Writing Conditions

On-the-Spot Decisions	decision about which alternative is based only on the current value of animation or system variable(s)

- How many alternatives are there? → if structure
- Is "do nothing" an option?
  - "to do or not to do" (do something or do nothing) → if
  - two alternatives → if else
  - more than two alternatives
    - "do nothing is not an option" → if / else if / ... / else
    - "do nothing" is an option → if / else if / ...
- What are the alternatives?   → body of each part
- When does each alternative occur?
                             → condition for each part

Prior Happenings Decisions	decision about which alternative depends on prior events / what has been going on

- What is the decision about?   → state variable name
- How many alternatives are there? → state variable type
- Is "do nothing" an option?   → if structure
- Which alternative do we start with?
                              → state variable initialization
- What are the alternatives? → body of each part (usage if)
- For each alternative, when do we change to that alternative?   → condition for each part (update if)

Conditions are *boolean expressions* – something which evaluates to true or false.

Tools –
- *relational operators* ==, !=, <, <=, >, >= for comparing two values
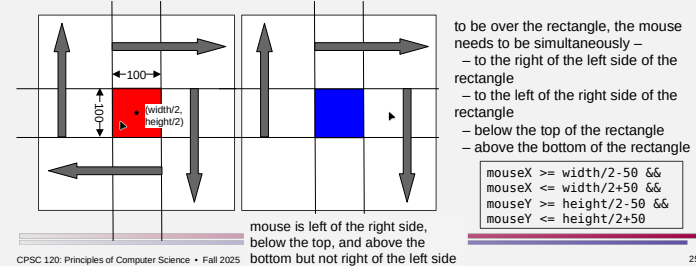- *logical operators* &&, ||, ! for combining simple conditions into more complex conditions

A B	!A	A && B	A || B
T T	F	T	T
T F	F	F	T
F T	T	F	T
F F	T	F	F

&& – both parts must be true for the whole condition to be true
|| – only one part needs to be true for the whole condition to be true

---

Writing Conditions

A sketch contains a 100x100 rectangle centered in the drawing window. The rectangle should be outlined in black, and should be filled red when the mouse is over it and blue otherwise.

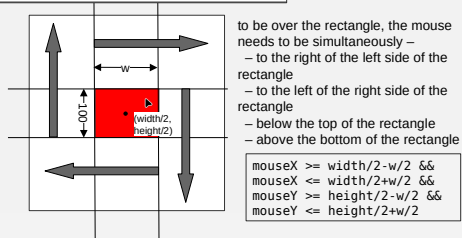| on the spot or prior happenings?	on the spot – look at the mouse position to know the color
is "do nothing" an option?	no, the rectangle must have a color
when does each alternative occur?	red when the mouse is over the rectangle, blue otherwise

to be over the rectangle, the mouse needs to be simultaneously –
 – to the right of the left side of the rectangle
 – to the left of the right side of the rectangle
 – below the top of the rectangle
 – above the bottom of the rectangle

```
mouseX >= width/2-50 &&
mouseX <= width/2+50 &&
mouseY >= height/2-50 &&
mouseY <= height/2+50
```

mouse is left of the right side, below the top, and above the bottom but not right of the left side

---

Writing Conditions

A sketch contains a 100x100 rectangle centered in the drawing window. The rectangle should grow in width when the mouse is over the rectangle, and should reset to 100 when it reaches the edge of the window. (The rectangle should remain centered in the window.)

on the spot or prior happenings?	on the spot – can just look at the mouse position
is "do nothing" an option?	yes – grow or not
when does each alternative occur?	grow when the mouse is over the rectangle, do nothing otherwise

to be over the rectangle, the mouse needs to be simultaneously –
 – to the right of the left side of the rectangle
 – to the left of the right side of the rectangle
 – below the top of the rectangle
 – above the bottom of the rectangle

```
mouseX >= width/2-w/2 &&
mouseX <= width/2+w/2 &&
mouseY >= height/2-w/2 &&
mouseY <= height/2+w/2
```

Writing Conditions

A sketch contains a 100x100 rectangle centered in the drawing window. The rectangle should grow in width when the mouse is over the rectangle, and should reset to 100 when it reaches the edge of the window. (The rectangle should remain centered in the window.)

on the spot or prior happenings?	on the spot – can just look at the current width and the window size
is "do nothing" an option?	yes – only reset at the edge of the window
when does each alternative occur?	reset when the sides of the rectangle reach the sides of the window

reset when –
 – the right side of the rectangle is past (to the right of) the right side of the window
 – and when the left side of the rectangle is past (to the left of) the left side of the window – but these things will occur at the same moment, so it is only necessary to check one

```
width/2+w/2 >= width
```

width