

This is CS50.  
Games Edition

Super Mario Bros.



# Our Goal



# Topics

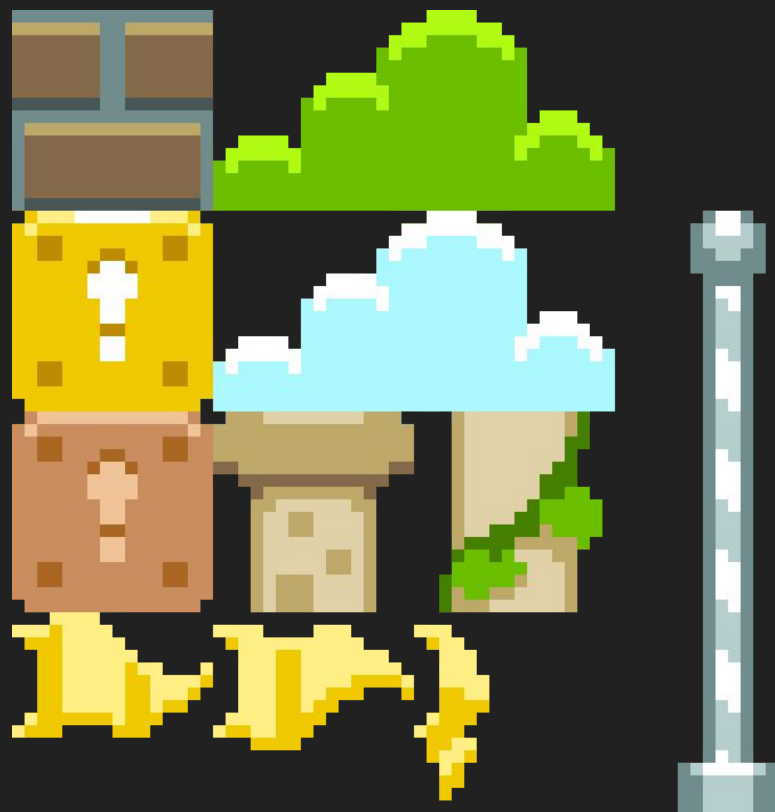
- Spritesheets and Raster Graphics
- Tile Maps
- Gravity
- Animation
- Procedural Generation

mario0

"The Tiles Update"



# Sprite Sheets



# Tile Maps

- Data representation of a grid of tiles to be rendered to the screen
- For a simple tile map, 1:1 mapping between the # in the map and the # of the sprite in our sprite sheet we want to draw
- Can be stored in a text file, generated via visual editors (like Tiled), or generated programmatically via some procedural algorithm

# Example Tile Map Data

```
map = {  
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,  
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,  
    1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,  
    1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1  
}
```



