

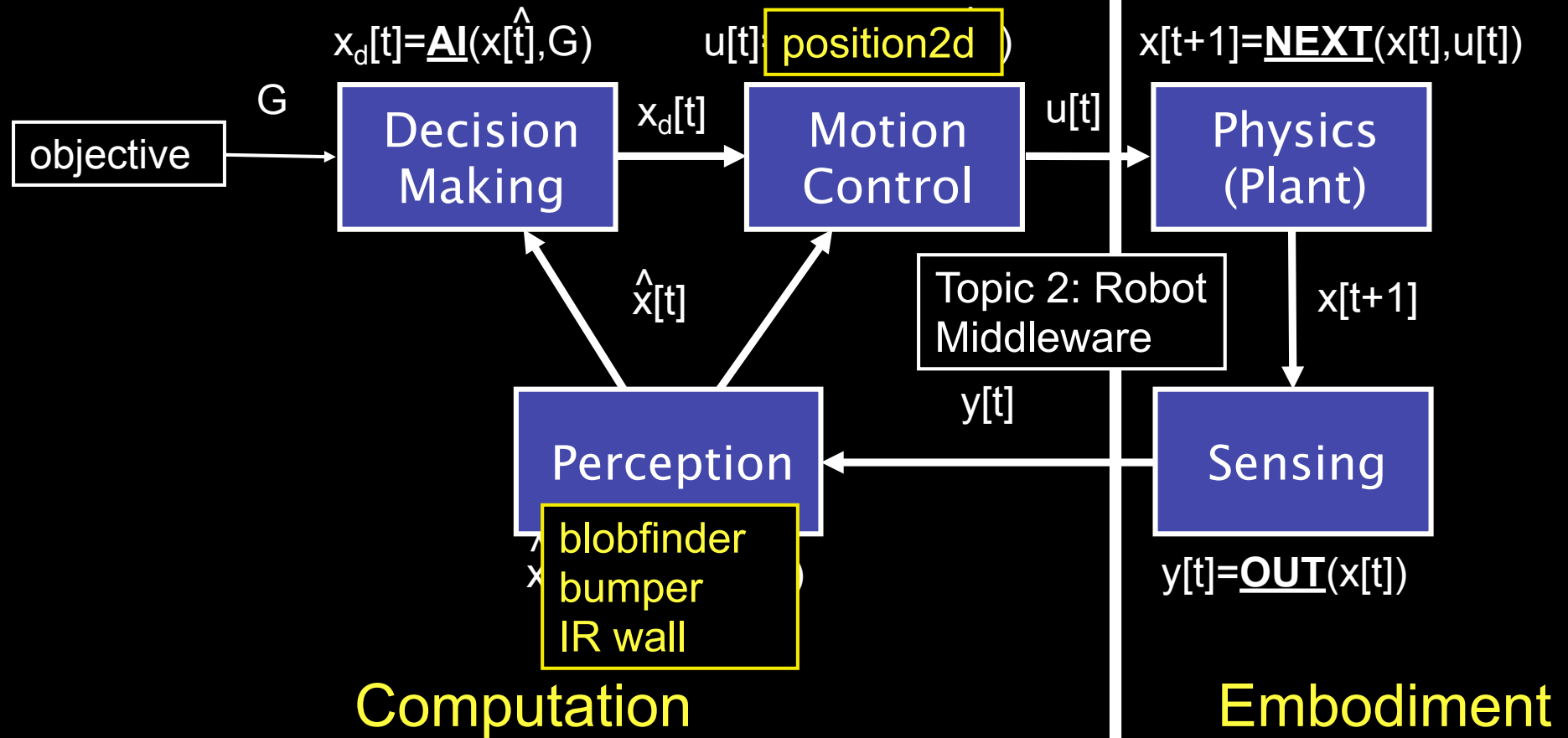
Topic 15

Multi-robot systems

# robot control loop

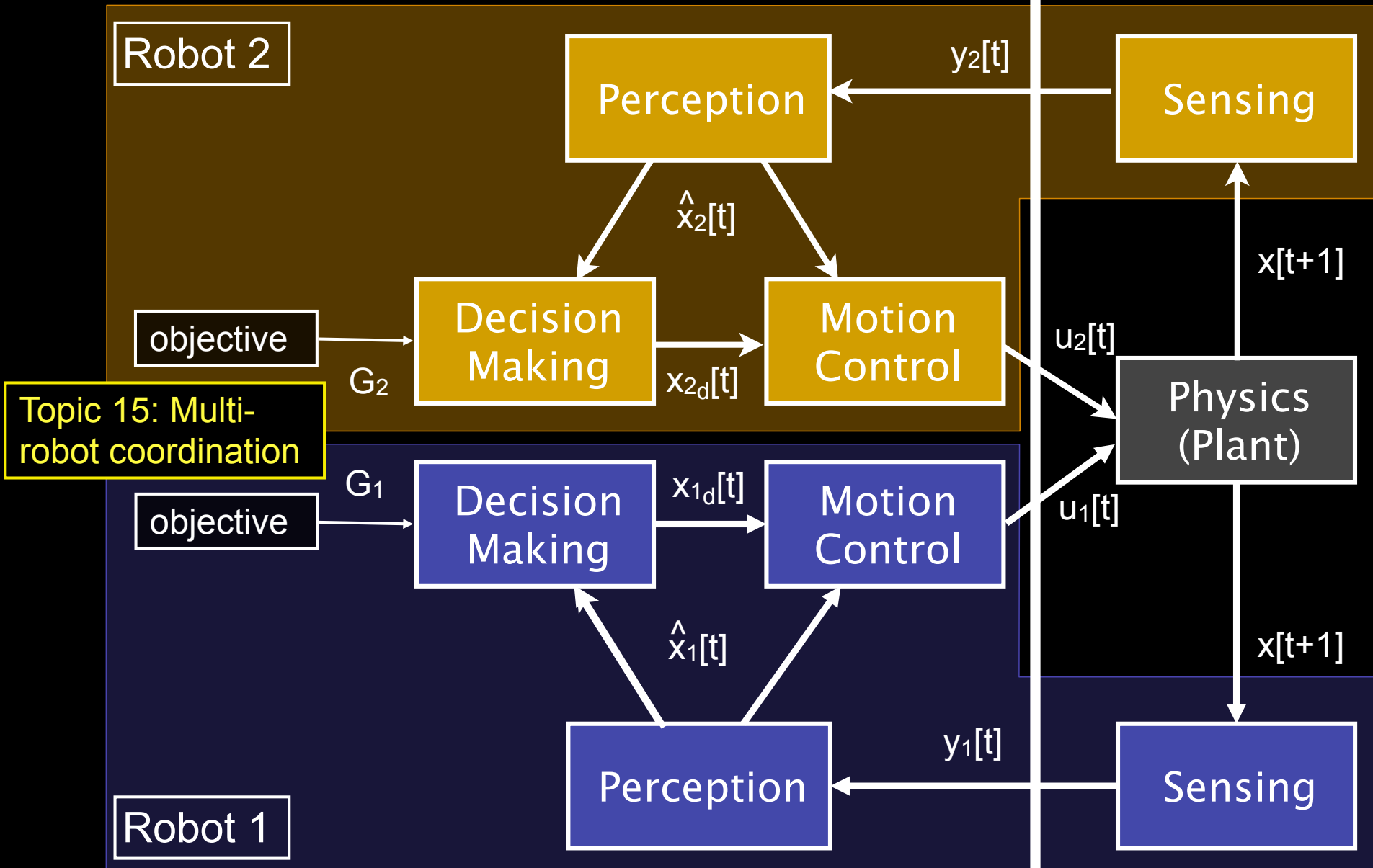
- someone please sketch on the board

How to represent and coordinate