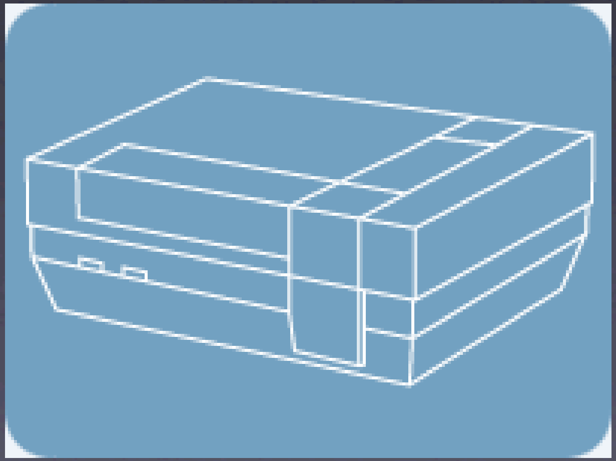
98-026
Nintendo

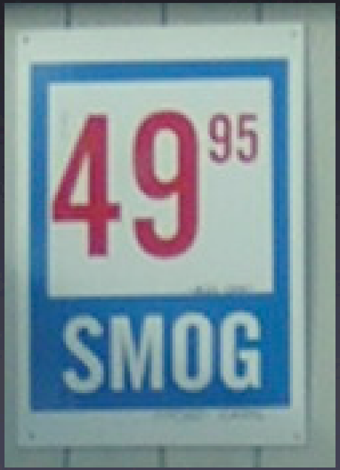
Bob Rost
March 31, 2004



Today

- Announcements
- Making levels, the Tomley way
- Side Scroller Design

Announcements



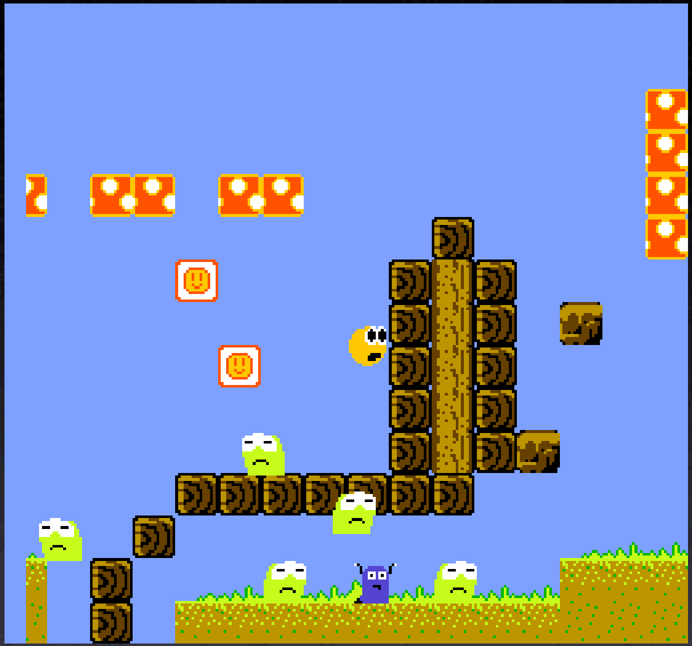
Announcements

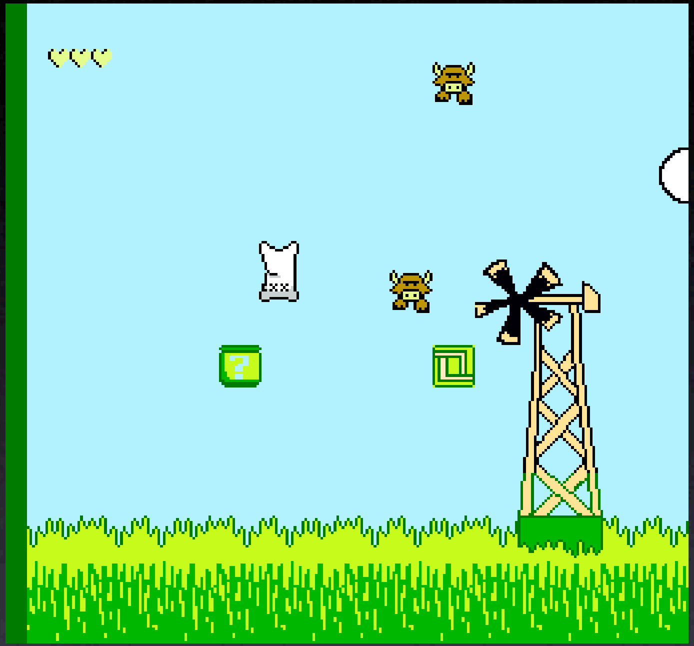
- There was no class last week. How many of you attended anyway?
- Assignment 3 general warning:
  due probably in about 4 or 5 weeks.
- Sean’s weekly demo, and maybe more.

