GD50

Lecture 5: Legend of Zelda

Colton Ogden
cogden@cs50.harvard.edu

David J. Malan
malan@harvard.edu


IT'S DANGEROUS TO GO
ALONE! TAKE THIS.


LEVEL-4
X48
B
A
-LIFE-
X1
X0

Topics

- Top-Down Perspective
- Infinite Dungeon Generation
- Hitboxes/Hurtboxes
- Events
- Screen Scrolling
- Data-Driven Design

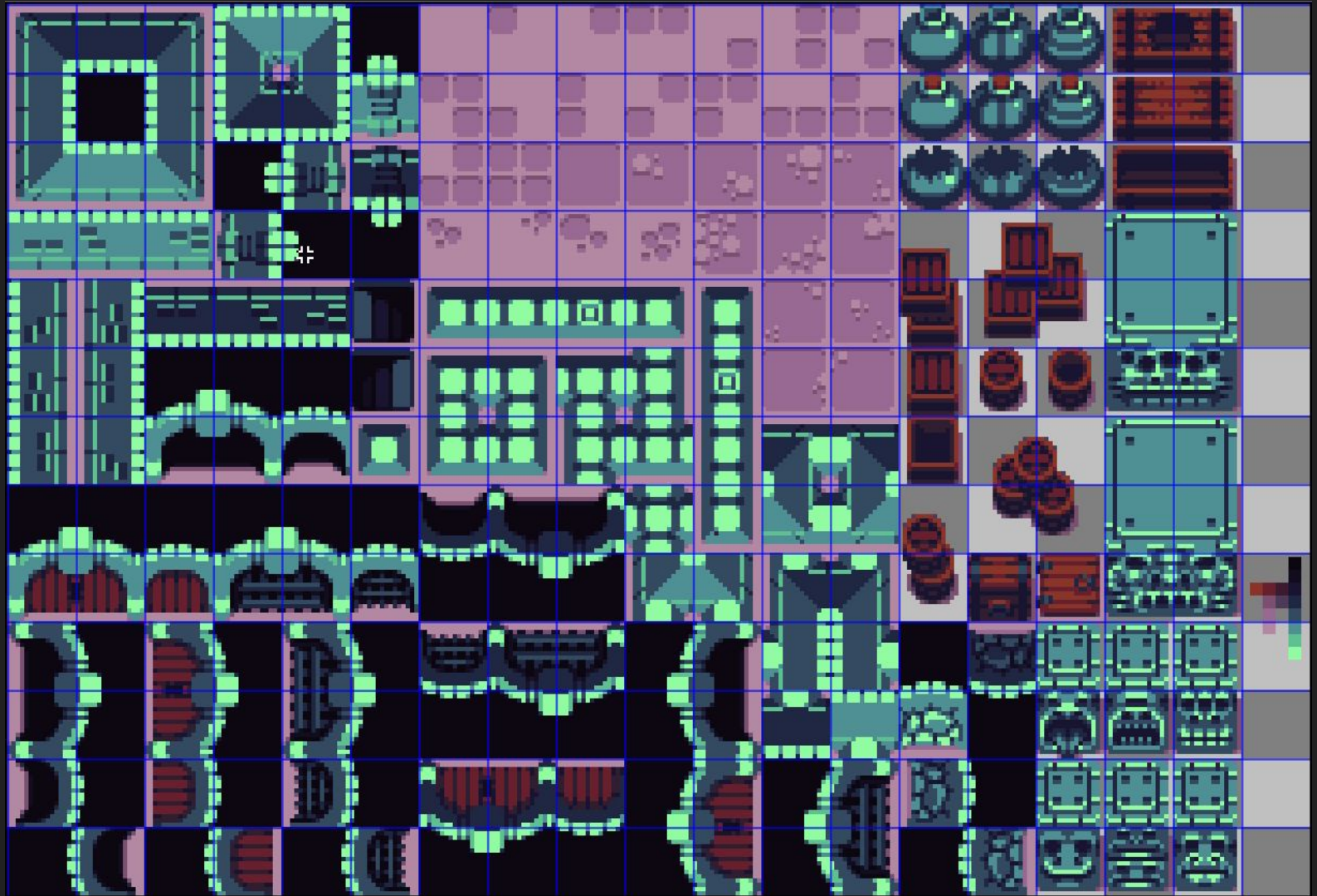
But first, a demo!

