# Roombots-CPG, Symmetries and online optimization.

Tuleu Alexandre

Midterm presentation

November 3, 2008

# Goal of the project.

- Implement sevral Roombot configurations, and find a way to identify morphological symmetries, in order to design a CPG along this structure.

# Goal of the project.

- Implement sevral Roombot configurations, and find a way to identify morphological symmetries, in order to design a CPG along this structure.
- Check out and test new optimization methods.

# Goal of the project.

- Implement sevral Roombot configurations, and find a way to identify morphological symmetries, in order to design a CPG along this structure.
- Check out and test new optimization methods.
- Test and select different CPG type.

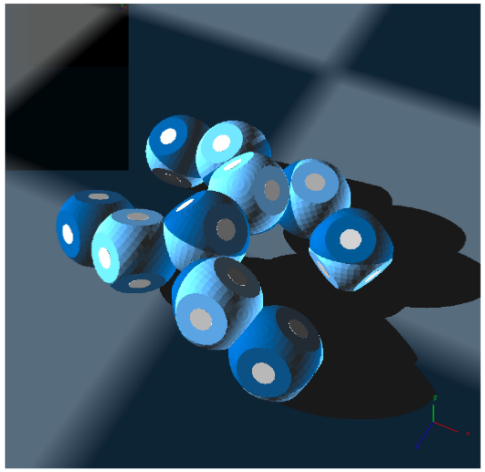
# Goal of the project.

- Implement sevral Roombot configurations, and find a way to identify morphological symmetries, in order to design a CPG along this structure.
- Check out and test new optimization methods.
- Test and select different CPG type.
  - add sensor feedback from the roombot to the CPG.

# Goal of the project.

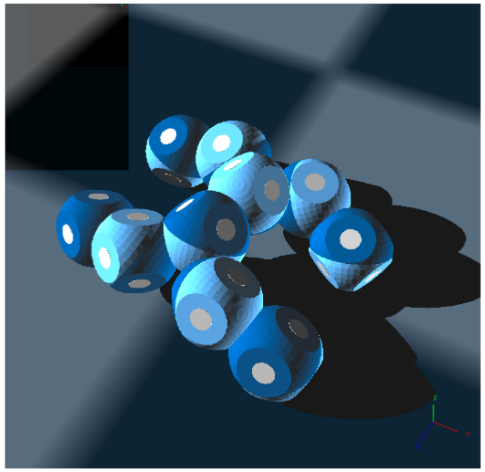
- Implement sevral Roombot configurations, and find a way to identify morphological symmetries, in order to design a CPG along this structure.
- Check out and test new optimization methods.
- Test and select different CPG type.
  - add sensor feedback from the roombot to the CPG.
- Lot of previous work done by former student (YaMor host 3.0).

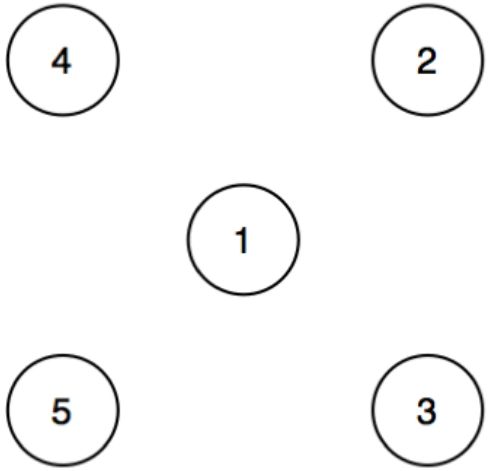
# A Way to Caracterize a Roombot Structure.



- Based on previous work on YaMor [?].