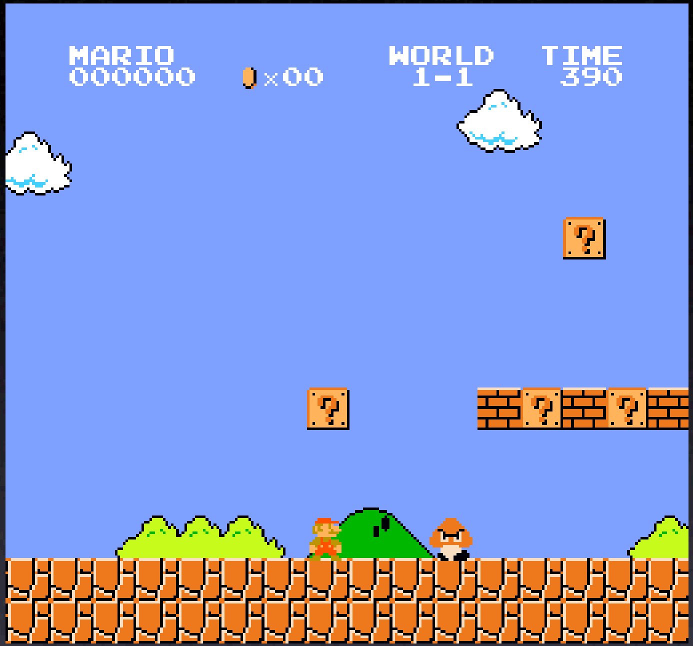
Side Scrollers

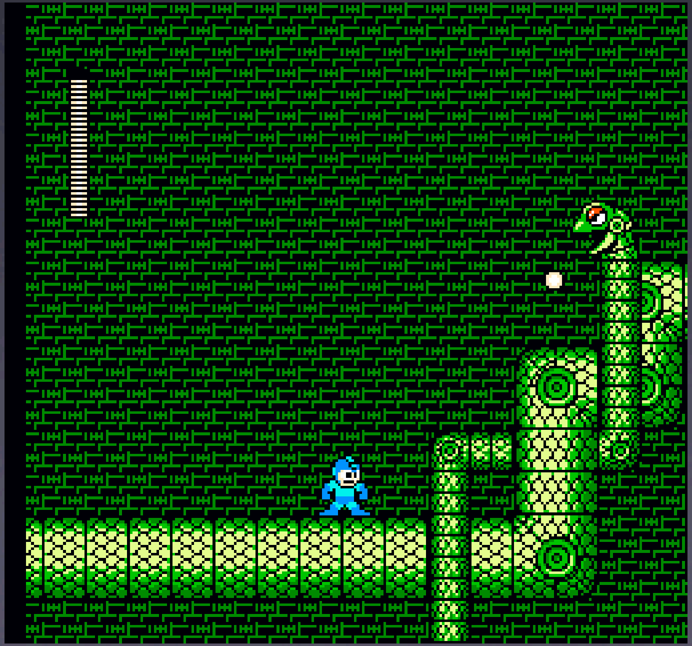
- Gameplay format popularized by the NES
- Has a main character controlled by the player, moving mostly horizontally across the level
- Character often jumps on floating platforms, hence the term “platformer game”.