multiple collaborating robots?



# Computation

# Embodiment

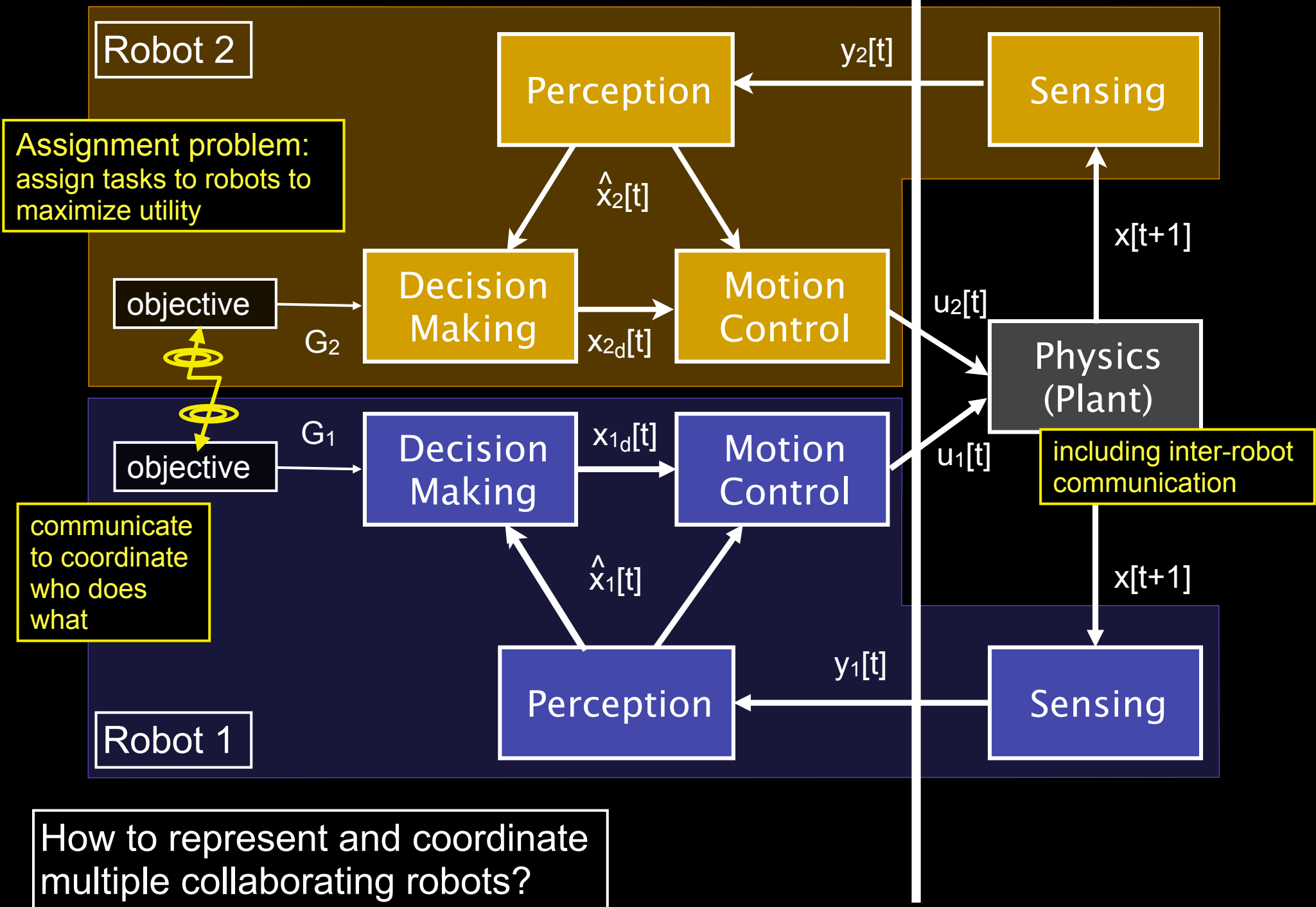


Topic 15: Multi-robot coordination

How to represent and coordinate multiple collaborating robots?

