

# Roger-the-Crab

Rod Grupen  
Department of Computer Science  
University of Massachusetts Amherst

## Un-Crating your Roger - C and X windows

**Platforms:**

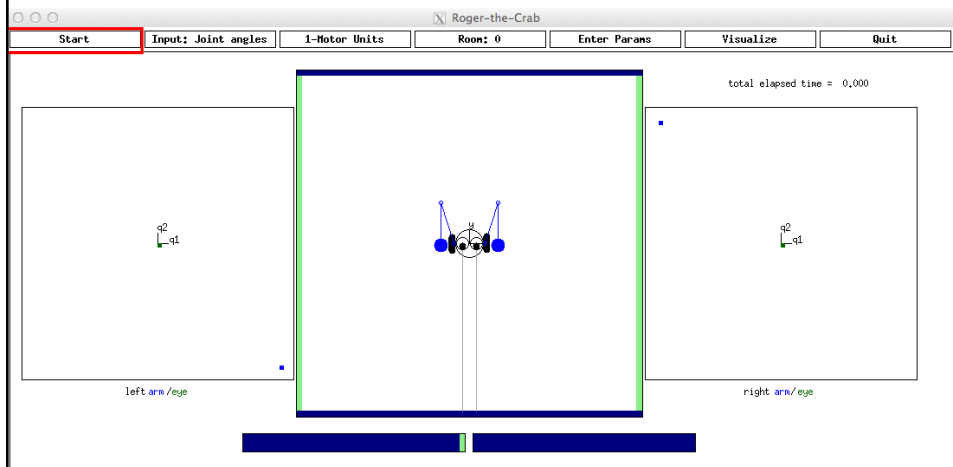
- Linux (VM, dual booting works), OS X (+Xquartz)

**Getting Started:**

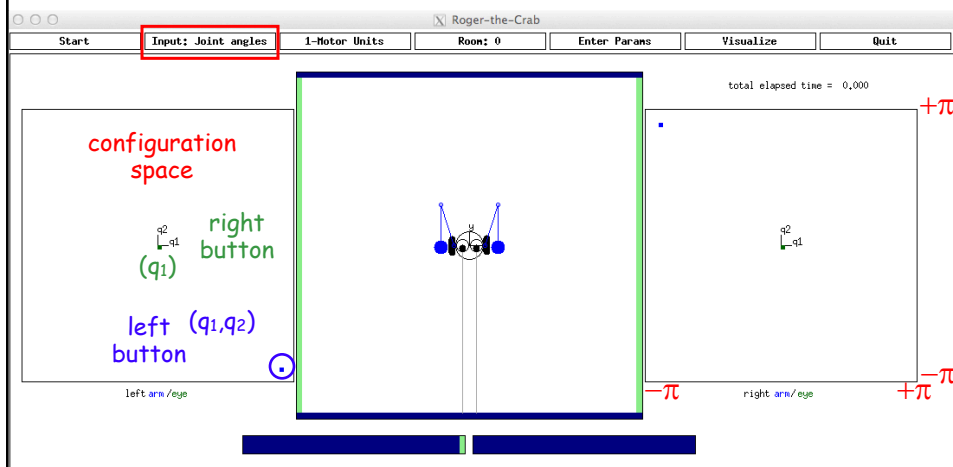
- download your copy of the Roger simulator from  
[www-robotics.cs.umass.edu/~grupen/403/code](http://www-robotics.cs.umass.edu/~grupen/403/code)
- unpack the compressed tar/zip file:  

```
tar -xvf roger-2020.tar
```
- follow the directions in the README