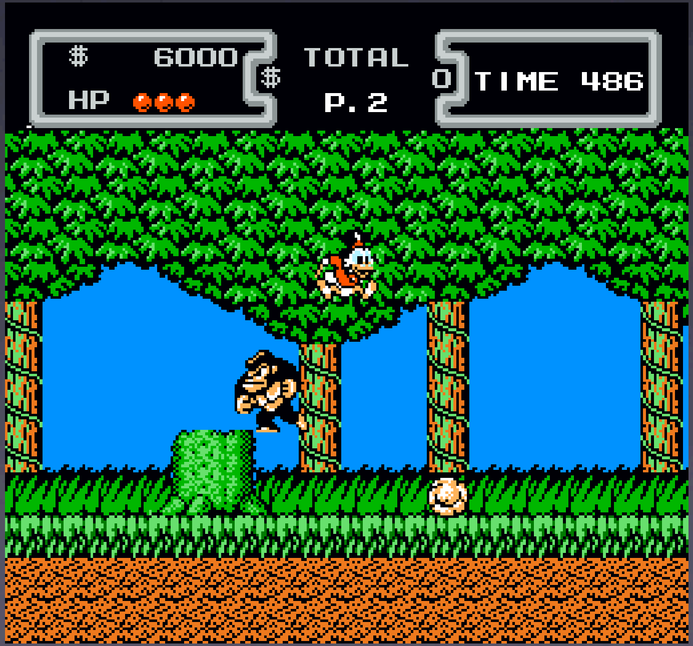
Common Platformers







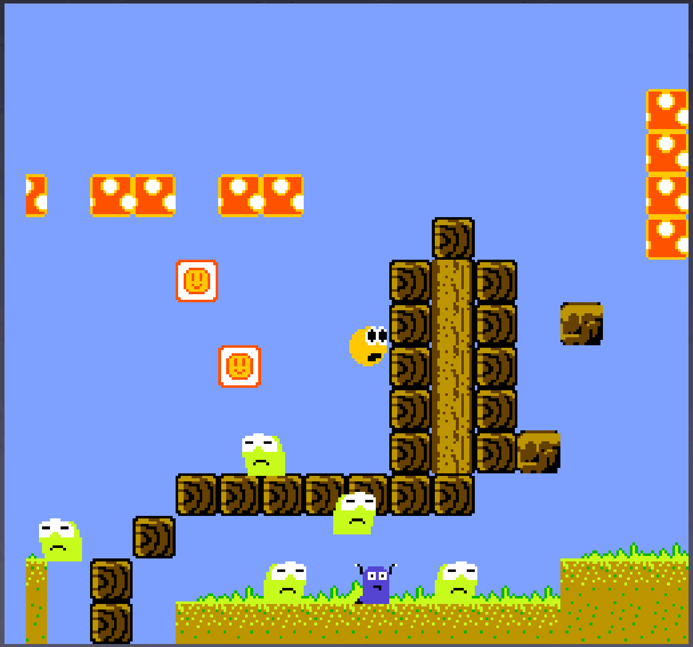




Making the Level

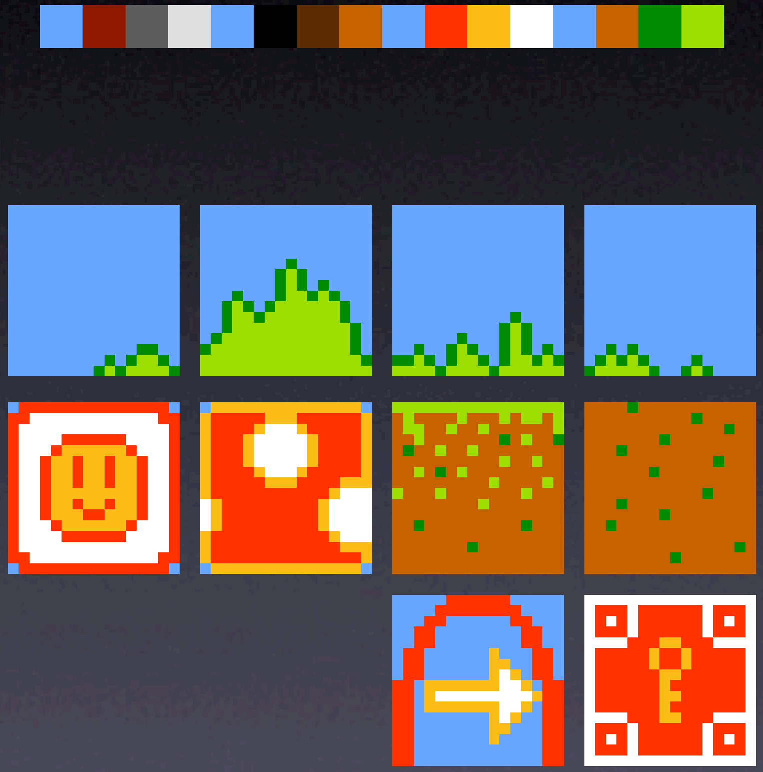
- The Tomley Kleene demo uses the tool bmp2level to create usable level data with 4-square compression. It requires several input images:

  level map
  pattern table
  palette
  collision map
  special data



Making the Level

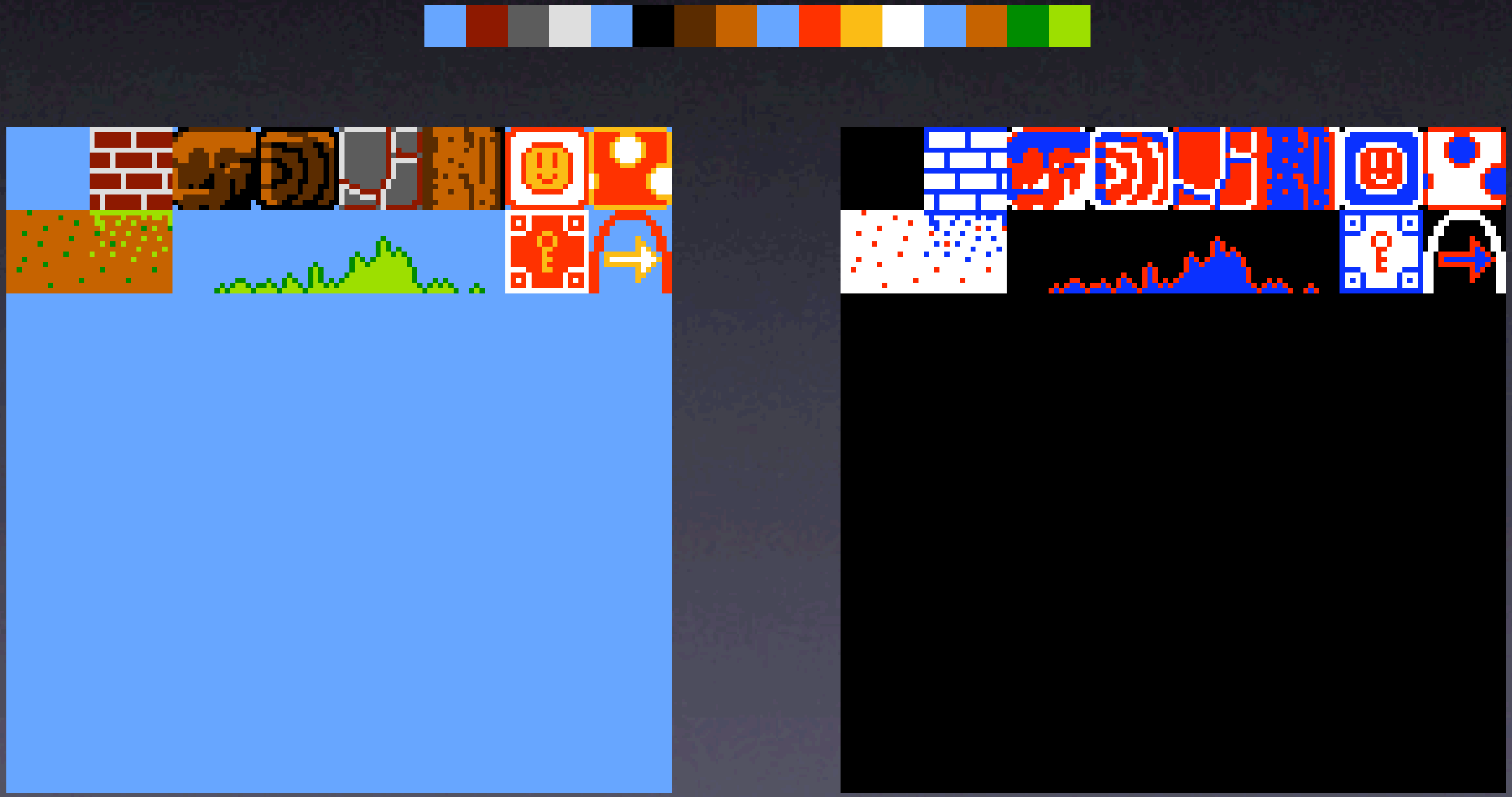
Create background blocks and palette



Tomley Kleene demo: pattern table and palette

Making the Level

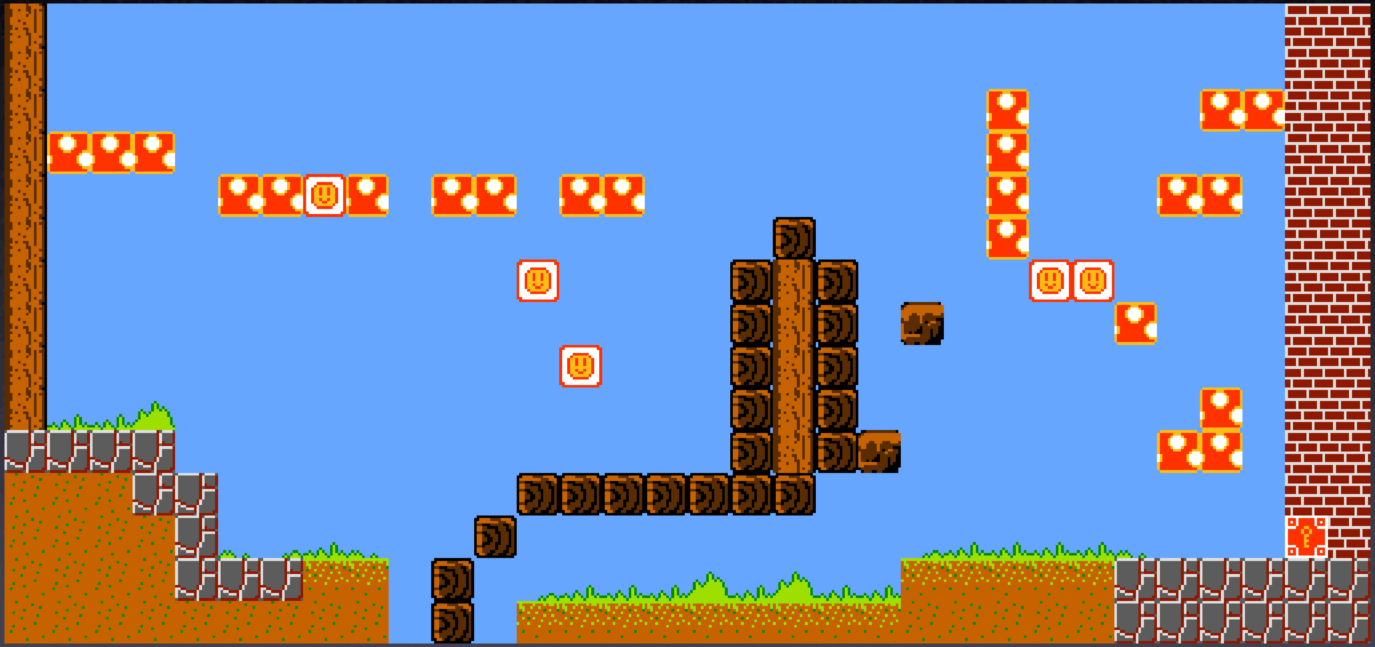
Create the background pattern tables.
(convert to 4 color with bmp2table)



