length property
        - collectible value * 4
    - Reset the collectible to a new position
    - Add score to the snake that picked up the collectible
- Helpful to display collectible collision cells during debug testing

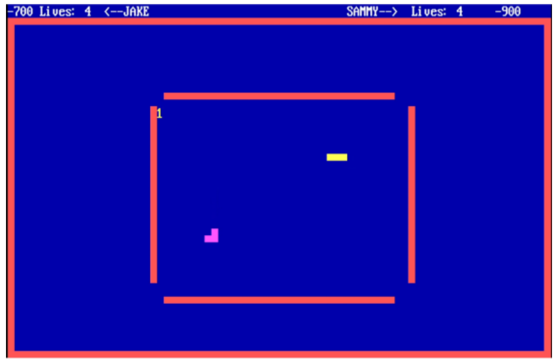


Step 4 – Adding levels, Game States

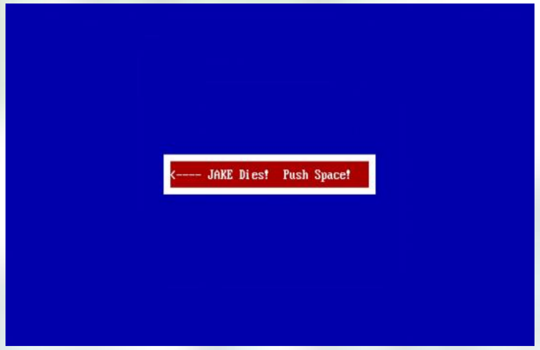
- Add switch statement in level setup code to add the specific wall layout for each level
- Add GameState enum and property to track the title, pre-level, playing, death, and game over states
- Load next level when a snake gets the collectible with value 9
    - Increment level number
    - Reload level with new layout
    - Set snake length property back to default
- Transition to GameOver state when either player's lives falls below 0
- Also added helper methods to center text and filled rectangle around text
- Used Ac437 IBM VGA 8x16 font from *The Oldschool PC Font Resource*
- Note – In QBasic the arena row index started at 3 and col index started at 1. This implementation starts at row index 0 and col index 0.



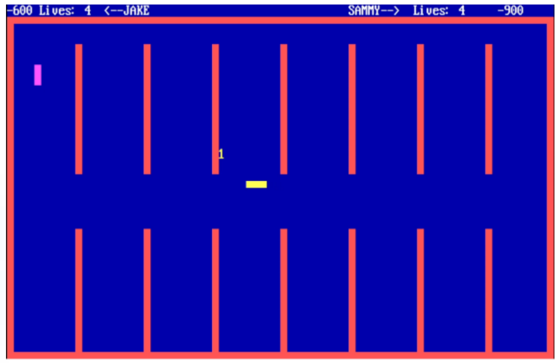
```
200 Lives: 5 <--JAKE                SAMMY--> Lives: 5    100
```



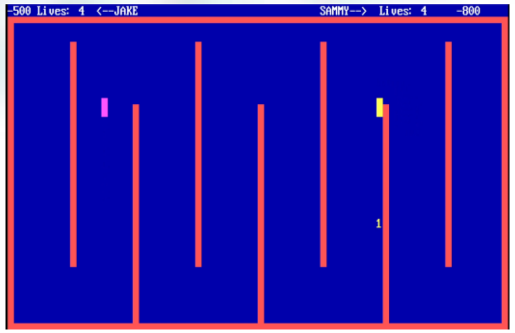
```
-700 Lives: 4 <--JAKE               SAMMY--> Lives: 4    -900
```