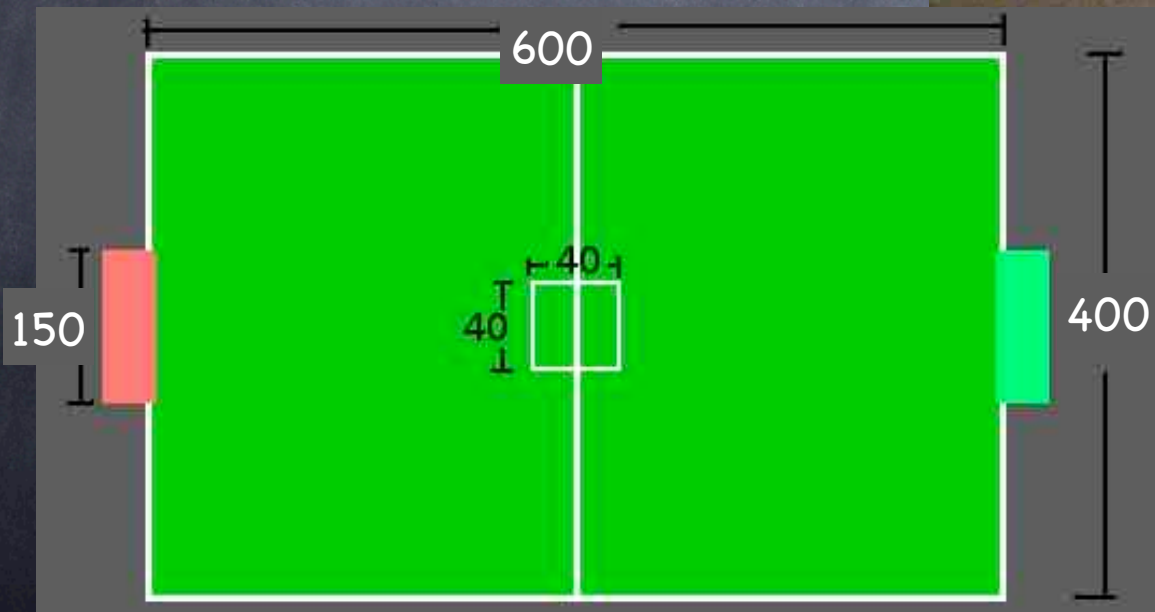
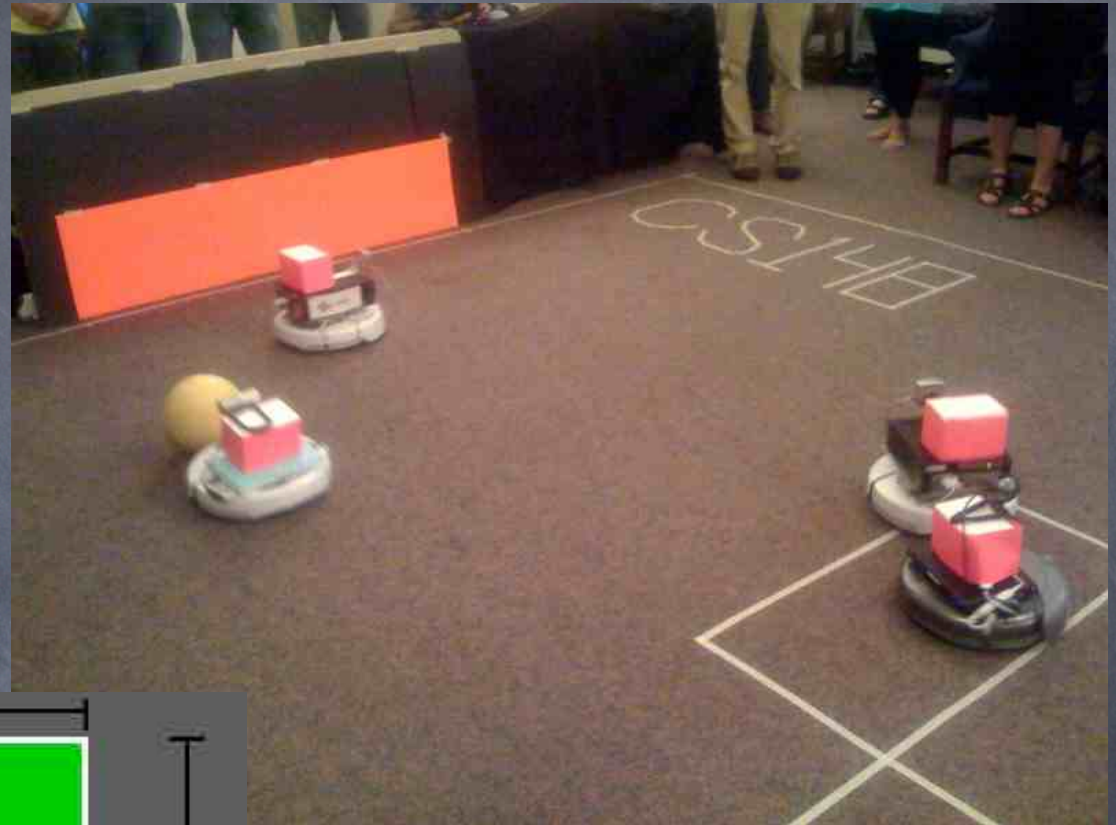
# Computation