**mario1**

**"The Scrolling Update"**



# Scrolling

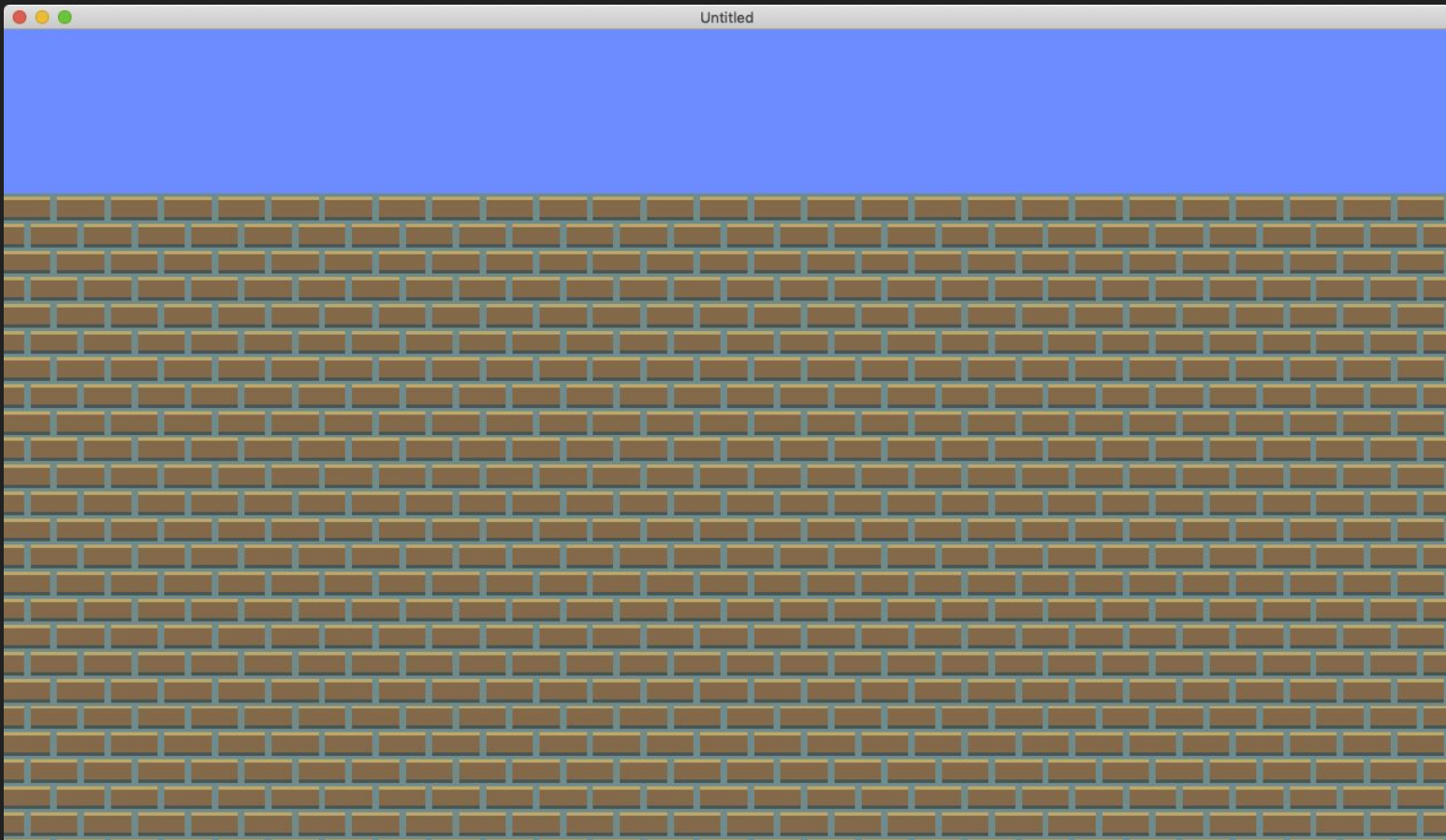
- We keep track of the top-left corner of where we want our “camera” to be, so to speak
- Whenever we want the camera to scroll in any direction, we adjust either the X or Y offset value
- When the scene is drawn, we “translate” the entire scene by the negative value of these offsets; provides the illusion of camera movement
- Performed by a call to `love.graphics.translate(x, y)`

# mario1, Important Functions

- `love.graphics.translate(x, y)`
  - Shifts the coordinate system by x, y; useful for simulating camera behavior.

mario2

"The Controls Update"



**mario3**

**"The ProcGen Update"**





mario4

"The 'Mario' Update"



mario5

"The Movement Update"



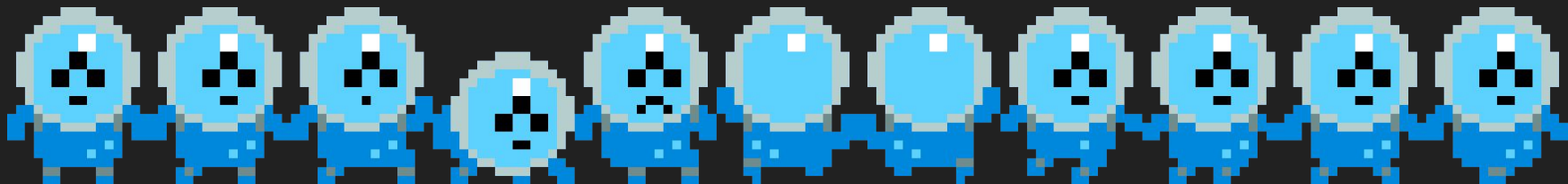
mario6

"The Animation Update"