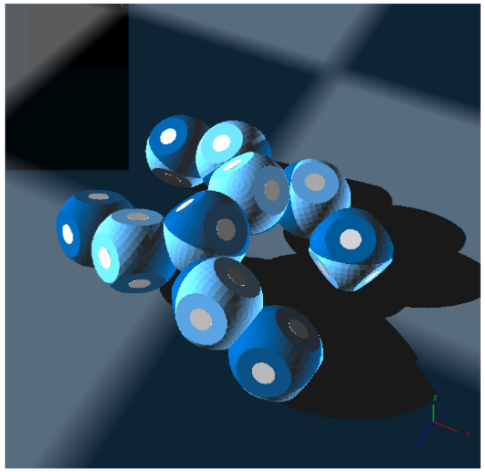
# A Way to Caracterize a Roombot Structure.

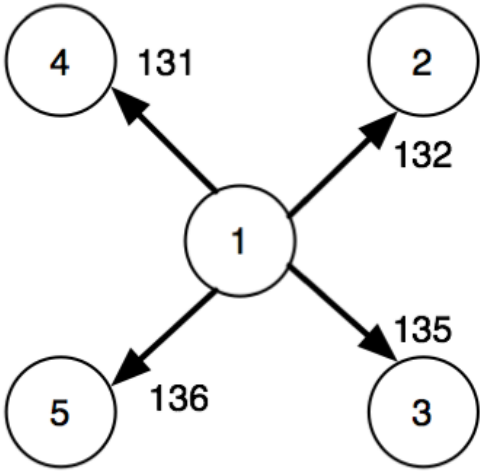





- Based on previous work on YaMor [?].
- We represent the structure as a graph :
  - Each module is a vertice.

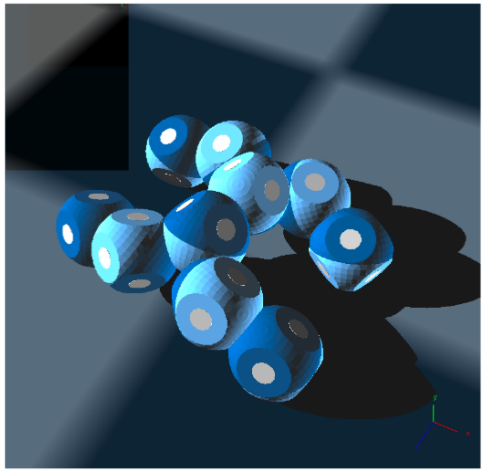
# A Way to Caracterize a Roombot Structure.

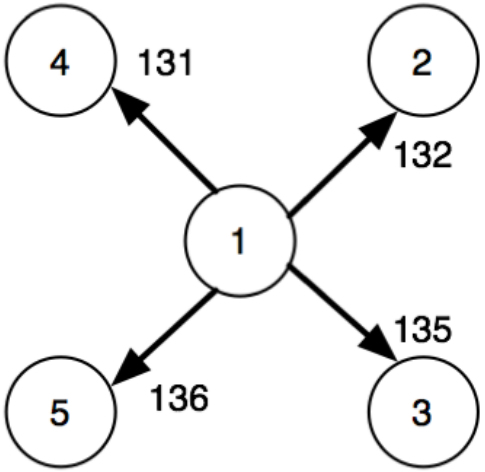





- Based on previous work on YaMor [?].
- We represent the structure as a graph :
  - Each module is a vertice.
  - Each connection between module are a edge.

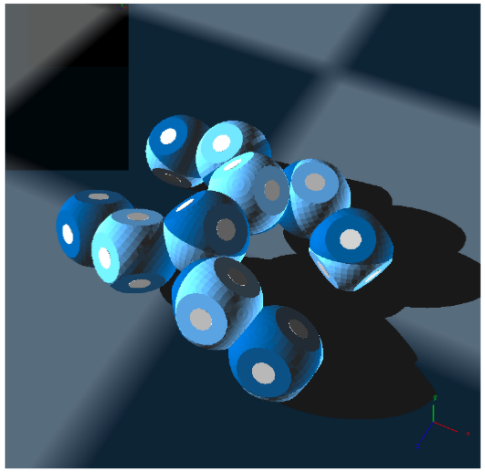
# A Way to Caracterize a Roombot Structure.

