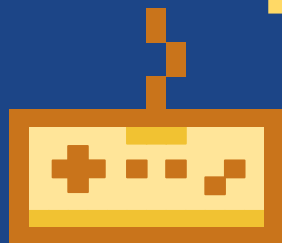


CS 91S Game Systems
Fall 2024

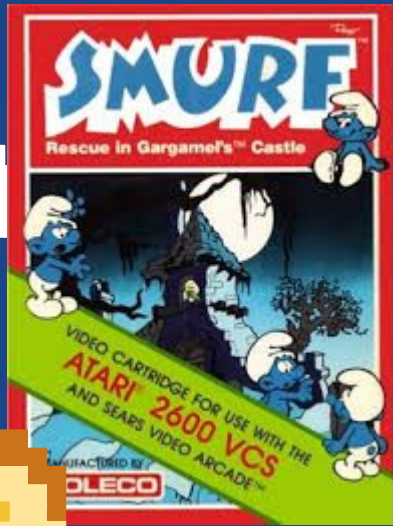


Hack/Remake

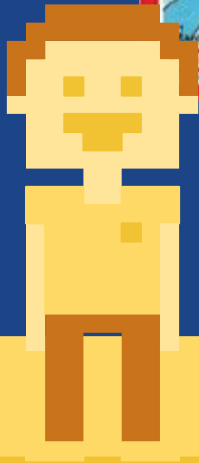
Smurf: Rescue from Gargamel's Castle



Sumin Byun
Stephanie Kim



01 HACK





Hack



Changing color of pants

1. Found where the code was determining the white color.
→ found where a block of the lines was “BLACK”, the next block was “BLUE”, and the next block was “BLACK” but with the bytes ‘0e’
2. Replace with alternating pink and orange to create striped pants.



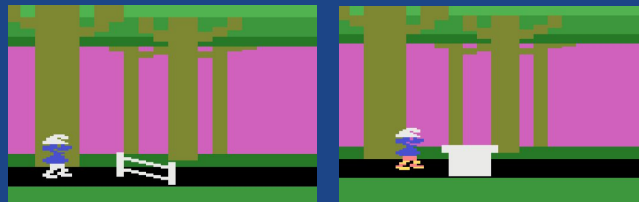
Now, our smurf character has a new fashionable look!



Changing obstacle

1. The fence was drawn in Bank #1, in binary.
2. Wanted to change the sprite's bitmap to replace the fence with a warp pipe from Mario Bros.
→ Changed binary values to change the shape of obstacle. The warp pipe was drawn in the stella debugger at the same location the fence was in.

Now, the smurf game has a fun reference to Nintendo's Mario franchise!





Hack



Changing number of lives

1. The original game starts with 4 lives
→ quite difficult, especially with more enemies attack the smurf as the game goes on
2. In the RAM widget, found "lives" counter in: \$00b6
3. Found corresponding line in code
4. Replaced it with the no operation NOP opcode, "\$EA"

*Now, even though we get killed, we still see 4 lives.
We have infinite lives!*



Changing sound

1. Whenever the player walks or jumps, there's sound.
→ walking sound: too low-pitched, wanted to make it higher
→ jumping sound: too short, wanted to make it longer
2. Located the AUDF1 – controls the frequency (pitch)
3. Found where corresponding register (register A) was being loaded into and changed dd → d3.
4. This changed the frequency and length of walking and jumping sound.

Now we have more interesting walking and jumping sound from our smurf!





Key Points of Game

Original Game Essence:

- Navigating obstacles to rescue Smurfette (movement mechanics)
- Timer aspects
- Smooth walking and jumping cycle
- Colorful, detailed maps and sprites



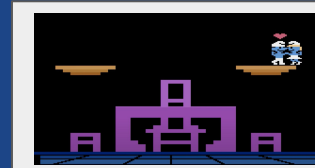
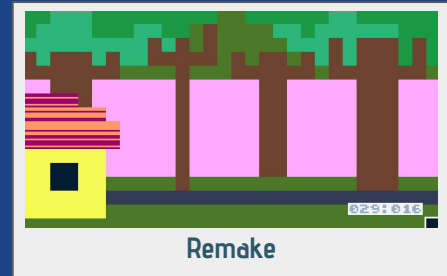
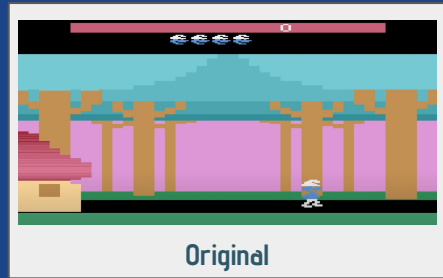
Original