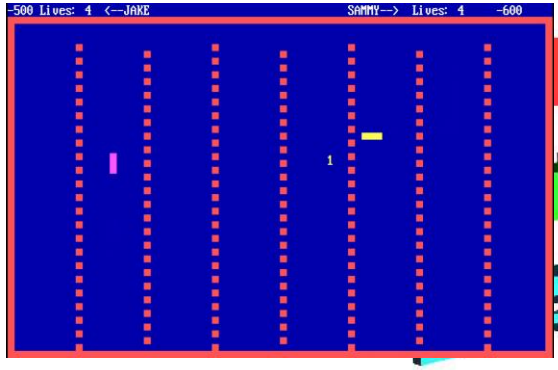
```
<---- JAKE Dies!  Push Space!
```



```
-600 Lives: 4 <--JAKE               SAMMY--> Lives: 4    -900
```



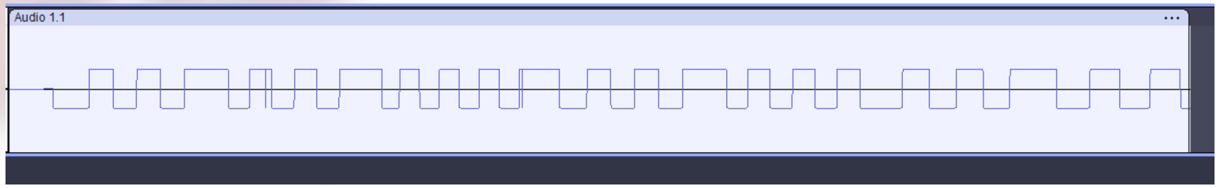
```
-500 Lives: 4 <--JAKE               SAMMY--> Lives: 4    -800
```

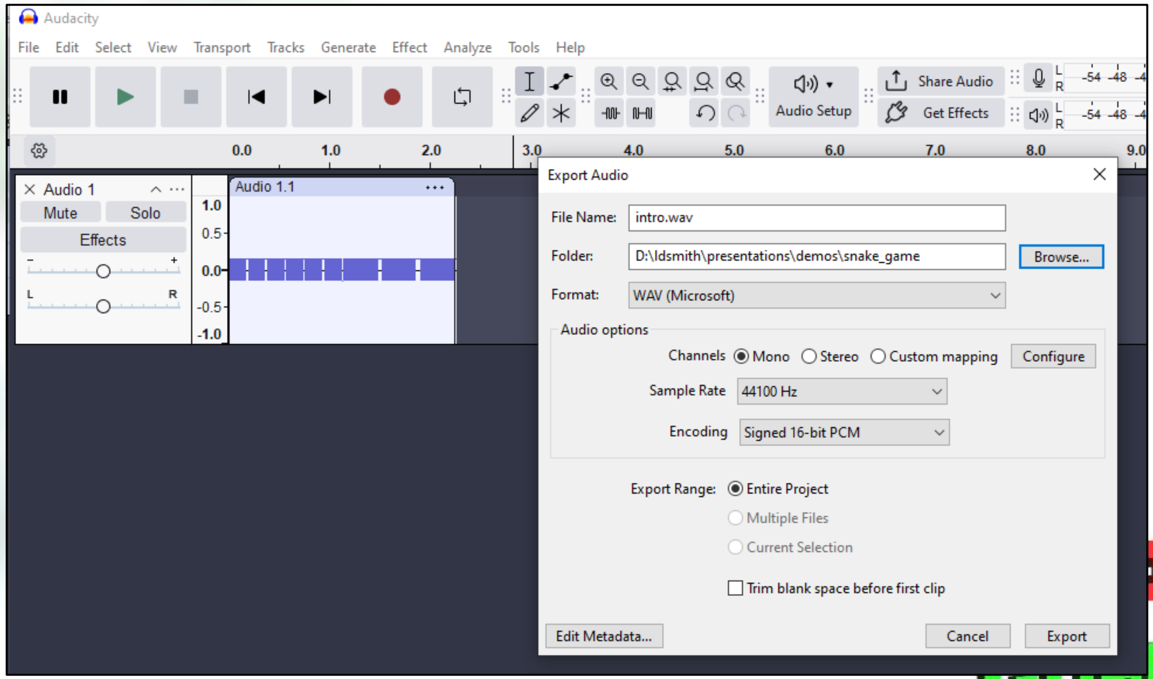


```
-500 Lives: 4 <--JAKE               SAMMY--> Lives: 4    -600
```

