17 Program Design

Modular Development
Testing
Demo Creating Full Program
(Review) Coding Teaches You How to Think

Cutts et al., The Abstraction Transition Taxonomy
What have we’ve learned so far?

Drawing
.
.
.


What have we’ve learned so far?

Drawing
Variable
Interaction
Conditional
What have we’ve learned so far?

Drawing
Variable
Interaction
Conditional
Rectangle Hit Test
Circle Hit Test
For Loop
While Loop
Nested Loop
Functions
Our programs are getting complicated

- Code is getting longer...
Our programs are getting complicated

- Code is getting longer...

What should we do?
Our programs are getting complicated

- Code is getting longer...

What do we do when we see a huge mess?
Put similar things into boxes to organize
Modularize

Put similar **code** into **modules** to organize
Modular Programming Techniques

- **Idea**: “Let’s be organized!”
Modular Programming Techniques

- How?

#1. Incremental
- start simple and build up
- focus on **functionality** first, not design/graphics ("wireframing")
Modular Programming Techniques

- How?

#2. Modular
- break the program into parts, and code each part individually

```javascript
function draw() {
    // detect key press
    if (keyIsPressed) {
    }

    // detect mouse press
    if (mouseIsPressed) {
    }
}

function keyPressed() {
}

function mousePressed() {
}
```

```javascript
function draw() {
}

function keyPressed() {
}

function mousePressed() {
}
```
Modular Programming Techniques

- How?

#2. Modular
- break the program into parts, and code each part individually
- “parameterize” each part so it’s easy to combine later
  
  “to represent in terms of a parameter”
  (i.e. functions with parameters)
Modular Programming Techniques

- How?

#2. Modular
- break the program into parts, and code each part individually
- “parameterize” each part so it’s easy to combine later

```javascript
function tree(x, y) {
  // use the tree function to
  // create a forest
  tree(40, 90);
  tree(100, 90);
  tree(160, 90);
}

// draw tree
// with bottom of trunk at x, y
```
Modular Programming Techniques

- How?

#2. Modular
- break the program into parts, and code each part individually
- “parameterize” each part so it’s easy to combine later

```javascript
strokeJoin(join)
// where join = BEVEL, MITER, or ROUND

strokeCap(cap)
// cap = SQUARE, PROJECT, or ROUND
```
Modular Programming Techniques

How?

#2. Modular
- break the program into parts, and code each part individually
- “parameterize” each part so it’s easy to combine later
Modular Programming Techniques

▪ How?

#3. Testing
- decide what is correct behaviour and what isn’t (requirements)
- think of all different kinds of input you could have
- think of all different kinds of program states you could have
- test each part separately ("unit testing")

```javascript
function myFunction(value1) {
}
```

will this work for...
- 0?
- -2?
- 10?
- 10000?
- 10.4?
Space Invaders | Design Icons

- [https://www.youtube.com/watch?v=Jbn8IRmSq8M](https://www.youtube.com/watch?v=Jbn8IRmSq8M)
Space Invaders

http://www.tripletsandus.com/80s/80s_games/html5_SpaceInvaders.htm
Example: Space Invader

- Keys move a ship side-to-side along bottom of screen.
- An alien moves left and right and steadily down
- You lose if the alien reaches the ship
- Fire one laser burst at a time. Can’t fire again until the laser burst is off screen or hits the alien.
- You get a point if the laser burst “hits” the alien
Space Invaders: Parts?

http://www.tripletsandus.com/80s/80s_games/html5_SpaceInvaders.htm
Space Invaders: Parts?

http://www.tripletsandus.com/80s/80s_games/html5_SpaceInvaders.htm
Space Invader Parts

- function to move and draw **alien**
  - function to draw alien

- function to update and draw **ship**
  - function to draw ship
  - functions to move ship left and right

- function to move and draw **laser**
  - function to draw laser
  - function to fire laser

- laser-to-alien **hit test** function (rectangle hit test)
Space Invader Modules

- function to move and draw alien
  - function to draw alien

- function to update and draw ship
  - function to draw ship
  - functions to move ship left and right

- function to move and draw laser
  - function to draw laser
  - function to fire laser

- laser-to-alien hit test (rectangle hit test)

Each part can form a module
Writing Modular Code

▪ use custom user **functions**
▪ use built-in **variables** like width and height
▪ use **unique** global **variable** names
  - e.g. shipX, shipWidth, alienX, alienWidth, ...
▪ add local **variable** “stubs” to represent other modules
  - e.g. mouseX and mouseY in hit test module
▪ separate functionality from visual design
  - e.g. ship() has functionality, drawShip() has visual design

Write each “**module**” separately to simplify problem and to test, then combine modules later
spaceinvader (outline)

each frame of draw:

// clear background
// display score (and game over message)
// move and draw the laser (if fired)
// move and draw the ship
// move and draw the alien
// if the alien reaches the bottom, you lose
// check if laser hits alien

game input will be:

// "a" move left
// "d" move right
// SPACE " " fire laser

https://editor.p5js.org/cs105/sketches/D-PAP5YLd
ship-module

let shipWidth = 40;
let shipHeight = 20;
let shipX;
let shipY;
let shipSpeed = 5;

function shipUpdate() {
  // keep the ship in canvas
  // draw the ship
  shipDraw();
}

function shipMoveLeft() {}
function shipMoveRight() {}
Ship Testing

- Does it move horizontally with keys?
- Does it stay inside canvas?
  - if canvas width is changed?
  - if ship width is changed?
let alienX;
let alienY;
let alienSpeed = 5;
let alienDirection = alienSpeed;
let alienWidth = 64;
let alienHeight = 64;

function alienUpdate() {
  // move the alien
  // reverse alien and move down when hits sides
  // draw the alien
  alienDraw();
}
Alien Testing

- Does it stay inside canvas?
- Does it move down each time it hits left or right?
- Do you lose if it reaches the bottom?
laser-module

let laserX = 100;
let laserY = 0;
let laserHeight = 10;
let laserSpeed = 3;
let laserFired = false;

function laserUpdate() {
    // move the laser *up* if it was fired
    // check if it goes off screen
    // draw the laser if it was fired
    laserDraw();
}

function laserFire(x, y) {}
Laser Testing

- Can fire only one laser at a time?
- Does it keep the trajectory after it’s fired? (doesn’t move with mouse/ship?)
hittest-module

// returns true if px, py is inside rectangle centred
// at location x,y with width w and height h
function hitTest(px, py, x, y, w, h) {
    if (px >= x - w/2 && px <= x + w/2 &&
        py >= y - h/2 && py <= y + h/2) {
        return true;
    } else {
        return false;
    }
}
Hittest Testing

- Does mouse cursor trigger the hit test?
- Does it work for different sizes and positions?
Merging Completed Modules into One Program

- Merge modules one-by-one
  - test after each merge
- Merge modules with “stubs” after dependent modules
  - hittest module should be merged after alien and laser
- Integrate modules where necessary
  - keyPressed
  - move ship to bottom
  - hittest laser with alien, and reset if hit
- Add in game graphics to drawing “stub” functions
spaceinvader

merge order:

ship 🡪 laser 🡪 alien 🡪 hittest

integrate alien and laser with hit test

add in game graphics
update ship and alien size parameters

https://editor.p5js.org/cs105/sketches/uzrhwOivX
Possible enhancement ideas

- keep track of score
- ?