# Embodiment



# Final Tournament

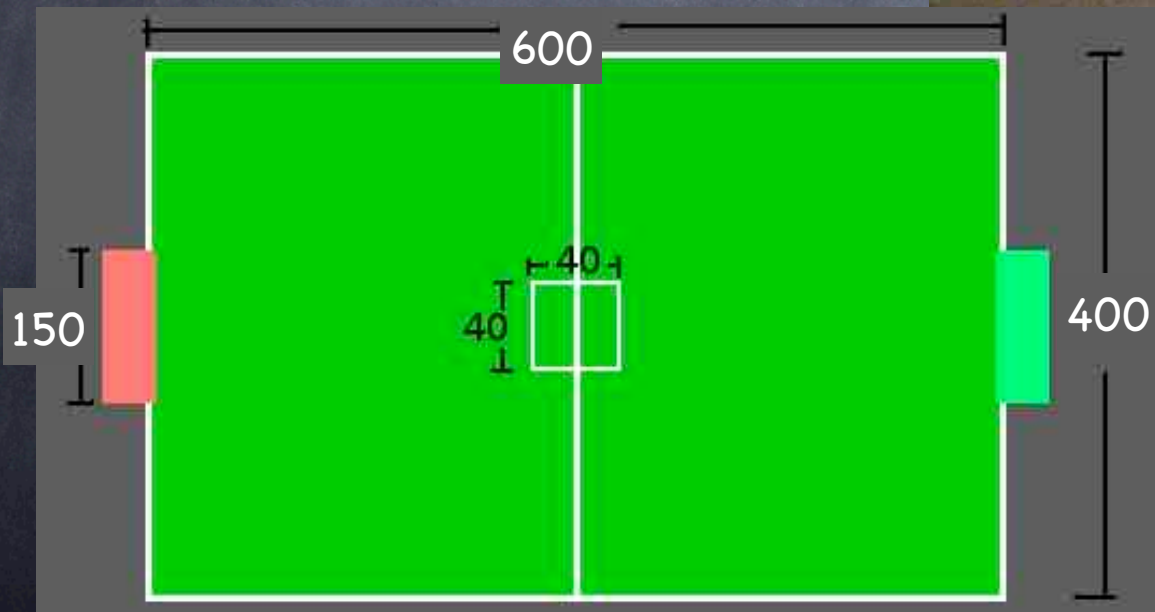
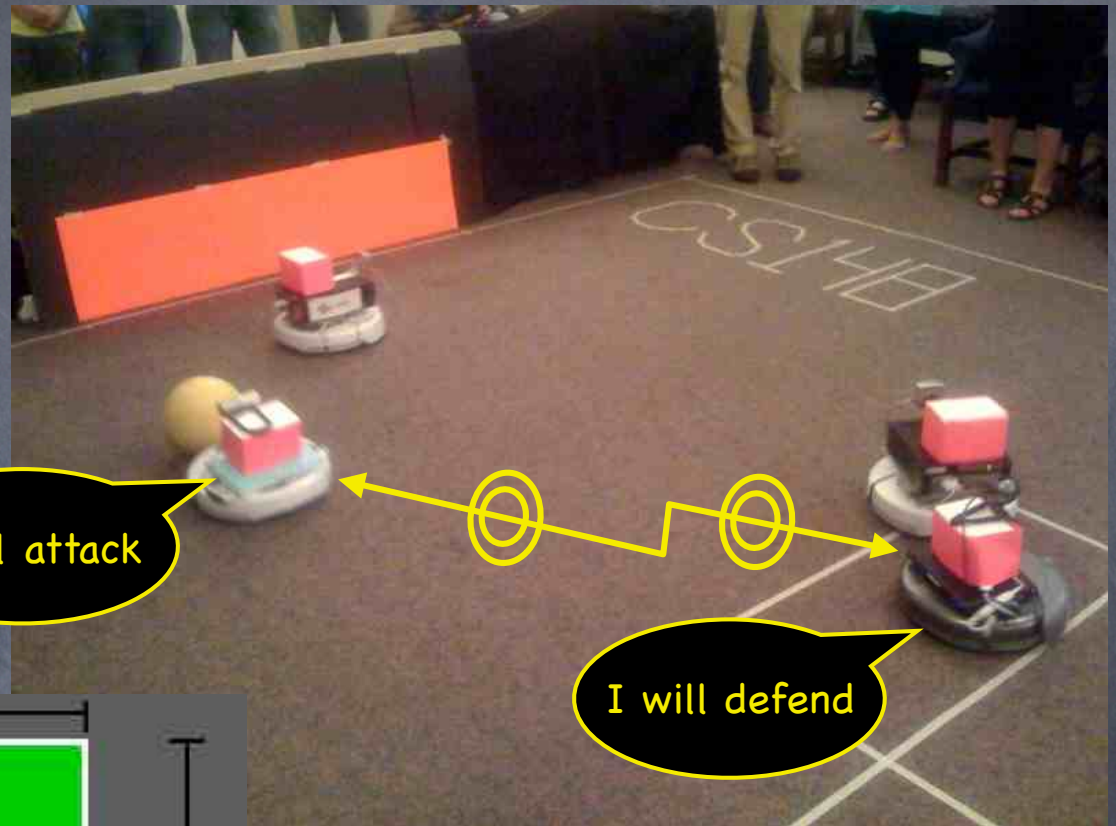
- 2-on-2 matches
- double elimination
- larger field (6x4m) and goals (1.5m)



- any on-board algorithms allowed
- hardware can be reconfigured, but not modded

# Final Tournament

- 2-on-2 matches
- double elimination
- larger field (6x4m) and goals (1.5m)



- communication and specialization allowed
- how to communicate?
- how to coordinate?

# Multi-robot coordination

## • Implicit coordination

- direct comms not necessary

- Embodied intelligence: coordination arises from dynamics of embodied interaction (SFU Create video)