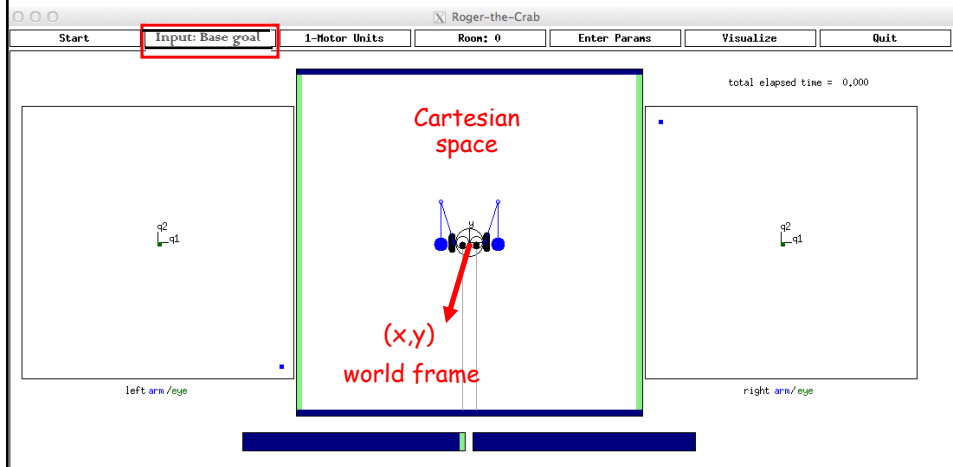
# The Simulator (Development) Environment