Tomley Kleene demo: pattern table and palette

Making the Level

Copy blocks from pattern table into level

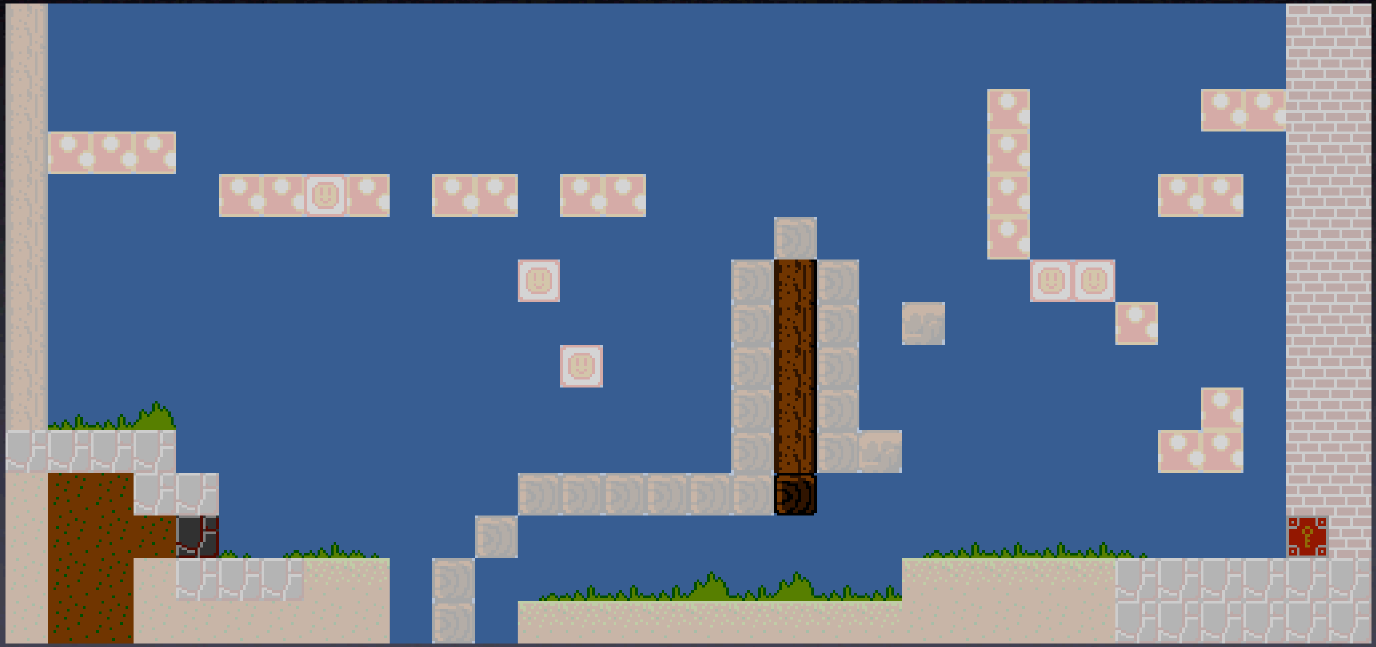


in this case, the level must be exactly 1 or 2 screens wide

Tomley Kleene demo: level map

Making the Level

create collision data to match the map

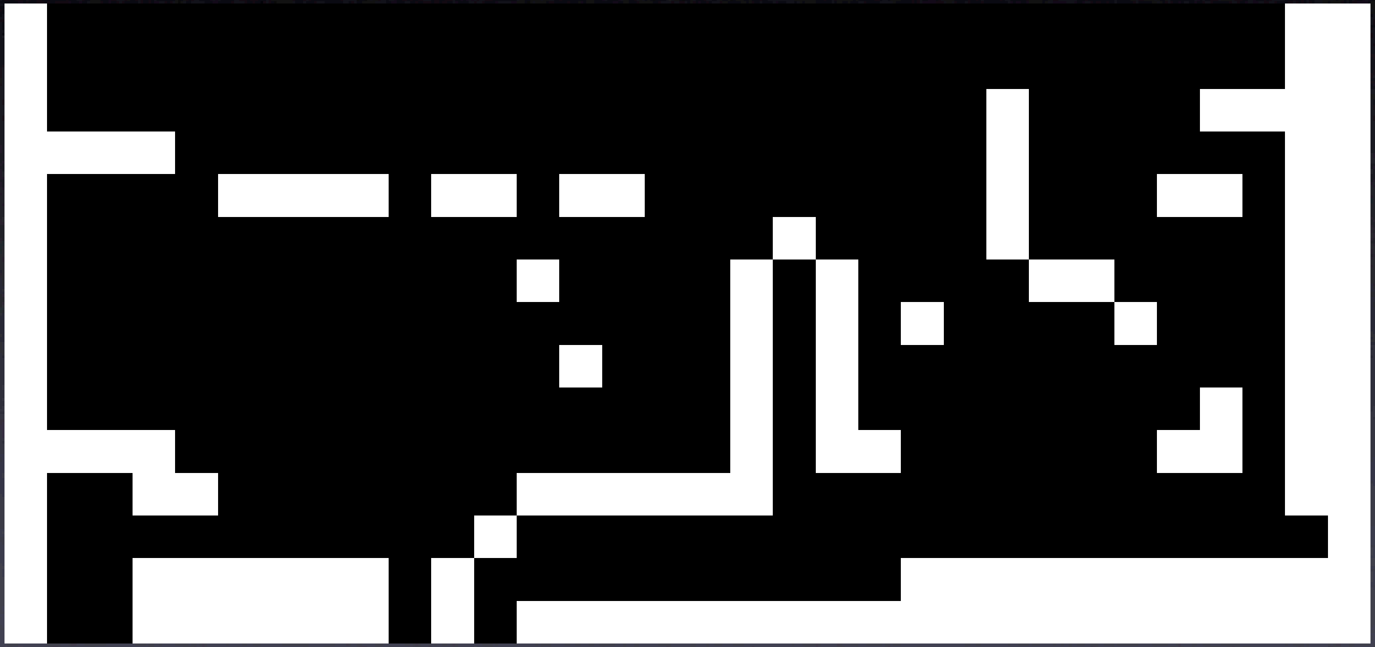


Each square on the map is either solid or pass through. The collision map contains white or black to mark each square type. This overlay shows how it corresponds to the map image.