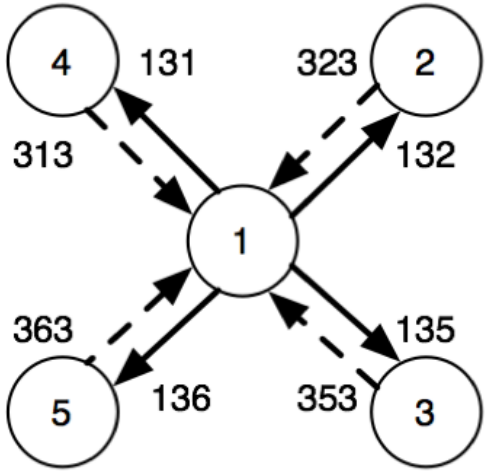





- Based on previous work on YaMor [?].
- We represent the structure as a graph :
  - Each module is a vertice.
  - Each connection between module are a edge.
  - We put label on edge that fully caracaterize the conection between module.

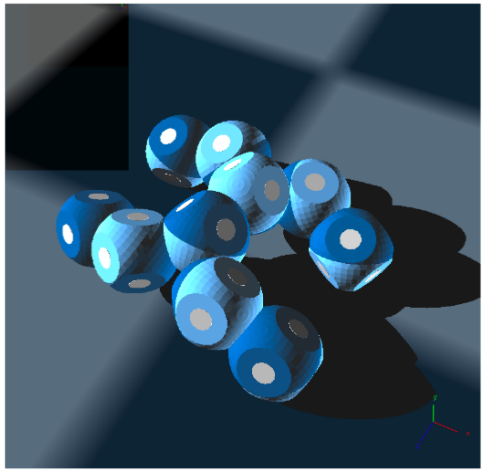
# A Way to Caracterize a Roombot Structure.

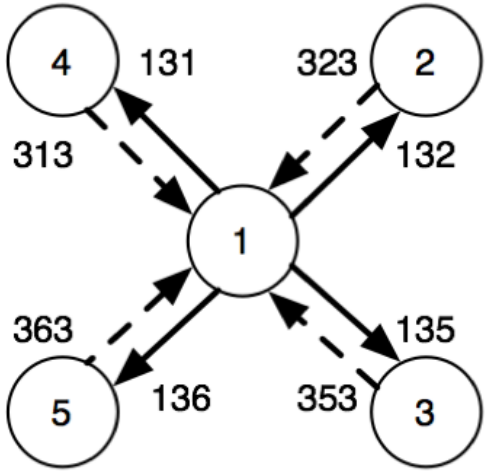





- Based on previous work on YaMor [?].
- We represent the structure as a graph :
  - Each module is a vertice.
  - Each connection between module are a edge.
  - We put label on edge that fully caracaterize the conection between module.

# A Way to Caracterize a Roombot Structure.






- Based on previous work on YaMor [?].
- We represent the structure as a graph :
  - Each module is a vertice.
  - Each connection between module are a edge.
  - We put label on edge that fully caracaterize the conection between module.
- Different frome Jocelyne's one, since our graph must be servo values independant.

# Analyse the structure.

- Main Idea : Symmetries in a CPG, conduct to the same symmetries on the gait.

# Analyse the structure.

- Main Idea : Symmetries in a CPG, conduct to the same symmetries on the gait.
- Find a way to detect symmetries in a structure, and then determine the available gait.

# Analyse the structure.

- Main Idea : Symmetries in a CPG, conduct to the same symmetries on the gait.
- Find a way to detect symmetries in a structure, and then determine the available gait.
- Maybe try to memorize some patterns.

# Online Learning.

- Our goal is to learn a gait by optmization :

$$\max_{\mathbf{X}} f(\mathbf{X}, \alpha) \tag{1}$$

- We don't know a mathematical expression for $f$, but $f$ is:
  - nonlinear.
  - multimodal.