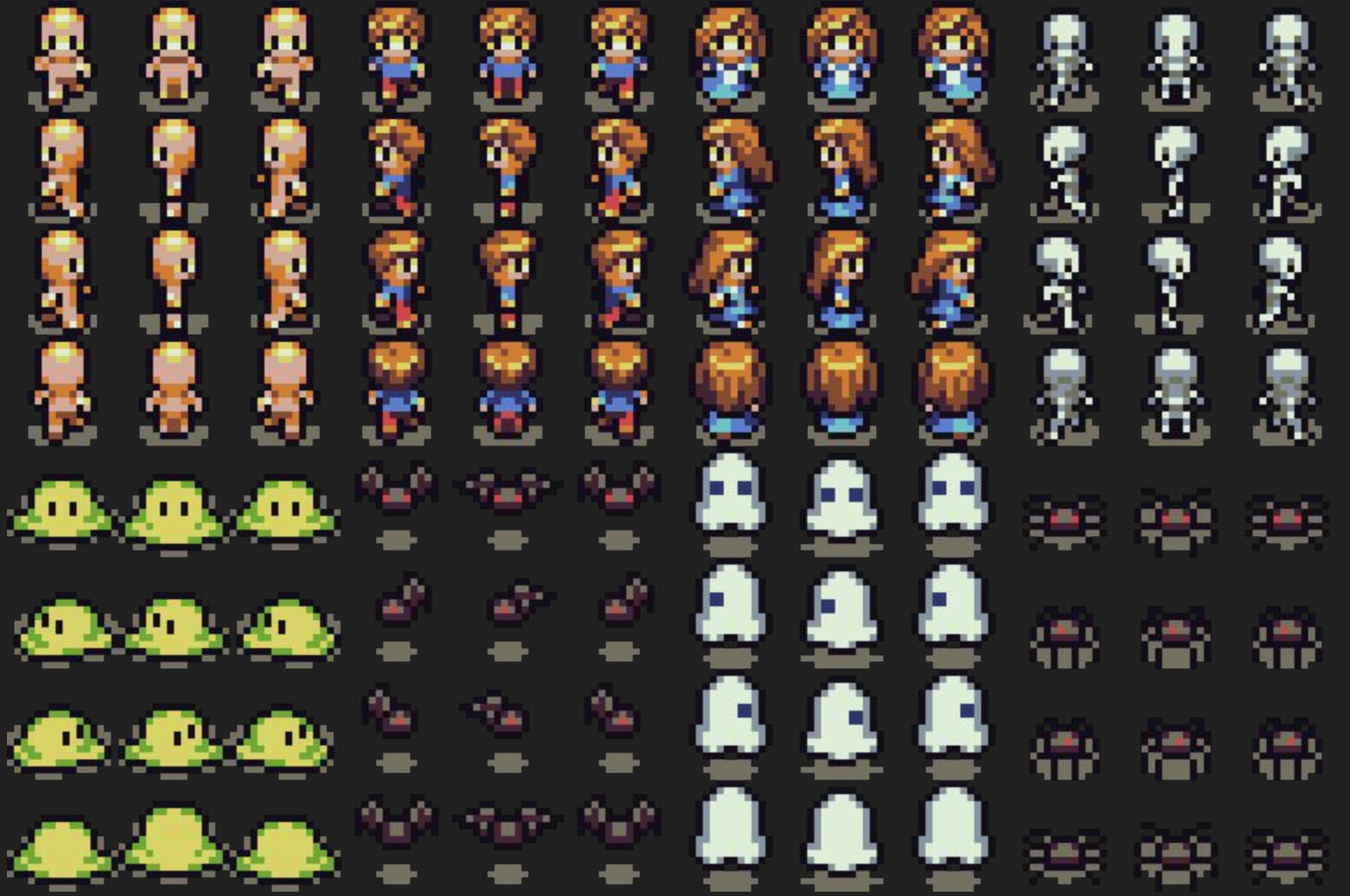
Our Goal


Legend of Zelda
LEGEND OF SO
PRESS ENTER


Legend of Zelda


Legend of Zelda
GAME OVER
PRESS ENTER

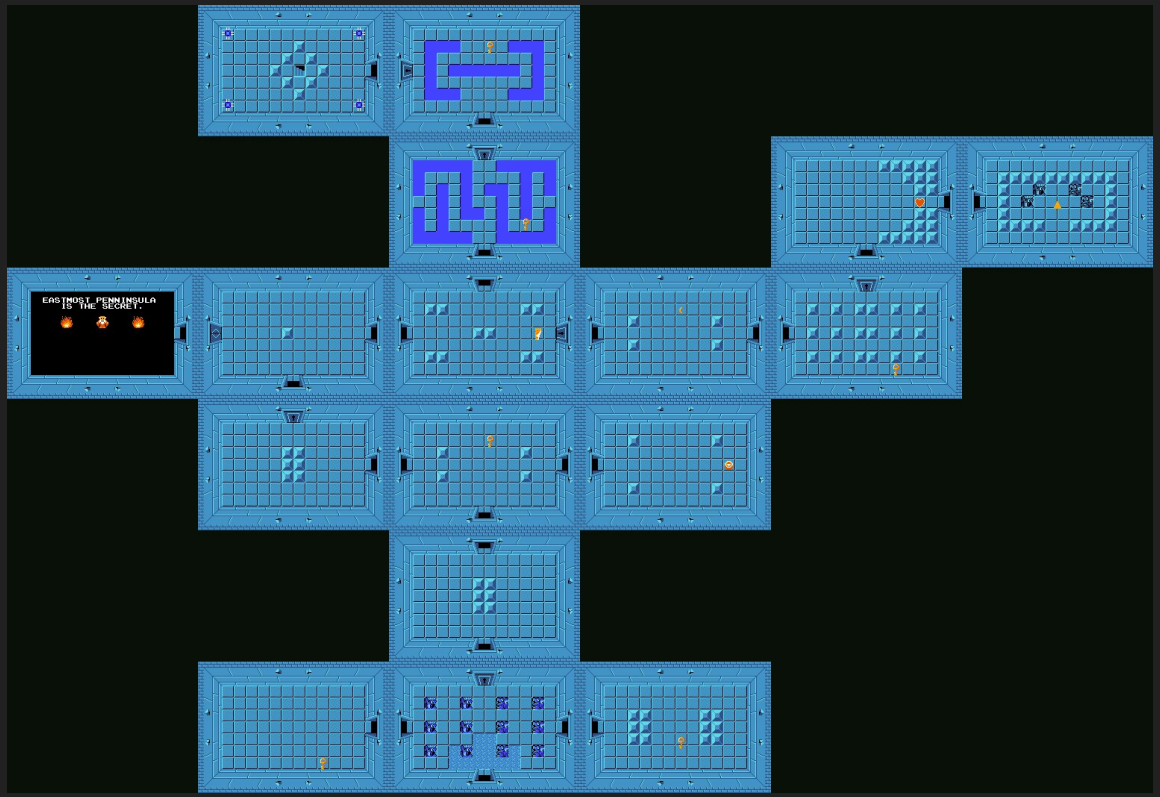





1 2 3 4 5 6 7 8 9 10 11 12
13 14 15 16 17 18 19 20 21 22 23 24
25 26 27 28 29 30 31 32 33 34 35 36
37 38 39 40 41 42 43 44 45 46 47 48
49 50 51 52 53 54 55 56 57 58 59 60
61 62 63 64 65 66 67 68 69 70 71 72
73 74 75 76 77 78 79 80 81 82 83 84
85 86 87 88 89 90 91 92 93 94 95 96