- Stigmergy: coordination through modifying the environment

- Analogy to ants coordinating by pheromone trails

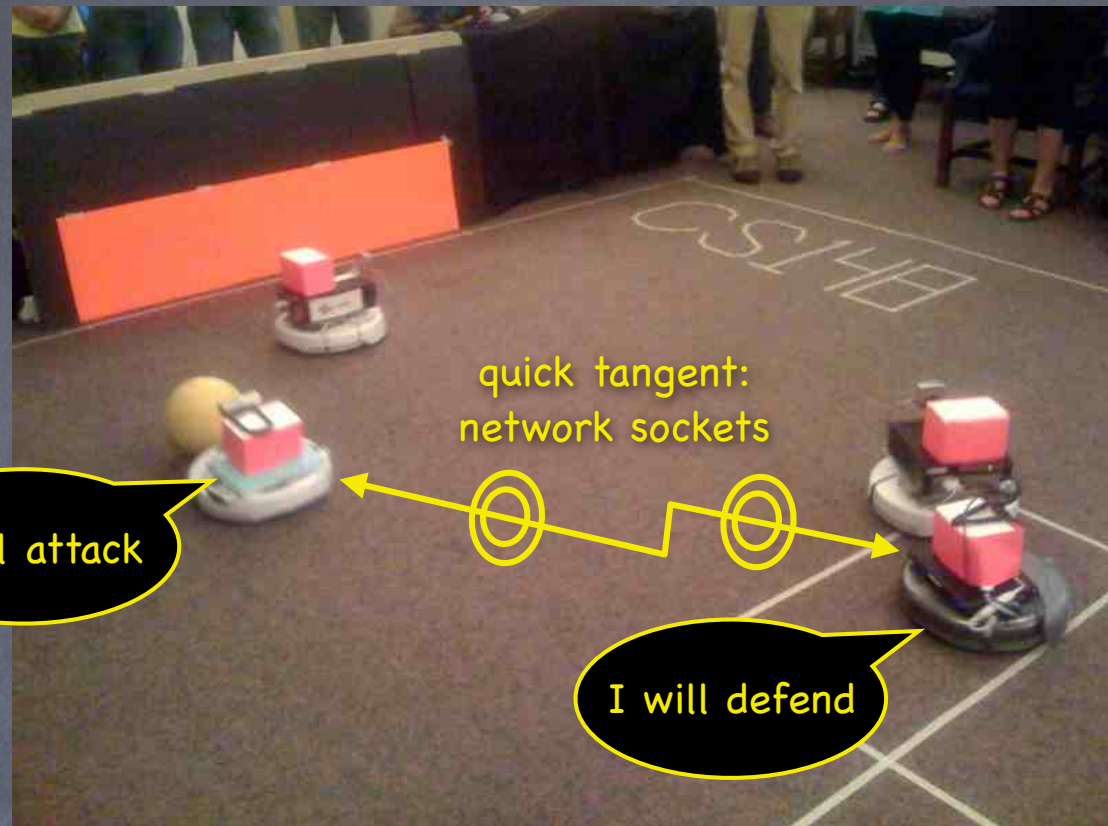


# Vaughan's 36 Creates



# Multi-robot coordination

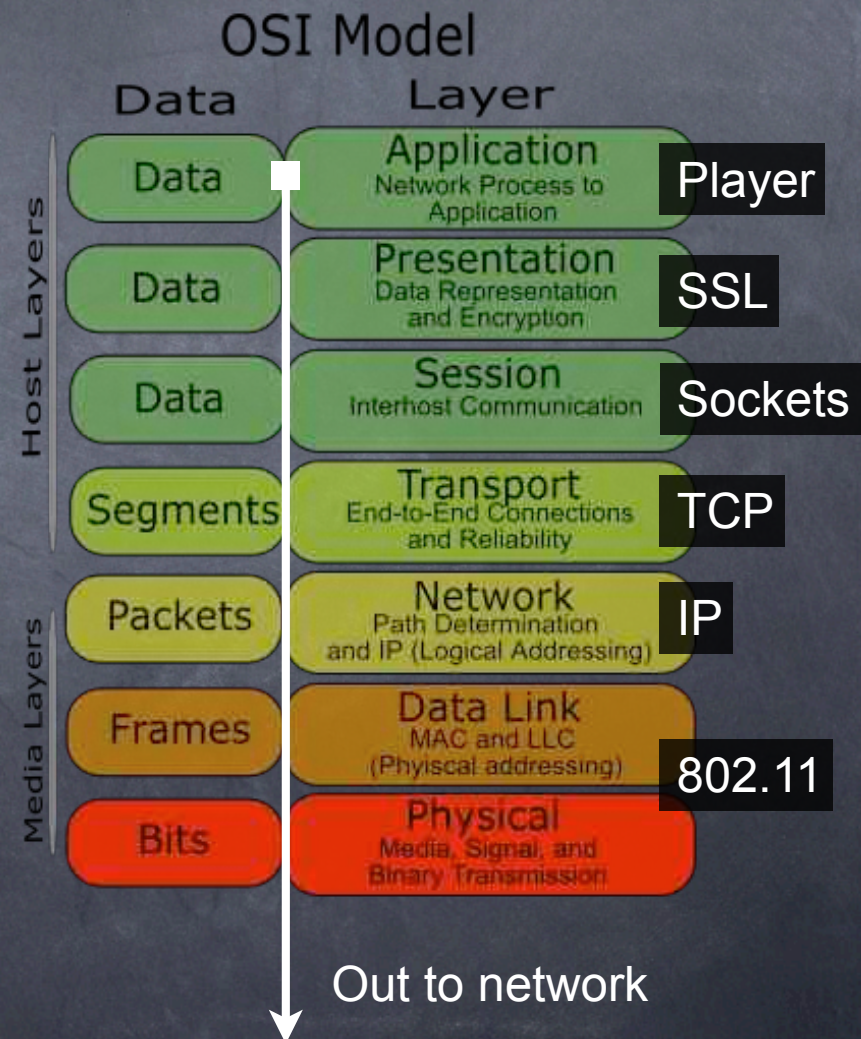
- Implicit coordination
- **Explicit coordination**



- Given  $M$  robots and  $N$  tasks, assign robots to tasks
- Assume each robot has existing controllers for performing each applicable task
- Cast as "assignment problem" to find optimal assignment (or allocation) of robots to tasks

# How to communicate? socket programming

- BSD socket API in C/C++
  - [http://en.wikipedia.org/wiki/Berkeley\\_sockets](http://en.wikipedia.org/wiki/Berkeley_sockets)
- Control sender-receiver connections at session level
- Example of network protocol layers shown for Player msgs
- Simple client/server code ...



```
server.c
/* A simple server in the internet domain using TCP
   The port number is passed as an argument */
#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>

void error(char *msg)
{
    perror(msg);
    exit(1);
}

int main(int argc, char *argv[])
{
    int sockfd, newsockfd, portno, clien;
    char buffer[256];
    struct sockaddr_in serv_addr, cli_addr;
    int n;
    if (argc < 2) {
        fprintf(stderr,"ERROR, no port provided\n");
        exit(1);
    }
    sockfd = socket(AF_INET, SOCK_STREAM, 0);
    if (sockfd < 0)
        error("ERROR opening socket");
    bzero((char *) &serv_addr, sizeof(serv_addr));
    portno = atoi(argv[1]);
    serv_addr.sin_family = AF_INET;
    serv_addr.sin_addr.s_addr = INADDR_ANY;
    serv_addr.sin_port = htons(portno);
    if (bind(sockfd, (struct sockaddr *) &serv_addr,
        sizeof(serv_addr)) < 0)
        error("ERROR on binding");
    listen(sockfd,5);
    clien = sizeof(cli_addr);
    newsockfd = accept(sockfd,
        (struct sockaddr *) &cli_addr,
        &clien);
    if (newsockfd < 0)
        error("ERROR on accept");
    bzero(buffer,256);
    n = read(newsockfd,buffer,255);
    if (n < 0) error("ERROR reading from socket");
    printf("Here is the message: %s\n",buffer);
    n = write(newsockfd,"I got your message",18);
    if (n < 0) error("ERROR writing to socket");
    return 0;
}
```