# Different Algorithm for Online Learning.

- Lot of *random*, and *evolutaniory* algorithms :

# Different Algorithm for Online Learning.

- Lot of *random*, and *evolutaniory* algorithms :
  PSO,

# Different Algorithm for Online Learning.

- Lot of *random*, and *evolutaniory* algorithms :
  PSO, Cross-Entropy Method,

# Different Algorithm for Online Learning.

- Lot of *random*, and *evolutaniory* algorithms :
  PSO, Cross-Entropy Method, Harmony search.

# Different Algorithm for Online Learning.

- Lot of *random*, and *evolutaniory* algorithms :
  PSO, Cross-Entropy Method, Harmony search.
- Powel's method can be extended to multimodal function :

# Different Algorithm for Online Learning.

- Lot of *random*, and *evolutaniory* algorithms :
  PSO, Cross-Entropy Method, Harmony search.
- Powel's method can be extended to multimodal function :
  - use of systematical search instead of golden section search.

# Different Algorithm for Online Learning.

- Lot of *random*, and *evolutaniory* algorithms :
  PSO, Cross-Entropy Method, Harmony search.
- Powel's method can be extended to multimodal function :
  - use of systematical search instead of golden section search.
  - Boender-Rinnooy-Stougie-Timmer (BRST) algorithm.

# Estimating the objective function.

- What is more important than the algorithm is the expression of objective function.

# Estimating the objective function.

- What is more important than the algorithm is the expression of objective function.
- We want to avoid "traps".

# Estimating the objective function.

- What is more important than the algorithm is the expression of objective function.
- We want to avoid "traps".
  - done by integrating over more time, but has a cost.

# Estimating the objective function.

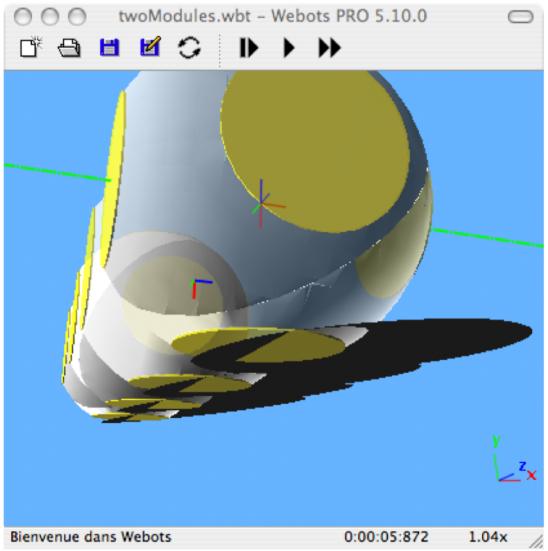
- What is more important than the algorithm is the expression of objective function.
- We want to avoid "traps".
  - done by integrating over more time, but has a cost.
- Idea : mesure the periodicity of the function :

# Estimating the objective function.

- What is more important than the algorithm is the expression of objective function.
- We want to avoid "traps".
  - done by integrating over more time, but has a cost.
- Idea : mesure the periodicity of the function :

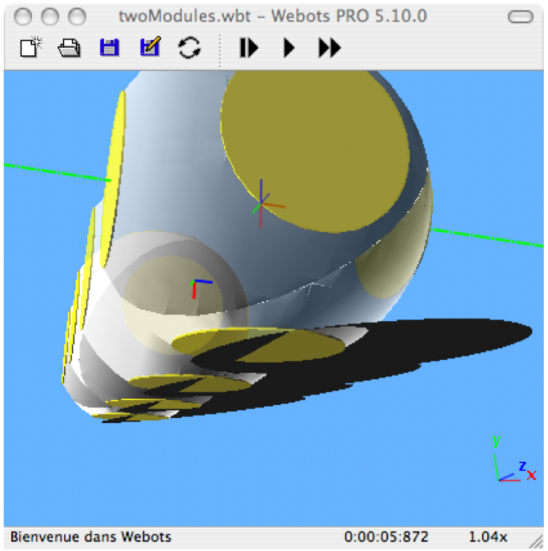
$$f(X) = \frac{s(t_{end}) - s(t_{deb})}{t_{end} - t_{deb}} - K. \int_{t_{deb}+T}^{t_{end}} (\dot{s}(t) - \dot{s}(t-T))^2 \, dt \tag{2}$$