Tomley Kleene demo: collision map

Making the Level

save collision data as an image



The collision data by itself.

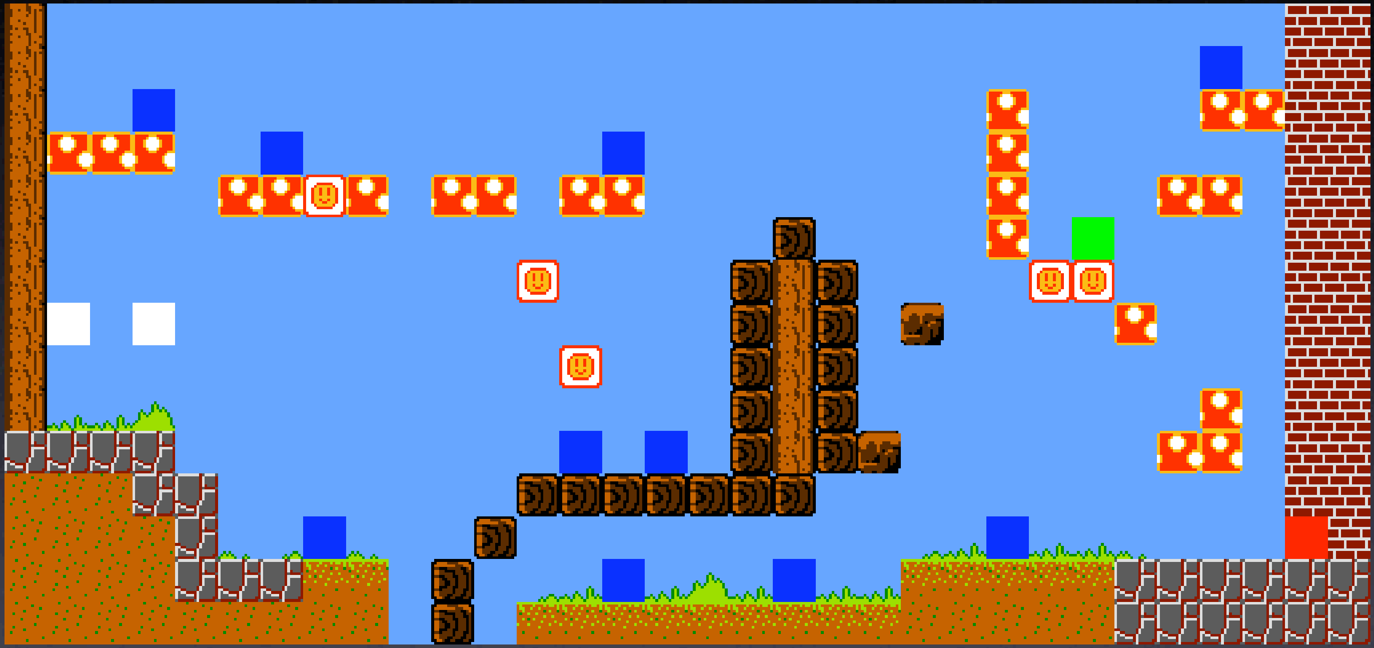
Tomley Kleene demo: collision map

Making the Level

- Tomley Kleene uses “special data” to designate spawn points and objects on the map.
- The special data image is similar to the collision map, with colors mapping to types of objects.