Note: easy to  
get started,  
hard to get  
right

```
client.c
#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>

void error(char *msg)
{
    perror(msg);
    exit(0);
}

int main(int argc, char *argv[])
{
    int sockfd, portno, n;
    struct sockaddr_in serv_addr;
    struct hostent *server;

    char buffer[256];
    if (argc < 3) {
        fprintf(stderr,"usage %s hostname port\n", argv[0]);
        exit(0);
    }
    portno = atoi(argv[2]);
    sockfd = socket(AF_INET, SOCK_STREAM, 0);
    if (sockfd < 0)
        error("ERROR opening socket");
    server = gethostbyname(argv[1]);
    if (server == NULL) {
        fprintf(stderr,"ERROR, no such host\n");
        exit(0);
    }
    bzero((char *) &serv_addr, sizeof(serv_addr));
    serv_addr.sin_family = AF_INET;
    bcopy((char *)server->h_addr,
        (char *)&serv_addr.sin_addr.s_addr,
        server->h_length);
    serv_addr.sin_port = htons(portno);
    if (connect(sockfd,&serv_addr,sizeof(serv_addr)) < 0)
        error("ERROR connecting");
    printf("Please enter the message: ");
    bzero(buffer,256);
    fgets(buffer,255,stdin);
    n = write(sockfd,buffer,strlen(buffer));
    if (n < 0)
        error("ERROR writing to socket");
    bzero(buffer,256);
    n = read(sockfd,buffer,255);
    if (n < 0)
        error("ERROR reading from socket");
    printf("%s\n",buffer);
    return 0;
}
```

```
server.c
/* A simple server in the internet domain using TCP
   The port number is passed as an argument */
#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>

void error(char *msg)
{
    perror(msg);
    exit(1);
}

int main(int argc, char *argv[])
{
    int sockfd, newsockfd, portno, clien;
    char buffer[256];
    struct sockaddr_in serv_addr, cli_addr;
    int n;
    if (argc < 2) {
        fprintf(stderr,"ERROR, no port provided\n");
        exit(1);
    }
    sockfd = socket(AF_INET, SOCK_STREAM, 0);
    if (sockfd < 0)
        error("ERROR opening socket");
    bzero((char *) &serv_addr, sizeof(serv_addr));
    portno = atoi(argv[1]);
    serv_addr.sin_family = AF_INET;
    serv_addr.sin_addr.s_addr = INADDR_ANY;
    serv_addr.sin_port = htons(portno);
    if (bind(sockfd, (struct sockaddr *) &serv_addr,
        sizeof(serv_addr)) < 0)
        error("ERROR on binding");
    listen(sockfd,5);
    clien = sizeof(cli_addr);
    newsockfd = accept(sockfd,
        (struct sockaddr *) &cli_addr,
        &clien);
    if (newsockfd < 0)
        error("ERROR on accept");
    bzero(buffer,256);
    n = read(newsockfd,buffer,255);
    if (n < 0) error("ERROR reading from socket");
    printf("Here is the message: %s\n",buffer);
    n = write(newsockfd,"I got your message",18);
    if (n < 0) error("ERROR writing to socket");
    return 0;
}
```

```
> gcc server.c -o server
> ./server 6664
Here is the message:
sealion
```

```
client.c
#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>

void error(char *msg)
{
    perror(msg);
    exit(0);
}

int main(int argc, char *argv[])
{
    int sockfd, portno, n;
    struct sockaddr_in serv_addr;
    struct hostent *server;

    char buffer[256];
    if (argc < 3) {
        fprintf(stderr,"usage %s hostname port\n", argv[0]);
        exit(0);
    }
    portno = atoi(argv[2]);
    sockfd = socket(AF_INET, SOCK_STREAM, 0);
    if (sockfd < 0)
        error("ERROR opening socket");
    server = gethostbyname(argv[1]);
    if (server == NULL) {
        fprintf(stderr,"ERROR, no such host\n");
        exit(0);
    }
    bzero((char *) &serv_addr, sizeof(serv_addr));
    serv_addr.sin_family = AF_INET;
    bcopy((char *)server->h_addr,
        (char *)&serv_addr.sin_addr.s_addr,
        server->h_length);
    serv_addr.sin_port = htons(portno);
    if (connect(sockfd,&serv_addr,sizeof(serv_addr)) < 0)
        error("ERROR connecting");
    printf("Please enter the message: ");
    bzero(buffer,256);
    fgets(buffer,255,stdin);
    n = write(sockfd,buffer,strlen(buffer));
    if (n < 0)
        error("ERROR writing to socket");
    bzero(buffer,256);
    n = read(sockfd,buffer,255);
    if (n < 0)
        error("ERROR reading from socket");
    printf("%s\n",buffer);
    return 0;
}
```

```
> gcc client.c -o client
> ./client localhost 6664
Please enter the
message: sealion
I got your message
```

```
server.c
/* A simple server in the internet domain using TCP
   The port number is passed as an argument */
#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>

void error(char *msg)
{
    perror(msg);
    exit(1);
}

int main(int argc, char *argv[])
{
    int sockfd, newsockfd, portno, clien;
    char buffer[256];
    struct sockaddr_in serv_addr, cli_addr;
    int n;
    if (argc < 2) {
        fprintf(stderr,"ERROR, no port provided\n");
        exit(1);
    }
    sockfd = socket(AF_INET, SOCK_STREAM, 0);
    if (sockfd < 0)
        error("ERROR opening socket");
    bzero((char *) &serv_addr, sizeof(serv_addr));
    portno = atoi(argv[1]);
    serv_addr.sin_family = AF_INET;
    serv_addr.sin_addr.s_addr = INADDR_ANY;
    serv_addr.sin_port = htons(portno);
    if (bind(sockfd, (struct sockaddr *) &serv_addr,
        sizeof(serv_addr)) < 0)
        error("ERROR on binding");
    listen(sockfd,5);
    clien = sizeof(cli_addr);
    newsockfd = accept(sockfd,
        (struct sockaddr *) &cli_addr,
        &clien);
    if (newsockfd < 0)
        error("ERROR on accept");
    bzero(buffer,256);
    n = read(newsockfd,buffer,255);
    if (n < 0) error("ERROR reading from socket");
    printf("Here is the message: %s\n",buffer);
    n = write(newsockfd,"I got your message",18);
    if (n < 0) error("ERROR writing to socket");
    return 0;
}
```

init socket, resolve  
hostname and port

bind to port  
and listen

accept connection,  
read message

send ack

```
client.c
#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>

void error(char *msg)
{
    perror(msg);
    exit(0);
}

int main(int argc, char *argv[])
{
    int sockfd, portno, n;
    struct sockaddr_in serv_addr;
    struct hostent *server;

    char buffer[256];
    if (argc < 3) {
        fprintf(stderr,"usage %s hostname port\n", argv[0]);
        exit(0);
    }
    portno = atoi(argv[2]);
    sockfd = socket(AF_INET, SOCK_STREAM, 0);
    if (sockfd < 0)
        error("ERROR opening socket");
    server = gethostbyname(argv[1]);
    if (server == NULL) {
        fprintf(stderr,"ERROR, no such host\n");
        exit(0);
    }
    bzero((char *) &serv_addr, sizeof(serv_addr));
    serv_addr.sin_family = AF_INET;
    bcopy((char *)server->h_addr,
        (char *)&serv_addr.sin_addr.s_addr,
        server->h_length);
    serv_addr.sin_port = htons(portno);
    if (connect(sockfd,&serv_addr,sizeof(serv_addr)) < 0)
        error("ERROR connecting");
    printf("Please enter the message: ");
    bzero(buffer,256);
    fgets(buffer,255,stdin);
    n = write(sockfd,buffer,strlen(buffer));
    if (n < 0)
        error("ERROR writing to socket");
    bzero(buffer,256);
    n = read(sockfd,buffer,255);
    if (n < 0)
        error("ERROR reading from socket");
    printf("%s\n",buffer);
    return 0;
}
```

init socket, resolve  
hostname and port

connect to  
server

prepare and  
send message

get acknowledgement

# Coordination Taxonomy

[Gerkey, Mataric 2004]

