

CS126 Precept 6

Meeting 4: Arrays and Assignment 2

Randy Wang

Array Basics

- Initialization
- Accessing array elements
- Array out-of-bound errors
- Example codes for parallel arrays

HW 2 Suggestions

- Start early
- Get help if stuck
- Follow progress steps in the checklist (online)
- Understanding of physics not crucial

HW2 Suggestions: Pseudocode

```
Read in the input data from planets.txt into several
parallel arrays: positions, velocities, masses,
and images;

while (true) {
    step 1: calculate the net force on each body;
    step 2: calculate the new velocity and position of
            each body;
    step 3: draw each body at its current location;
}
```

HW2 Suggestions: Step 3

```
Read in the input data from planets.txt into several
parallel arrays: positions, velocities, masses,
and images;
```

```
while (true) {
    step 3: draw each body at its current location;
}
```

- Look at the `BouncingBall.java` example
- Need to translate physical coordinates to turtle coordinates:
how to convert $x=[-R, +R]$ to $turtleX=[0, 512]$?
- Don't forget `Turtle.pause(20)` at the end to have the display drawn

HW2 Suggestions: Step 2

```
Read in the input data from planets.txt into several
parallel arrays: positions, velocities, masses,
and images;
```

```
while (true) {
    step 2: calculate the new velocity and position of
            each body; (assuming zero force/acceleration)
    step 3: draw each body at its current location;
}
```

- The bodies should move off the screen in straight lines

HW2 Suggestions: Step 1

```
Read in the input data from planets.txt into several
parallel arrays: positions, velocities, masses,
and images;
```

```
while (true) {
    step 1: calculate the net force on each body;
    step 2: calculate the new velocity and position of
            each body;
    step 3: draw each body at its current location;
}
```

HW2 Suggestions: Step 1

```
Calculate the net force on each body i:
```

```
for each body i {
    initialize the net x and y forces on body i;
    for each body j {
        if body i and body j are not the same {
            calculate x and y forces exerted by body j on body i;
            add these forces to the net x and forces on body i;
        }
    }
}
```

HW2 Suggestions: Step 1

- What's all this nonsense about $dt/2$?
 - Don't worry about it
 - Everything in your program deals with dt
 - Each loop iteration spends dt
 - All formulas use dt
- Why do my bodies repel each other?
 - Make sure you have the signs right: locations, velocities, accelerations, forces