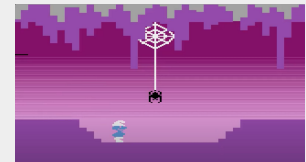
Remake

Our Approach:

- Physics
- Collision detection
- Timer library 
- Maps & Sprites recreation



Original



Remake





02

REMAKE



<https://github.swarthmore.edu/pages/CS91S-F24/remake-sbyun1-skim9/ga>
[me/](#)

Sprites:

- Walking (3)
- Jumping
- Collided
- Squatting
- Meet with Smurfette

Map:

- 6 maps based on Smurf's x-position
- 10s per map, shown with countdown bar, doesn't count repeats
- Set ground-levels for each section of maps
- Instruction screen, game over screen

Design





Physics



Jumping State:

- Up button once: jump up
- Up button twice within 1 sec: jump forward
- Uses Atari's one button to work as two

Jumping Physics:

- Jump up: y velocity
- Jump forward: x velocity & y velocity
- Gravity = 0.2

Moving Left/Right:

- Subtract from/add to Smurf's x-position

```
--jumping state of the smurf
local Button = Enum("unpressed",
"single", "double")
        :
        :
        if self.pressed > 0 and
btnp(0) and (t-self.pressed) <
1000 and self.vy==0 and
self.alive == 1 then
            press_type = Button.double
            self.pressed = t
        elseif btnp(0) and self.vy==0
and self.alive == 1 then
            self.pressed = t
            press_type = Button.single
```




Physics

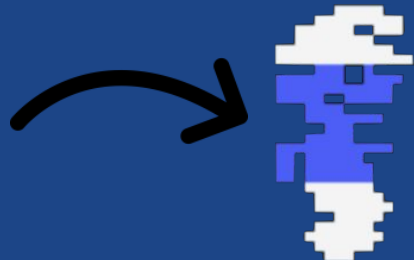


```
if press_type==Button.double then
  --jump forward
  if self.faceflag == 0 then
    self.vx = 1.75
    self.vy = -3.5
  elseif self.faceflag == 1 and self.v.x >=128 then
    self.vx = -1.75
    self.vy = -3.5
  end
end
```

```
elseif press_type == Button.single then
  --jump up
  self.vy = -3
```

```
elseif self.v.y >=
self.ground_level then
  --stop on ground
  self.vy=0
  self.v.y = self.ground_level
  self.vx = 0
else
  --come back down
  self.vy = self.vy +
```

self.gravity



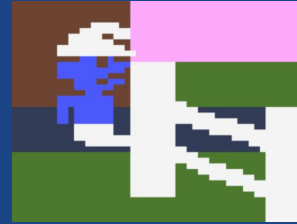


Collision Detection



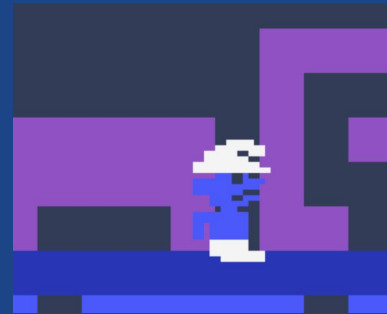
Collision:

- Smurf collides with obstacle (collision tile)



Real Collision:

1. Unhappy Smurf sits at obstacle
2. Lose one life
3. Reset countdown bar
4. Reposition Smurf at start of map after 2 seconds



Fake Collision:

- If collision occurs within 0.05s of previous collision



Overview

Accomplishments:

- Physics: accurate movements of Smurf
- Collision/Score: correct obstacles detection (+400 points once), accurate lives count (4)
- Graphics: 6 maps with several animation sprites, Smurf only goes to allowed areas

Limitations:

- Collision: unable to reposition after collision in the air
 - spider on the ground
- Graphics: limited color palette, less detail
- Ratio: less space, put the scoreboard on background



Easter Egg: quickly walk down the stairs on the spider scene!



References



- <https://github.com/Skaruts/Lua-Enum/blob/master/enum.lua> (enum file)
- https://www.youtube.com/watch?v=L8Q0bMHccko&ab_channel=Tamara0%27Malley (jumping)
- <https://stackoverflow.com/questions/33510736/check-if-array-contains-specific-value> (tile collision)
- <https://github.com/swarthmore.edu/gist/kohara2/9daf0b437cb874a1e2d01dd1c4d32c09> (enumerated types)
- <https://github.com/swarthmore.edu/CS91S-F24/state-flyweight/blob/main/timer.lua> (timer)
- <https://tic80.com/learn> (TIC-80 functions)