- Assume centralized coordinator (one of the robots)
- Coordination problem depends on 3 factors:
  - single-task robots (ST) vs. multi-task robots (MT)
  - single-robot tasks (SR) vs. multi-robot tasks (MR)
  - instantaneous assignment (IA) vs. time-extended assignment (TA)
- Focus on ST-SR-IA (most common), also of interest:
  - ST-MR-IA: coalition formation
  - ST-SR-TA: scheduling
  - MR-MR-IA: set cover problem

# ST-SR-IA as an Optimal Assignment Problem

- Given  $M$  robots and  $N$  tasks, form utility matrix  $U$
- $U_{ij}$  is relative benefit of robot  $i$  performing task  $j$
- Given  $U$ , select elements  $\alpha=[0,1]$  that maximize overall utility
- Assignment must be unique mapping of robots to tasks

formation of utility matrix

$$U_{RT} = \begin{cases} Q_{RT} - C_{RT} & \text{if } R \text{ is capable of executing} \\ & T \text{ and } Q_{RT} > C_{RT} \\ 0 & \text{otherwise} \end{cases}$$

computation of utility

$$U = \sum_{i=1}^m \sum_{j=1}^n \alpha_{ij} U_{ij} w_j$$

assign only 1 robot to a task

$$\sum_{i=1}^m \alpha_{ij} = 1, \quad 1 \leq j \leq n$$



$$\sum_{j=1}^n \alpha_{ij} = 1, \quad 1 \leq i \leq m.$$

assign only 1 task to a robot

# Soccer Example

- How to allocate given these utilities?

Utility matrix This matrix is guesstimated

Robot/Task	Defender	Attacker
 aibo	8	10
 smurv w/camera	1	7



Objective:  
"win soccer match"



# Soccer Example

How to allocate given these utilities?

Greedy allocation?

$$U=11=10+1$$

Utility matrix

Robot/Task	Defender	Attacker
 aibo	8	10
 smurv w/camera	1	7





Objective:  
"win soccer match"

# Soccer Example

- How to allocate given these utilities?
- Greedy allocation?  
 $U=11=10+1$
- Optimal allocation  
 $U=15=8+7$

Utility matrix

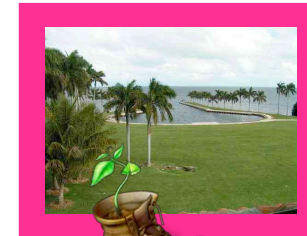
Robot/Task	Defender	Attacker
 aibo	8	10
 smurv w/camera	1	7



Objective:  
"win soccer match"

# "Directive" Example

Objective: put  
plant by window



Task: watch to make  
sure objective is met



Task: push table to window  
(2 robots: left, right)



Task: put  
plant on table

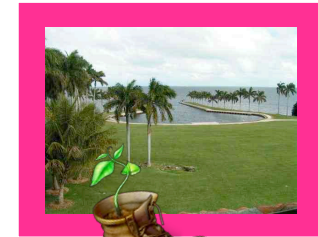
# "Directive" Example

Ideally, one robot could do all of these tasks... in the year 2105



Task: watch to make sure objective is met

Objective: put plant by window



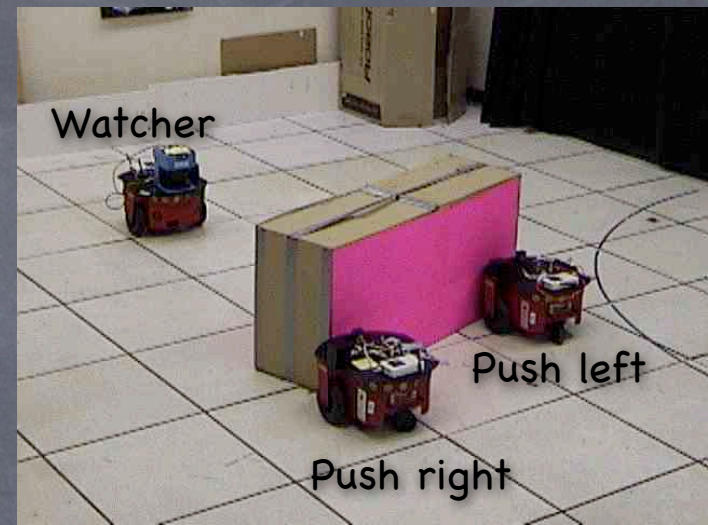
Task: push table to window  
(2 robots: left, right)



Task: put plant on table

# Box pushing task

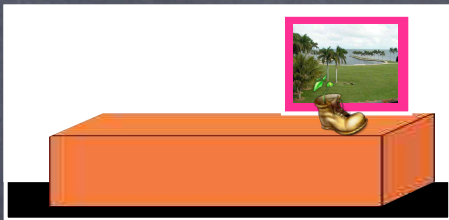
[Gerkey, Mataric 2002]



Straight push to landmark








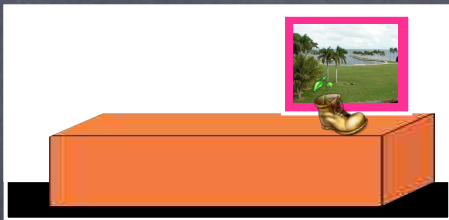
Turn box left, then  
straight push to landmark



Objective: put plant by window

How to allocate?

U	Robot/ Task	Watcher	Box Push Right	Box Push Left	Place Plant
Create w/camera		9	4	4	0
Pioneer w/camera and laser		4	7	7	0
Pioneer w/camera and grip		4	7	7	3
Aibo		3	1	1	0
PR2		12	11	11	10








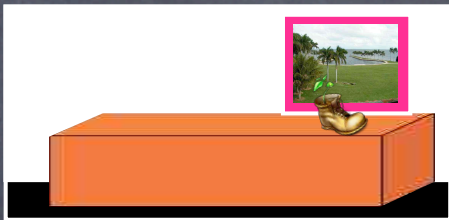
Objective: put plant by window

How to allocate?