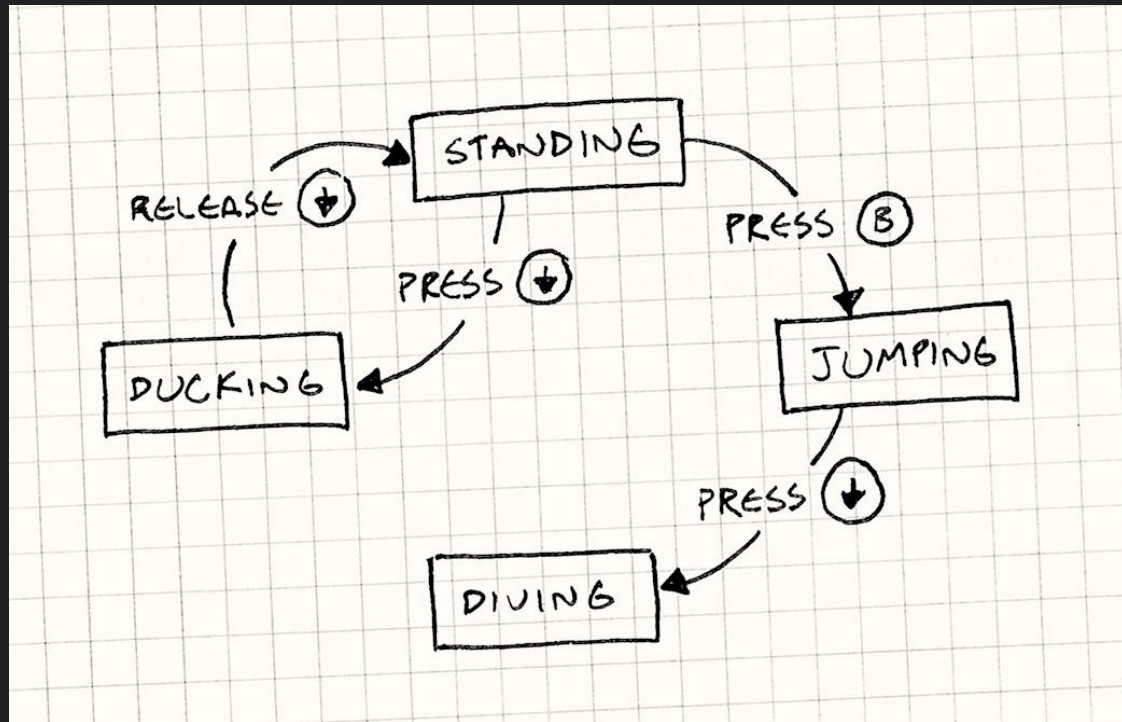


# Sprite Animation

- Relies on a sprite sheet (or “texture atlas”) just like our tile maps, with one Quad associated with each frame of animation
- The sprite’s current frame is swapped with another frame depending on how much time has elapsed and what that sprite’s animation interval is
- An interval of 0.25 means the animation will play at 4 FPS; an interval of 0.025 will be 40 FPS (once that amount of seconds has passed, move on to the next frame)



# State



Credit: Robert Nystrom

<http://gameprogrammingpatterns.com/state.html>



mario7

"The Jump Update"



# How to Jump?

- Jumping is similar to moving, only we're applying translations to our Y velocity instead of our X velocity
- By giving Mario a large negative Y velocity, he will shoot into the air, which is the first part of proper jumping behavior
- In order to properly implement jumping, we also need the notion of gravity to iteratively add to Mario's Y velocity and keep Mario grounded

**mario8**

**"The Block Update"**



# Collision Detection

- For Mario to hit a brick and for that to have some sort of result, we need to see whether or not he has collided with a brick while jumping
- This can be done by calculating whether the top of his “bounding box” (the imaginary rectangle comprising his sprite) overlaps the bounding box of the specific block tile we want to check for
- While jumping, we can ask the game to see whether the tiles directly above him match that tile’s ID, then switch their ID to the block’s “hit” frame
- We should be sure to check the tile directly above his top-left corner and top-right corner, in case he happens to be standing underneath two blocks at once (in between)