Step 5 – Sound Effects

- Record sounds with Audacity
  - Host - Windows WASAPI
  - Recording Device – loopback
- Cut clips and export to WAV files
- Import sound files into project Content.mgcb
- Create Dictionary of SoundEffect objects
- Load using Content.Load<SoundEffect>
- Play with sounds[“<key>”].Play()




Audacity
File Edit Select View Transport Tracks Generate Effect Analyze Tools Help
Audio Setup
Share Audio
Get Effects
Audio 1
Mute
Solo
Effects
Audio 1.1

Export Audio

File Name: intro.wav

Folder: D:\ldsmith\presentations\demos\snake_game Browse...

Format: WAV (Microsoft)

Audio options

Channels [✓ Mono] Stereo Custom mapping Configure

Sample Rate 44100 Hz

Encoding Signed 16-bit PCM

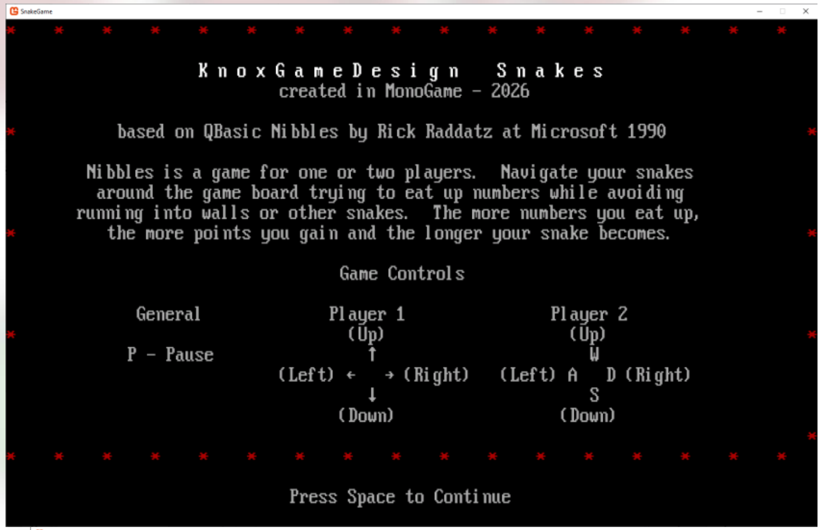
Export Range: [✓ Entire Project] Multiple Files Current Selection