Greedy can still be problematic

(deadlock condition)

U	Robot/ Task	Watcher	Box Push Right	Box Push Left	Place Plant
Create w/camera		9	4	4	0
Pioneer w/camera and laser		4	7	7	0
Pioneer w/camera and grip		4	7	7	3
Aibo		3	1	1	0
PR2		12	11	11	10







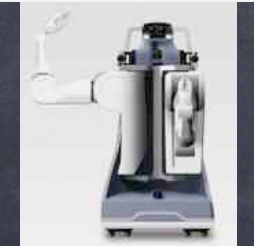
Objective: put plant by window

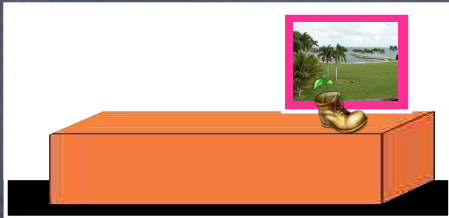
How to allocate?

Greedy can still be problematic

Combinatorial optimization:  
Hungarian algorithm  
[Kuhn 1955]

Let's do the example






U	Robot/ Task	Watcher	Box Push Right	Box Push Left	Place Plant
Create w/camera		9	4	4	0
Pioneer w/camera and laser		4	7	7	0
Pioneer w/camera and grip		4	7	7	3
Aibo		3	1	1	0
PR2		12	11	11	10

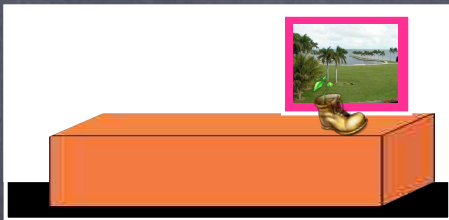


Hungarian alg.  
example

cost matrix

$$C = U - \max(U)$$

C	Robot/ Task	Watcher	Box Push Right	Box Push Left	Place Plant
Create w/camera		3	8	8	12
Pioneer w/camera and laser		8	5	5	12
Pioneer w/camera and grip		8	5	5	9
Aibo		9	11	11	12
PR2		0	1	1	2



Hungarian alg.  
example






cost matrix

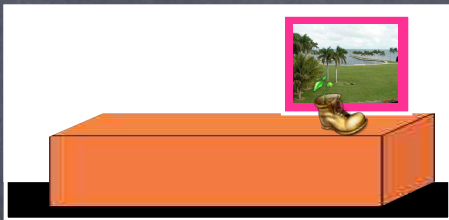
$$C = U - \max(U)$$

subtract each row  
by smallest entry

$$C_i = C_i - \min(C_i)$$

find unique asgns  
with zero cost

C	Robot/ Task	Watcher	Box Push Right	Box Push Left	Place Plant
Create w/camera		0	5	5	9
Pioneer w/camera and laser		3	0	0	5
Pioneer w/camera and grip		3	0	0	4
Aibo		0	2	2	3
PR2		0	1	1	2



Hungarian alg.  
example

cost matrix

$$C = U - \max(U)$$

subtract each row  
by smallest entry






$$C_i = C_i - \min(C_i)$$

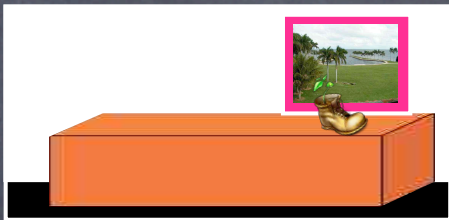
find unique asgns  
with zero cost

subtract each col  
by smallest entry

$$C_j = C_j - \min(C_j)$$

find unique asgns  
with zero cost

C	Robot/ Task	Watcher	Box Push Right	Box Push Left	Place Plant
Create w/camera		0	5	5	7
Pioneer w/camera and laser		3	0	0	3
Pioneer w/camera and grip		3	0	0	2
Aibo		0	2	2	1
PR2		0	1	1	0



Hungarian alg.  
example

cost matrix

$$C = U - \max(U)$$

subtract each row  
by smallest entry






$$C_i = C_i - \min(C_i)$$

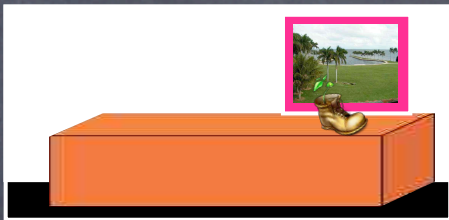
find unique asgns  
with zero cost

subtract each col  
by smallest entry

$$C_j = C_j - \min(C_j)$$






find unique asgns  
with zero cost

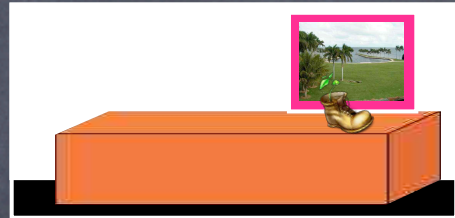
C	Robot/ Task	Watcher	Box Push Right	Box Push Left	Place Plant
Create w/camera		0	5	5	7
Pioneer w/camera and laser		3	0	0	3
Pioneer w/camera and grip		3	0	0	2
Aibo		0	2	2	1
PR2		0	1	1	0



Hungarian alg.  
**problem** example

Suppose we  
 slightly change  
 one entry

C	Robot/ Task	Watcher	Box Push Right	Box Push Left	Place Plant
Create w/camera		3	8	8	12
Pioneer w/camera and laser		8	5	5	12
Pioneer w/camera and grip		8	5	5	9
Aibo		10	11	11	12
PR2		0	1	1	2








Hungarian alg.  
**problem** example

Warning: be very  
 careful in  
 estimating utilities

Suppose we  
 slightly change  
 one entry

Result creates  
 uncertain  
 assignment

C	Robot/ Task	Watcher	Box Push Right	Box Push Left	Place Plant
Create w/camera		0	5	5	7
Pioneer w/camera and laser		3	0	0	3
Pioneer w/camera and grip		3	0	0	2
Aibo		0	1	1	0
PR2		0	1	1	0

# McLurkin's Swarmbots



**Follow the leader**