Top-Down Perspective

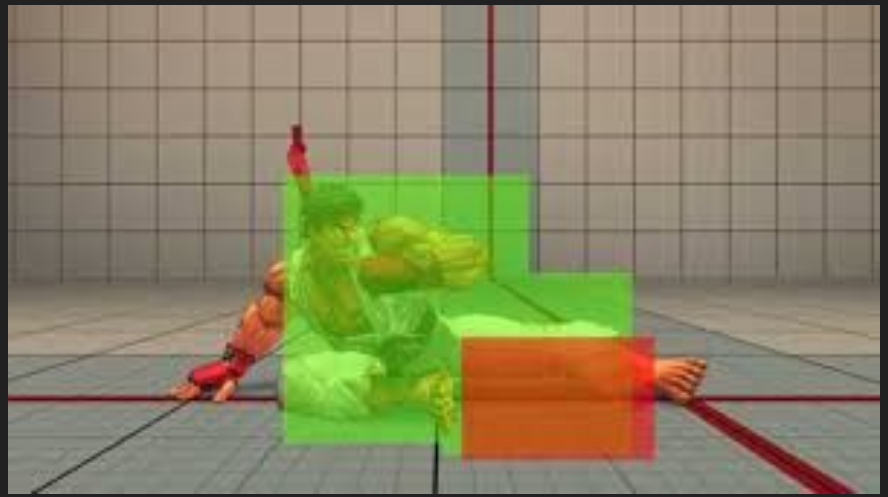
Dungeon Generation

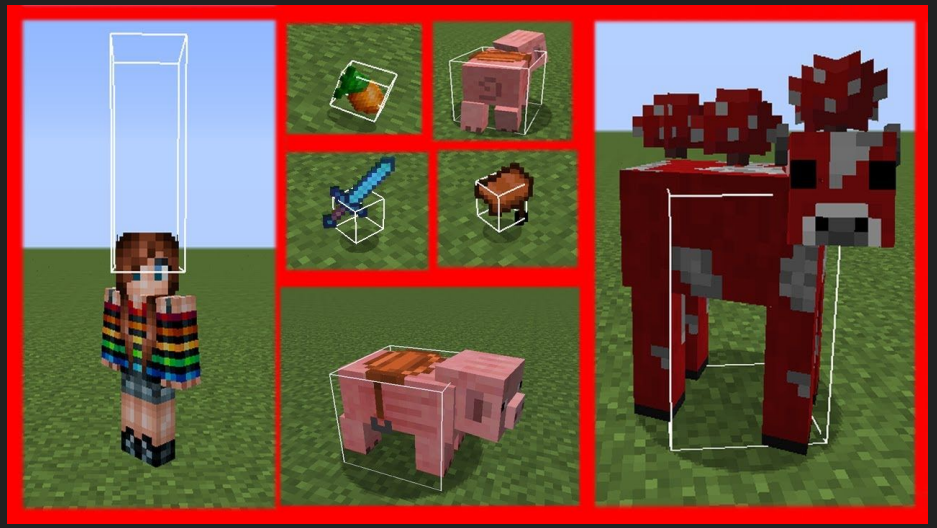


http://tartarus.rpgclassics.com/zelda1/1stquest/dungeonmaps.shtml


02
05
00

Hitboxes/Hurtboxes





Events

- An event is registered to trigger via some name, implemented via an anonymous function.
- Something in the game warrants the event being triggered, or “dispatched”.
- The anonymous callback function tied to the Event, the handler, is passed arguments via the event’s dispatch.

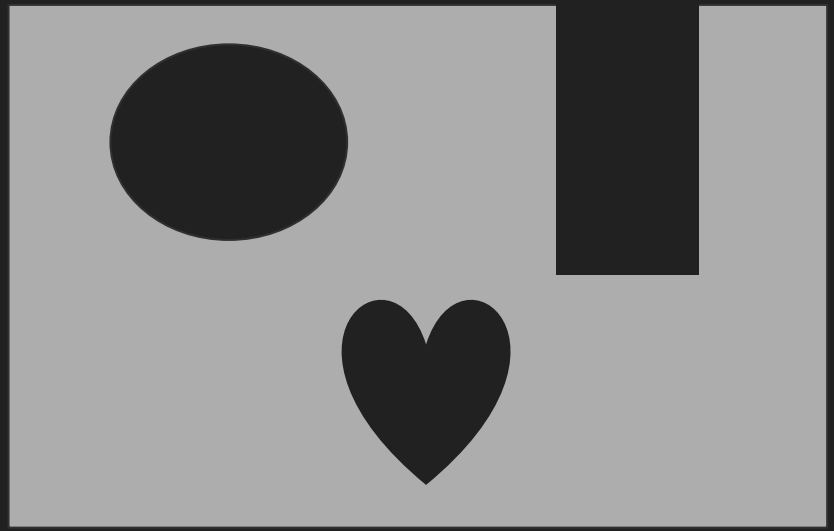
Event library: functions

- **Event.on(name, callback)**
  - Calls `callback`, which is a function, whenever the message by its `name` is dispatched via `Event.dispatch`.
- **Event.dispatch(name, [params])**
  - Calls the callback function registered to `name`, set by `Event.on`, with some optional `params` that will be sent to that callback function as arguments.