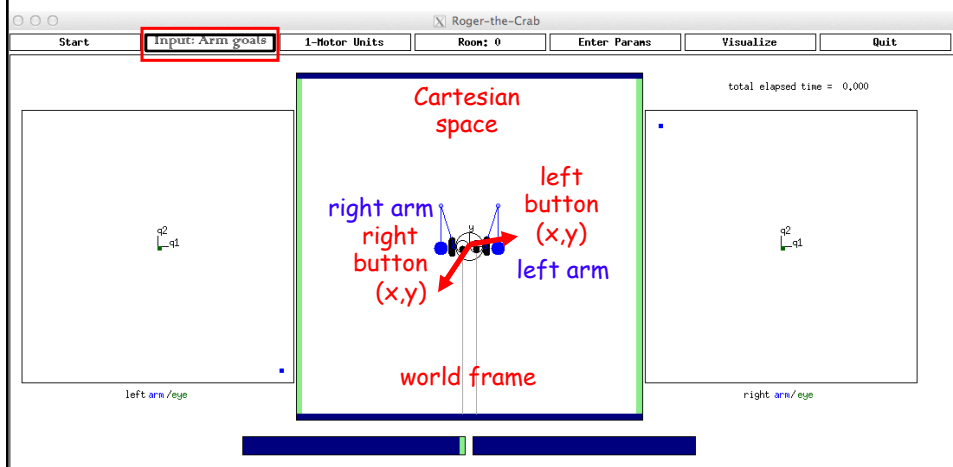
# Input Modes: Joint Angle inputs



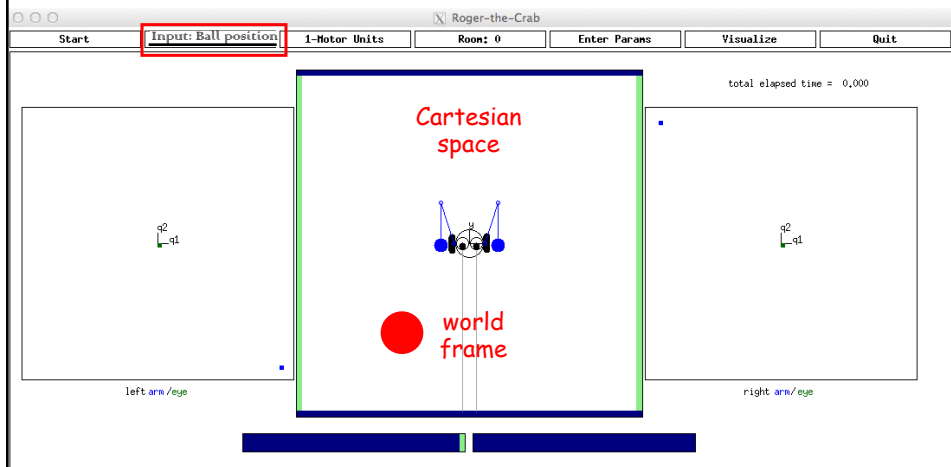
# Input Modes: Base goals



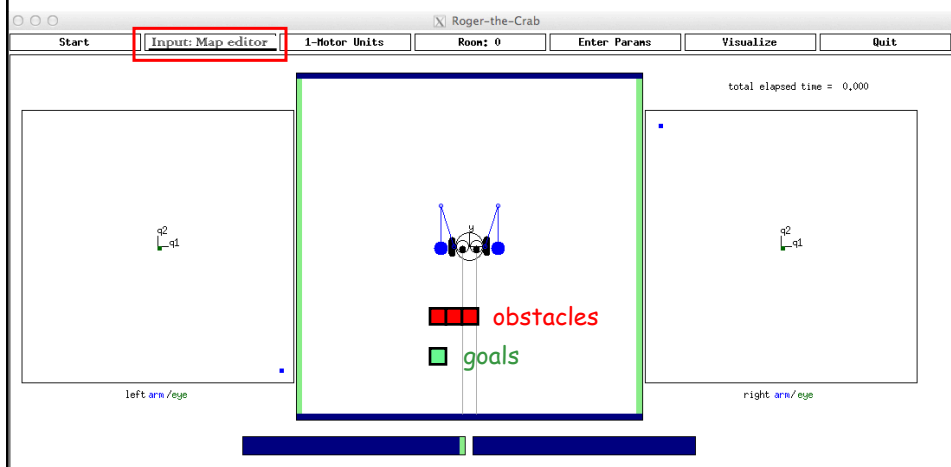
# Input Modes: Arm goals



## Input Modes: Introducing an Object (Ball)



## Input Modes: Map Editor



# Control Modes

The screenshot shows the 'Roger-the-Crab' software interface. At the top, there is a menu bar with buttons for 'Start', 'Input: Joint angles', '1-Motor Units' (highlighted with a red box), 'Room: 0', 'Enter Params', 'Visualize', and 'Quit'. The main window is divided into three sections. On the left, there is a plot with axes labeled  $q_2$  and  $q_1$  and the text 'left arm/eye'. In the center, there is a diagram of the robot's head and neck with two blue eyes and a central sensor. On the right, there is a list of control modes: '1-Motor Units', '2-ArmKinematics', '3-Vision', '4-SearchTrack', '5-StereoKinematics', '6-Kalman', '7-ChasePunch', '8-Path Planning', '9-PONG', '10-Model', and '11-Belief'. The text 'total elapsed time = 0,000' is visible in the top right corner.

# Environmental Maps

The screenshot shows the 'Roger-the-Crab' software interface. At the top, there is a menu bar with buttons for 'Start', 'Input: Joint angles', '1-Motor Units', 'Room: 0' (highlighted with a red box), 'Enter Params', 'Visualize', and 'Quit'. The main window is divided into three sections. On the left, there is a plot with axes labeled  $q_2$  and  $q_1$  and the text 'left arm/eye'. In the center, there is a diagram of the robot's head and neck with two blue eyes and a central sensor. On the right, there is a plot with axes labeled  $q_2$  and  $q_1$  and the text 'right arm/eye'. The text 'total elapsed time = 0,000' is visible in the top right corner.

