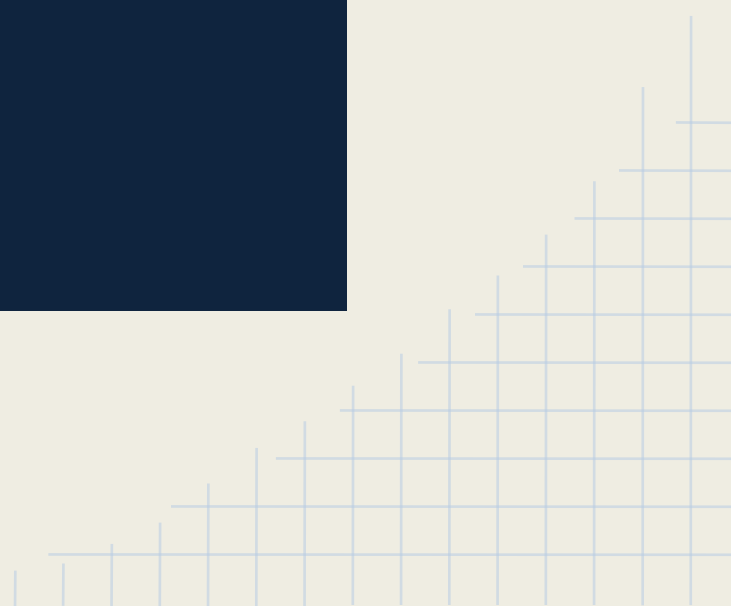


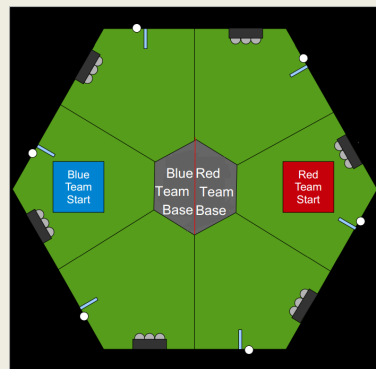
6.270 Lecture: Localization & Navigation

Jessie Zhang
2013 IAP



Overview

- Know how to drive straight, rotate, etc.
- Want to get somewhere
 - "moveToPoint(x, y)"
- Need to know where robot is in the first place

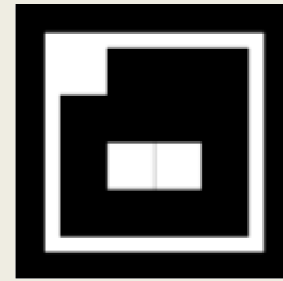


Localization

- Methods
 - Dead-reckoning
 - Gyroscope
 - Encoder shaft
 - Free wheel
 - Vision positioning system
 - Combining the two
 - In 6.270
 - In "real-life"

Localization

- Vision Positioning System (VPS)
 - Using it
 - `copy_objects()`
 - `games.coords[0]`
 - Some notes
 - Numerical values between -2048 and 2047
 - Height issue



Localization

- extern volatile uint8_t robot_id;
#define VPS_RATIO

```
int16_t current_x, current_y;  
float current_theta;
```

```
update() {  
    copy_objects();  
    current_x = games.coord[0].x * VPS_RATIO;  
    current_y = games.coord[0].y * VPS_RATIO;  
    current_theta = games.coord[0].theta / 2048 * 180; //gyro_get_degrees()  
}
```

```
void usetup(void) {  
    robot_id = ***your team number***;  
    //...  
}
```

Navigation

- Want something like "moveToPoint(x, y)"
- ```
moveToPoint(desired_point, velocity) {
 update();
 desired_heading = determineHeading(current_point, desired_point);
 rotateToHeading(desired_heading);

 while(distanceTo(current_point, desired_point) < TOLERANCE) {
 moveStraight(desired_heading, velocity);
 }

 brake();
}
```

# Navigation

- More higher level functions
- Levels of abstraction
  - Top-down / Bottom-up
- Some notes
  - Decelerating
  - Gyroscope
- Flexibility in code
- Threads

# Random Tips

- Mock competition 1
- Past team websites
- Knowing what doesn't work
- Have fun!