https://github.com/airstruck/knife/blob/master/readme/event.md

Screen Scrolling

Stenciling



Stenciling functions

- **love.graphics.stencil(func, [action], [value], [keepvals])**
  - Performs all stencil drawing within `func`; anything drawn during that time will act as the stencil pixels during `love.graphics.setStencilTest`. `action` defines how those pixels will behave with pixels drawn onto them during `love.graphics.setStencilTest`, while `value` is the value `action` is reliant upon.
- **love.graphics.setStencilTest(compare_mode, compare_value)**
  - Compares pixels drawn via `compare_mode` with that of `compare_value`, only drawing pixels whose result of this mode is true.

Game Design via Data

```
['goblin'] = {
   health = 10,
   strength = 2,
   texture = 'goblin',
   animations = {
      ['idle'] = {
         frames = {1},
         interval = 1
      },
      ['walking-left'] = {
         frames = {2, 3, 4, 2},
         interval = 0.2
      }
   },
   weapon = 'club',
   aggressive = true,
   sleepsAtNight = true,
   flammable = true
}
```

NES Homebrew

- http://wiki.nesdev.com/w/index.php/Nesdev_Wiki
- http://wiki.nesdev.com/w/index.php/Programming_guide
- http://wiki.nesdev.com/w/index.php/Installing_CC65

Super Mario Bros. Disassembly

```
5552  ;-------------------------------------------------------------------------------
5553  ;$07 - used to hold upper limit of high byte when player falls down hole
5554
5555  AutoControlPlayer:
5556        sta SavedJoypadBits         ;override controller bits with contents of A if executing here
5557
5558  PlayerCtrlRoutine:
5559              lda GameEngineSubroutine    ;check task here
5560              cmp #$0b                    ;if certain value is set, branch to skip controller bit loading
5561              beq SizeChk
5562              lda AreaType                ;are we in a water type area?
5563              bne SaveJoyp                ;if not, branch
5564              ldy Player_Y_HighPos
5565              dey                         ;if not in vertical area between
5566              bne DisJoyp                 ;status bar and bottom, branch
5567              lda Player_Y_Position
5568              cmp #$d0                    ;if nearing the bottom of the screen or
5569              bcc SaveJoyp                ;not in the vertical area between status bar or bottom,
5570  DisJoyp:    lda #$00                    ;disable controller bits
5571              sta SavedJoypadBits
5572  SaveJoyp:   lda SavedJoypadBits         ;otherwise store A and B buttons in $0a
5573              and #%11000000
5574              sta A_B_Buttons
5575              lda SavedJoypadBits         ;store left and right buttons in $0c
5576              and #%00000011
5577              sta Left_Right_Buttons
5578              lda SavedJoypadBits         ;store up and down buttons in $0b
5579              and #%00001100
5580              sta Up_Down_Buttons
5581              and #%00000100              ;check for pressing down
5582              beq SizeChk                 ;if not, branch
5583              lda Player_State            ;check player's state
5584              bne SizeChk                 ;if not on the ground, branch
5585              ldy Left_Right_Buttons      ;check left and right
5586              beq SizeChk                 ;if neither pressed, branch
5587              lda #$00
5588              sta Left_Right_Buttons      ;if pressing down while on the ground,
5589              sta Up_Down_Buttons         ;nullify directional bits
```

https://gist.github.com/1wErt3r/4048722

Assignment 5

- Make some enemies drop hearts randomly, which heal the player for 2 damage (one whole heart).
- Allow the player to lift pots (using the animation included in the sprite sheet).
- Pots should stick to the player while they are carrying them, and their walking animations should change while carrying.
- Allow the player to throw pots and damage enemies. If it hits a wall, an enemy, or travels farther than four tiles, destroy it.

Next Time...

https://opengameart.org/content/physics-assets

See you next time!