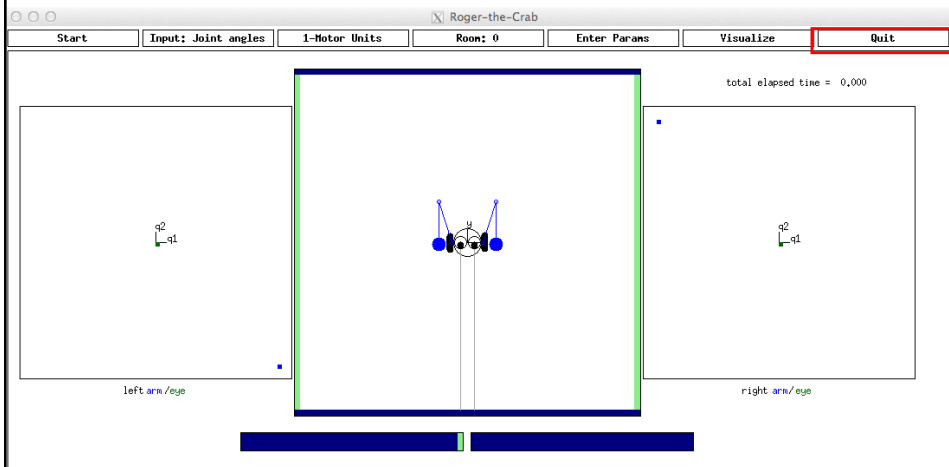
## Command line I/O

accurate setpoints, gains, etc  
for experiments  
starting/stopping plot data

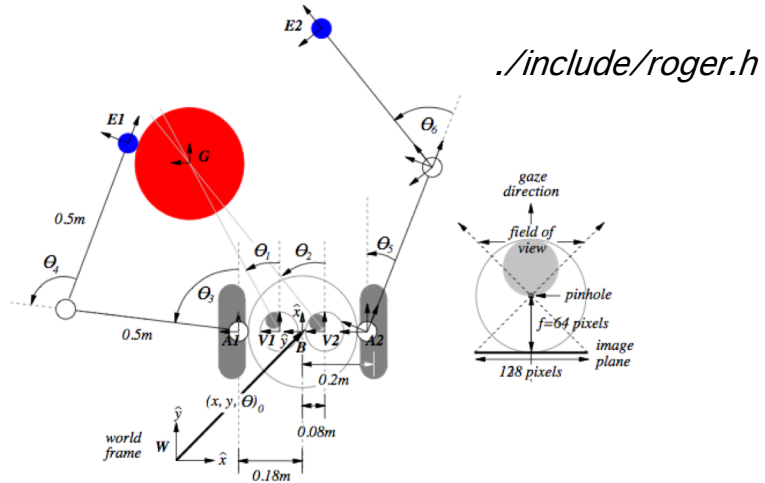
## Project Specific Visualization

Project/user defined tools  
location uncertainty  
path plans  
potential maps

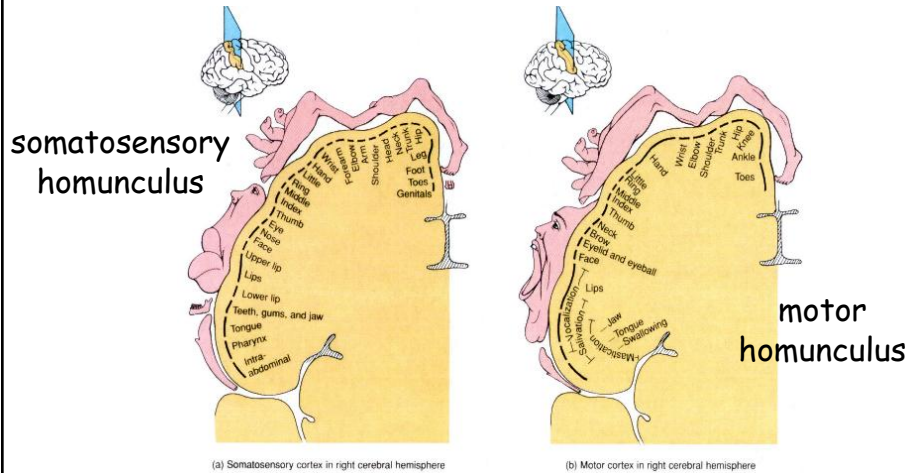
## Quiting the Simulator



## Roger-the-Crab - Kinematic Definition



## Primary Cortical Homunculi



## Homunculus Man

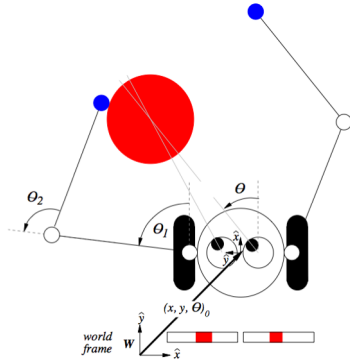


**somatosensory homunculus**





## Afferents



eyes:

- $\theta[2], \dot{\theta}[2], \text{images}[2][128][3]$

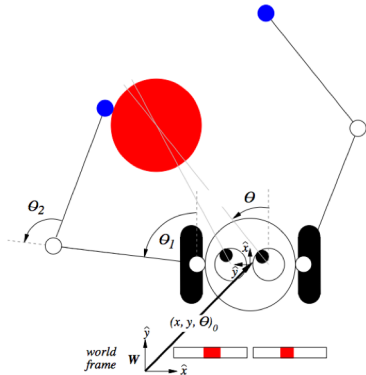
arms:

- $\theta_1[2], \dot{\theta}_1[2], \theta_2[2], \dot{\theta}_2[2]$
- tactile (force) sensors:  $f[2] \in \mathbb{R}^2$

mobile base:

- position  $(x, \dot{x}, y, \dot{y})$ , orientation  $(\theta, \dot{\theta})$
- bump (force) sensor:  $f \in \mathbb{R}^2$

## Efferents



eye torques:

- $\tau[2]$

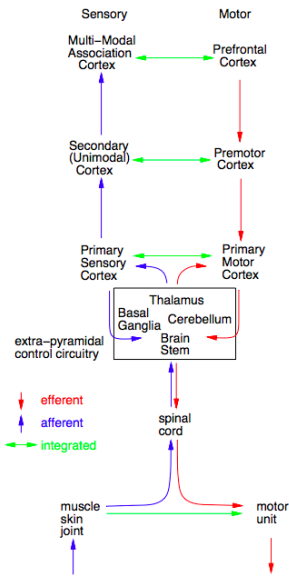
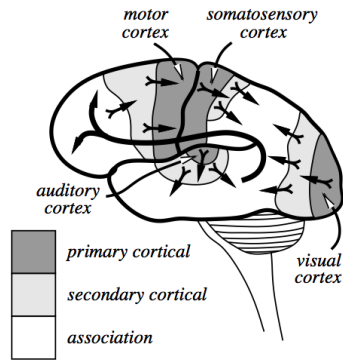
arm torques:

- $\tau_1[2], \tau_2[2]$

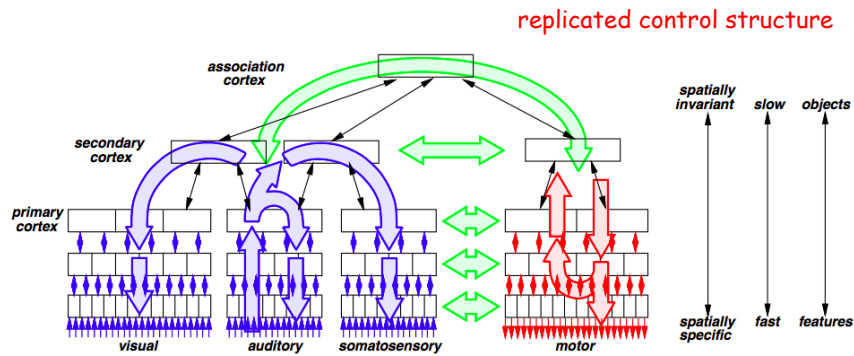
mobile base:

- wheel torques  $\tau[2]$

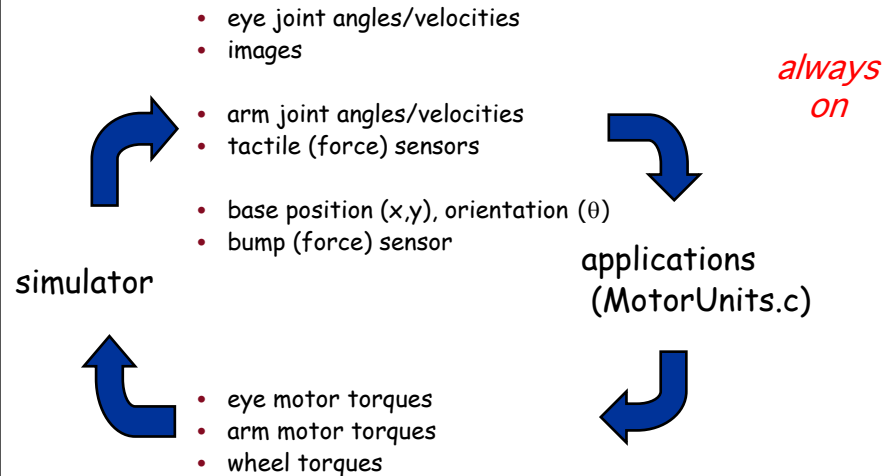
# Hierarchical Control



# Building Hierarchical Control Code



## Robot Interface: Project #1, #2



## Control Interface - control.h

```
typedef struct Robot_interface {
// SENSORS
double eye_theta[NEYES];
double eye_theta_dot[NEYES];
int image[NEYES][NPIXELS][NPRIMARY_COLORS]; /* rgb */
double arm_theta[NARMS][NARM_JOINTS];
double arm_theta_dot[NARMS][NARM_JOINTS];
double ext_force[NARMS][2]; /* (fx,fy) force on arm endpoint */
double base_position[3]; /* x,y,theta */
double base_velocity[3];

// MOTORS
double eye_torque[NEYES];
double arm_torque[NARMS][NARM_JOINTS];
double wheel_torque[NWHEELS];

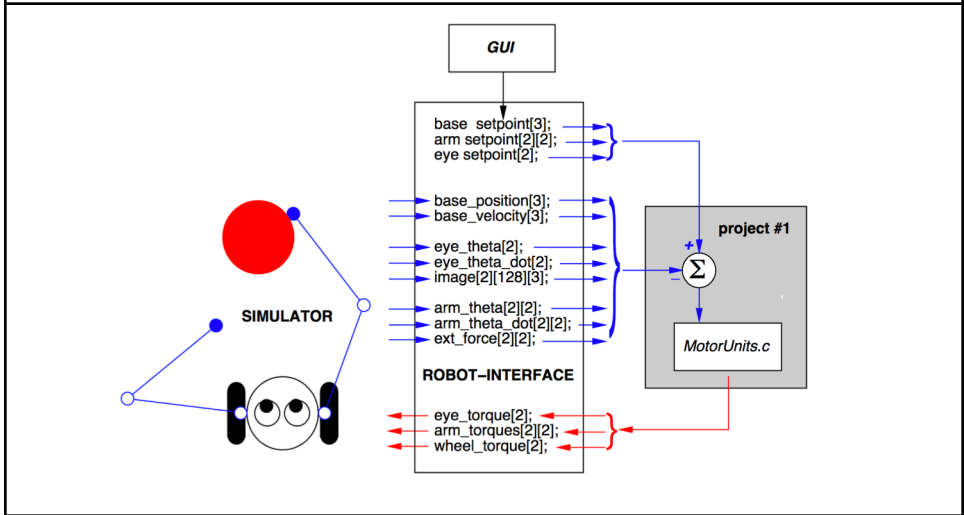
typedef struct _map {
int occupancy_map[NBINS][NBINS];
double potential_map[NBINS][NBINS];
int color_map[NBINS][NBINS];
} Map;

// TELEOPERATOR
int button_event;
double button_reference[2];

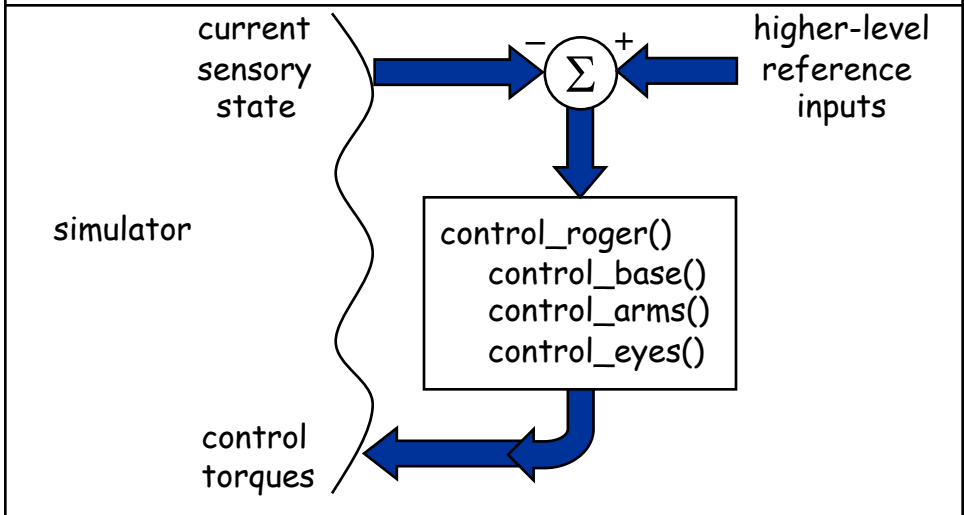
// CONTROL MODE
int control_mode;
int input_mode;
Map world_map, arm_map[NARMS];

// REFERENCE VALUE
double base_setpoint[3]; /* desired world frame base position (x,y,theta) */
double arm_setpoint[NARMS][NARM_JOINTS]; /* desired arm joint angles */
double eyes_setpoint[NEYES]; /* desired eye pan angle */
} Robot;
```

Project #1 - Code Structure



MotorUnits.c



## Cumulative Project Work

1. motor units
2. Cartesian goals
3. oculomotor behavior
4. visual reconstruction - triangulation
5. "hunting" - integrated behavior
6. ...

options:

1. path planning
2. learning
3. Pong
- 4....