# Tile Collision (up)



mario9

"The Collision Update"

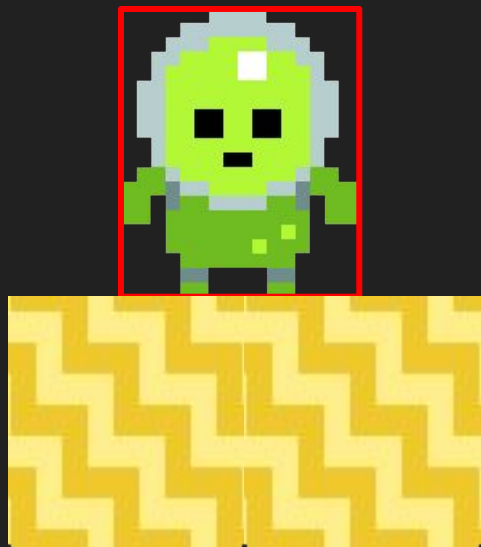




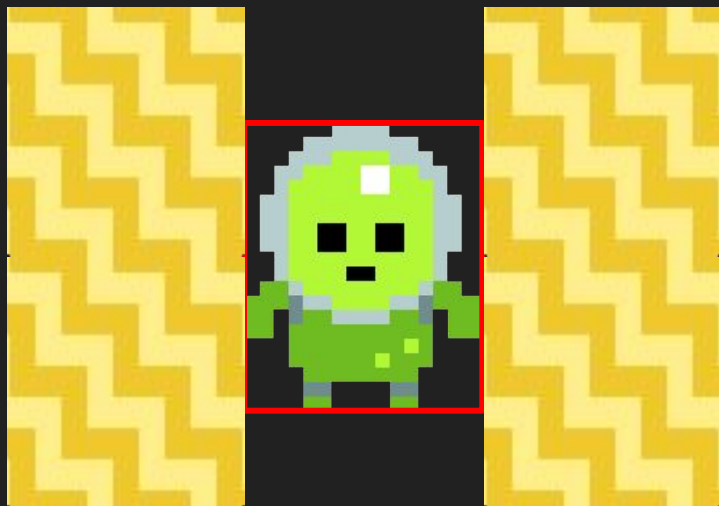
# Tile Collision (up)



# Tile Collision (down)



## Tile Collision (left/right)

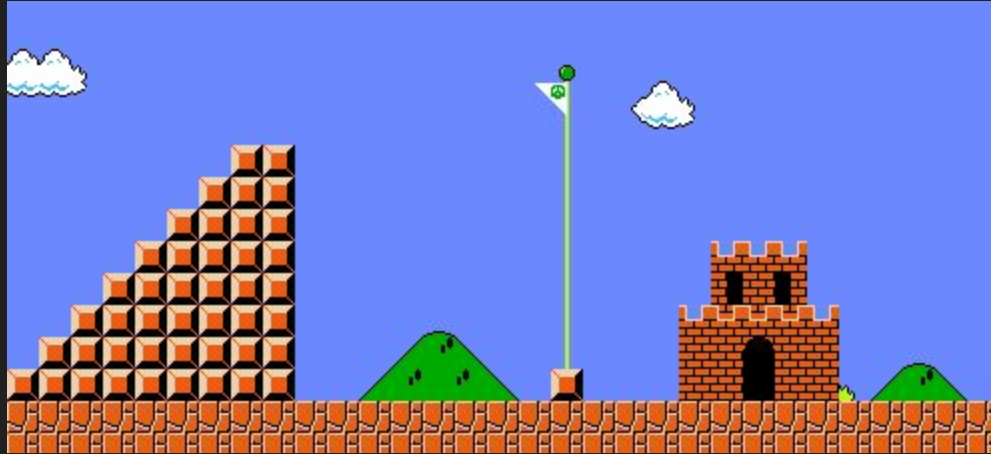


mario10

"The Musical Update"



# Mario Assignment



<https://github.com/bmuellerhstat/01-mario-lab>

This is CS50.