Tomley Kleene demo: special data

Making the Level



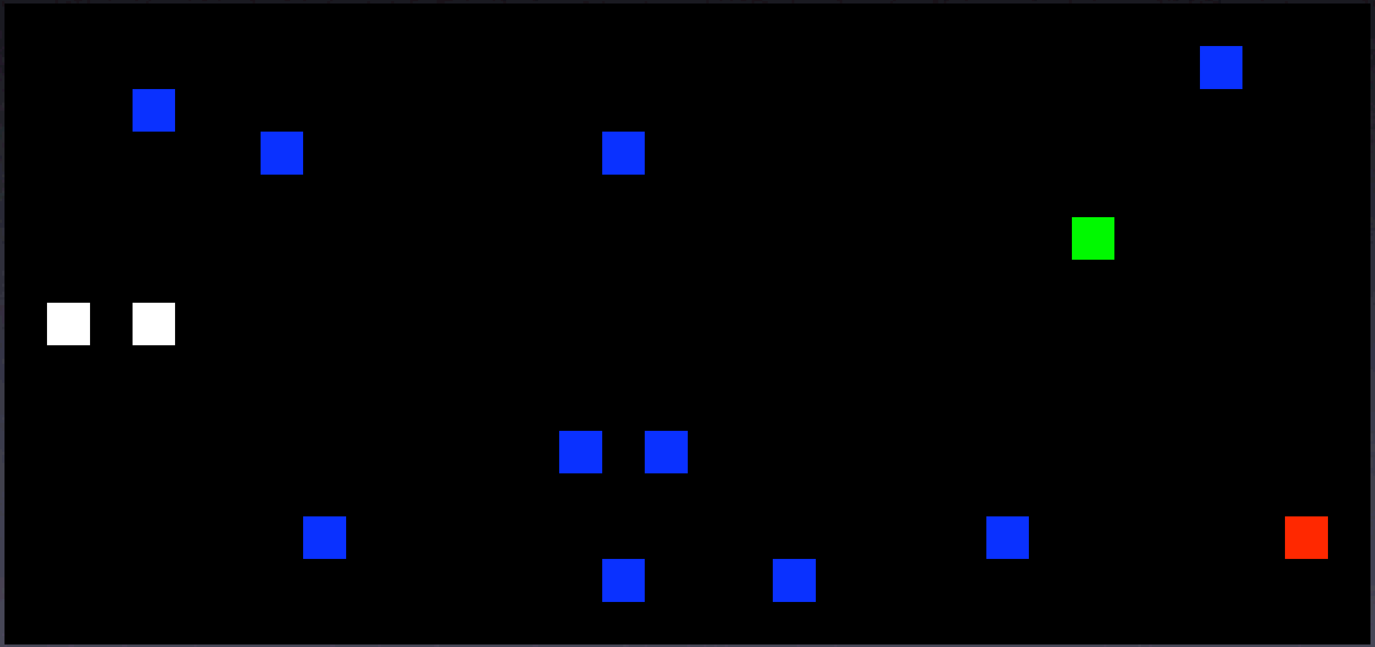
Special data overlaid on top of the map

white	hero spawn point
blue	Grank spawn point
green	Frump spawn point
red	level exit door

Tomley Kleene demo: special data

Making the Level

save special data as an image



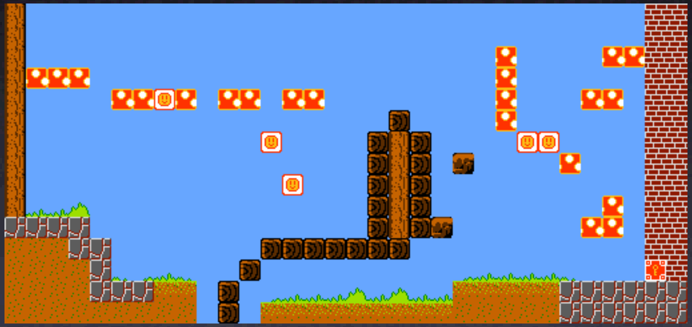
The final special data image

Create Level Data

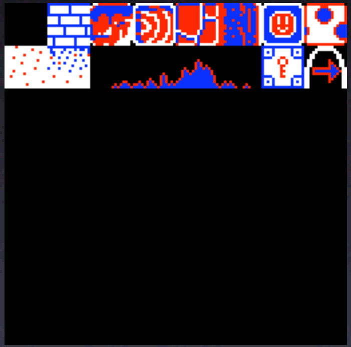
- Tomley uses the tool bmp2level to create level data from these images
- The output data file is included directly in the game. Multiple levels could be used with bank swapping.

bmp2level

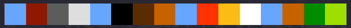
```
bmp2level map.bmp pattern.bmp palette.bmp
      collision.bmp special.bmp level.dat
```