☐ Trim blank space before first clip

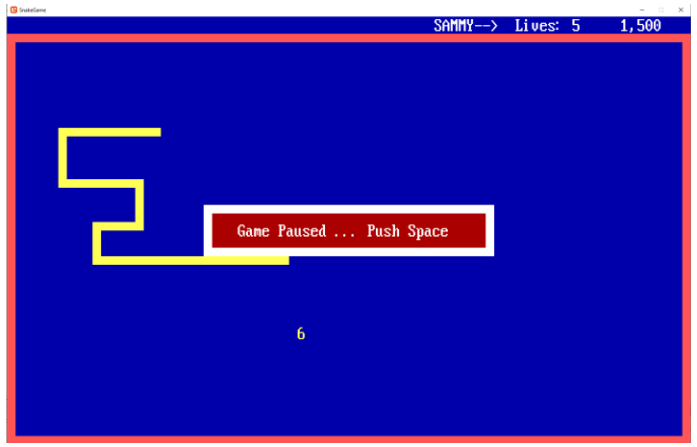
Edit Metadata... Cancel Export

Step 6 – Title Screen, Options, Pause

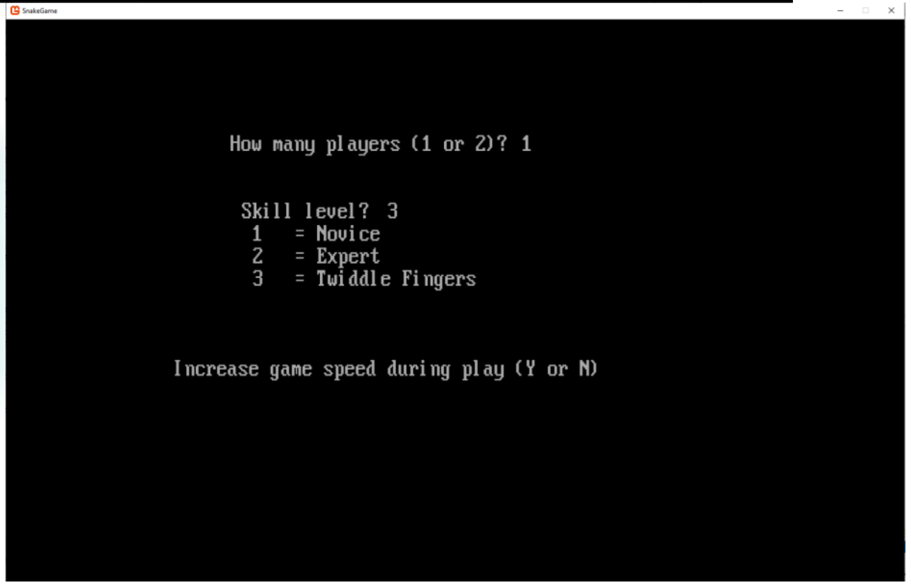
- Screen state tracked in the GameManager gamestate
- Add title screen at game start / intro state
  - Add additional CharacterRegion to spritefont to display the arrow keys
- Options screen to allow player to input
  - number of players
  - game speed
  - increase speed during play
- Pause screen – does nothing until space is pressed
  - First call Playing Draw method, then call the Pause Draw method



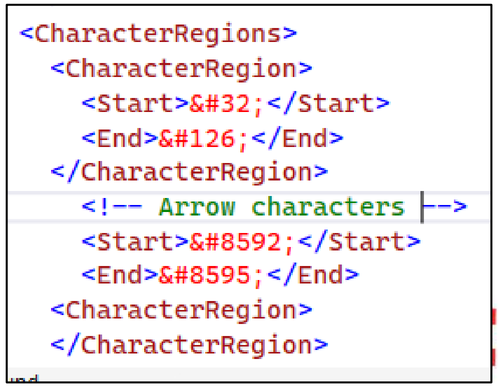
```
K n o x G a m e D e s i g n    S n a k e s
created in MonoGame - 2026

based on QBasic Nibbles by Rick Raddatz at Microsoft 1990

Nibbles is a game for one or two players.  Navigate your snakes
around the game board trying to eat up numbers while avoiding
running into walls or other snakes.  The more numbers you eat up,
the more points you gain and the longer your snake becomes.

Game Controls

General            Player 1              Player 2
                     (Up)                  (Up)
P - Pause             ↑                     W
            (Left) ←     → (Right)   (Left) A   D (Right)
                      ↓                     S
                    (Down)                (Down)

Press Space to Continue
```



```
SAMMY-->  Lives: 5     1,500

Game Paused ... Push Space

6
```



```
How many players (1 or 2)? 1


Skill level?  3
 1   = Novice
 2   = Expert
 3   = Twiddle Fingers


Increase game speed during play (Y or N)
```



```
<CharacterRegions>
  <CharacterRegion>
    <Start>&#32;</Start>
    <End>&#126;</End>
  </CharacterRegion>
    <!-- Arrow characters -->
    <Start>&#8592;</Start>
    <End>&#8595;</End>
  <CharacterRegion>
  </CharacterRegion>
```