Decomposition

Chris Piech

CS106A, Stanford University
Today’s Goal

1. Be able to approach a problem “top down” by using decomposition (aka top town refinement)
First, a cool program
Quick review
Karel the Robot
def main():
    goToMoon()

def go_to_moon():
    build_spaceship()
    # a few more steps

def build_spaceship():
    # todo
    put_beeper()
def main():
    # repeats the body 99 times
    for i in range(99):
        # the "body"
        put_beeper()
def main():

    # while condition holds runs body
    # checks condition after body completes
    while front_is_clear():
        move()
def main():
    # If the condition holds, runs body
    if front_is_clear():
        move()
def main():

    # If the condition holds,
    if beepers_present():
        # do this
        pick_beeper()
    else:
        # otherwise, do this
        put_beeper()
# The Full Karel

## Base Karel commands:
- `move()`
- `turn_left()`
- `put_beeper()`
- `pick_beeper()`

## Karel program structures:
- Comments can be included in any part of a program. They start with a `#` and include the rest of the line.
- `def main():
  code to execute

declarations of other functions`

## Conditions:
- `if condition:
  code run if condition passes`
- `if condition:
  code block for "yes"
else:
  code block for "no"`

## Loops:
- `for i in range(count):
  code to repeat`
- `while condition:
  code to repeat`

## Names of the conditions:
- `front_is_clear()`
- `front_is_blocked()`
- `beepers_present()`
- `no_beeper_present()`
- `beepers_in_bag()`
- `no_beeners_in_bag()`
- `left_is_clear()`
- `left_is_blocked()`
- `right_is_clear()`
- `right_is_blocked()`
- `facing_north()`
- `not_facing_north()`
- `facing_south()`
- `not_facing_south()`
- `facing_east()`
- `not_facing_east()`
- `facing_west()`
- `not_facing_west()`

## Function Declaration:
- `def name():
  code in the body of the function`

## Extra Karel Commands:
- `paint_corner(COLOR_NAME)`
- `corner_color_is(COLOR_NAME)`
End review
def friday():
    # here's our plan
decomposition()
double_bEEPERS()
rhoomba_karel()
if extra_time():
    word_search_karel()
def friday():
    # here's our plan
    decomposition()
    double_bEEPERS()
    rhoomba_kareL()
    if extra_time():
        word_search_kareL()
def friday():
    # heres our plan
    decomposition()
    double_beepers()
    rhoomba_karel()
    if extra_time():
        word_search_karel()
def friday():
    # here's our plan
    decomposition()
    double_beeper()
    rhoomba_karel()
    if extra_time():
        word_search_karel()
Double Beepers

Before

After

Muhammed ibn Musa Al Kwarizmi
Do your thing

Piech, CS106A, Stanford University
Aside: Common Errors

Lather, Rinse, Repeat
def turn_to_wall():
    while left_is_clear():
        turn_left()
Infinite Loop

```python
def turn_to_wall():
    while left_is_clear():
        turn_left()
```

Piech, CS106A, Stanford University
def turn_to_wall():
    while left_is_clear():
        turn_left()
def turn_to_wall():
    while left_is_clear():
        turn_left()
def jump_hurdles:
    for i in range(8):
        if front_is_clear():
            move()
        else:
            jump_hurdle()

What do you assume here?

Does the “post condition” match?
def friday():
    # heres our plan
    decomposition()
    double beepers()
    rhoomba_karel()
    if extra_time():
        word_search_karel()
def friday():
    # here's our plan
    decomposition()
    double_b beepers()
    rhoomba_karel()
    if extra_time():
        word_search_karel()
• Write a Roomba Karel that sweeps the entire world of all beepers.
  – Karel starts at (1,1) facing East.
  – The world is rectangular, and some squares contain beepers.
  – There are no interior walls.
  – When the program is done, the world should contain 0 beepers.
  – Karel's ending location does not matter.

• How should we approach this tricky problem?
Possible Algorithm 1
Possible Algorithm 3
Possible Algorithm 4
def friday()
    banishWinter()
    decomposition()
    doubleBeepers()
    rhoombaKarel()
    if(extraTime()){
        wordSearchKarel()
def friday(){
    banishWinter()
    decomposition()
    doubleBeepers()
    rhoombaKarel()
    if(extraTime()){  
        wordSearchKarel()  
    }
def friday(){
  banishWinter()
  decomposition()
  doubleBeepers()
  rhoombaKarel()
  if(extraTime()){
    wordSearchKarel()
  }
}
Happy Friday