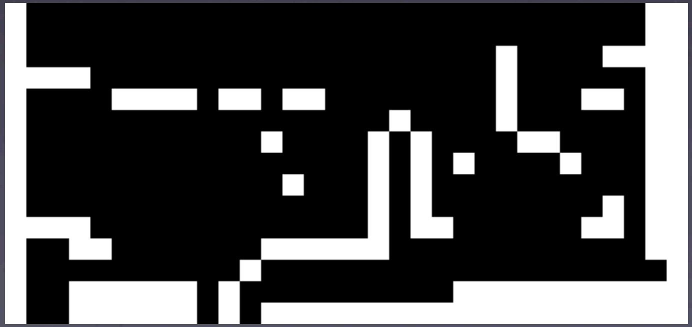
map.bmp



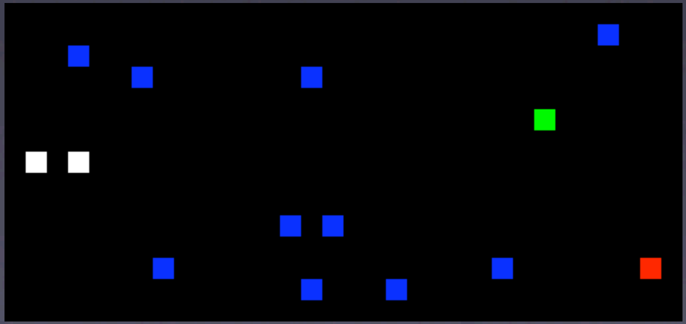
pattern.bmp



palette.bmp



collision.bmp



special.bmp

bmp2level

- The tool is on the webpage, with the nes sprite tools
- More documentation available there, including the binary format.
- Feel free to modify the tool for your own game if you have different needs.

Side Scroller Design

- Conventions
- Characters
- Jumping
- Levels

Conventions

A to jump, B to shoot

- The NES gives you two primary buttons
- If your game allows jumping, you should almost always jump with the A button
- Some games have double jump, to allow you to jump a second time from mid-air, usually also with the A button
- B button is usually used for firing weapons or other attacks
- Some games have multiple attacks and often use a menu screen to choose between weapons (Zelda, MegaMan, Strider)

Conventions

Running

- Two primary running conventions: B and double tap. Allows the player to move faster than walking.
- B button: popularized by Super Mario Bros. The button doubles as run and fire.
- Double tap: popularized by Kirby (others before?). Walking is press and hold left or right. Running is tap, then press and hold.

Characters

- Your main character defines much of the gameplay.
- Visual: size, style, demeanor
- Abilities: attacks, special movement
- Enemies should complement the hero’s abilities
- Most enemies should be easy to kill, and possibly come in large numbers
- Hard enemies (mini bosses) should have a weak point

Jumping

- Most platformers allow jumping. It is definitely higher than humanly possible.
- MegaMan, Mario, and SOF can all jump about 2 - 2.5 times their height. Plan the level around this.
- Bouncing off enemies while holding the jump button sometimes allows a much higher jump
- Timer + gravity jump: As long as jump button is pressed, go up at a constant speed, until a timer runs out or the button is released. Then turn on gravity.
- Air control: During a lateral jump, holding left or right will often directly control or add acceleration in the pressed direction
- Double jump: Not often allowed, but can be cool

Levels

Objectives

- Get to the end *(Super Mario Bros)*
- Find the key or other object *(Mickey Mousecapades)*
- Kill all the enemies *(Tomley)*
- Solve the puzzle

Levels

One-way scrolling

- Mazes usually do not work
- Often jumping between platforms, avoiding pits while killing/avoiding enemies
- Often a safe path to follow near a dangerous or difficult one with powerups
- No dead end death traps. Not fair.

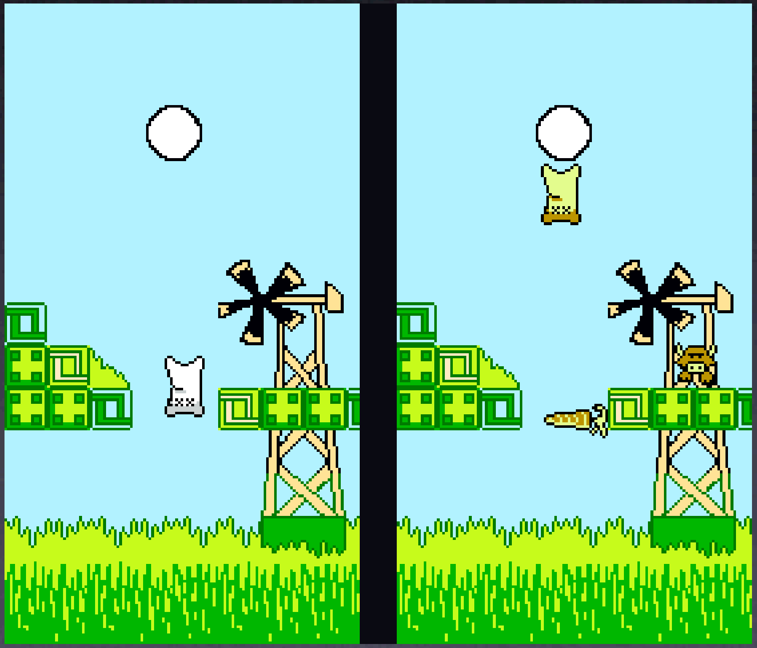
Levels

Two-way scrolling

- Need an incentive to scroll both ways
  - Kill all enemies
  - Find all powerups or secret areas
  - Unlocking doors after finding key
  - Level design requires going both ways

Levels

- Design your level around your character, enemies, and scrolling styles
- Beware of impossible jumps: take note of how high your character can jump



an impossible jump,
until you bounce off an enemy

GAME OVER