# Modelisation of the contacts.



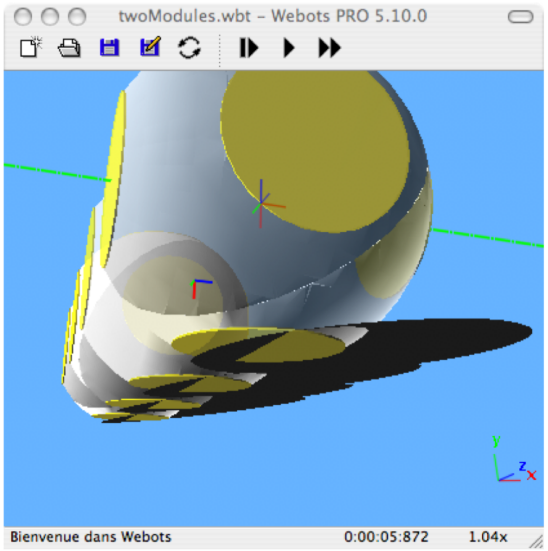

- Now modelized by a sphere.

# Modelisation of the contacts.




- Now modelized by a sphere.
- The real shape collision with a plane isn't so hard to compute.

# Modelisation of the contacts.

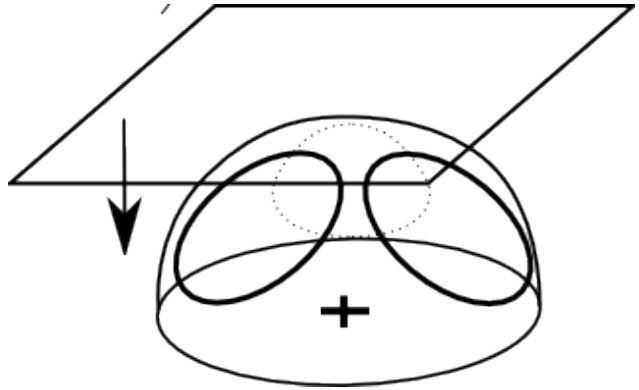



- Now modelized by a sphere.
- The real shape collision with a plane isn't so hard to compute.

Problem :
Difficulty to implement it in webots.

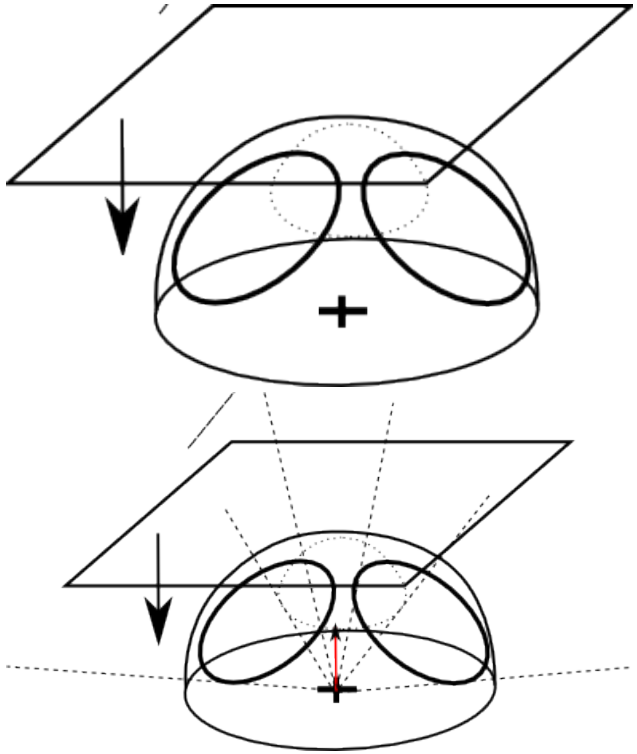
# Detection colision with a plane.

1. We project the center of the module on the plane.



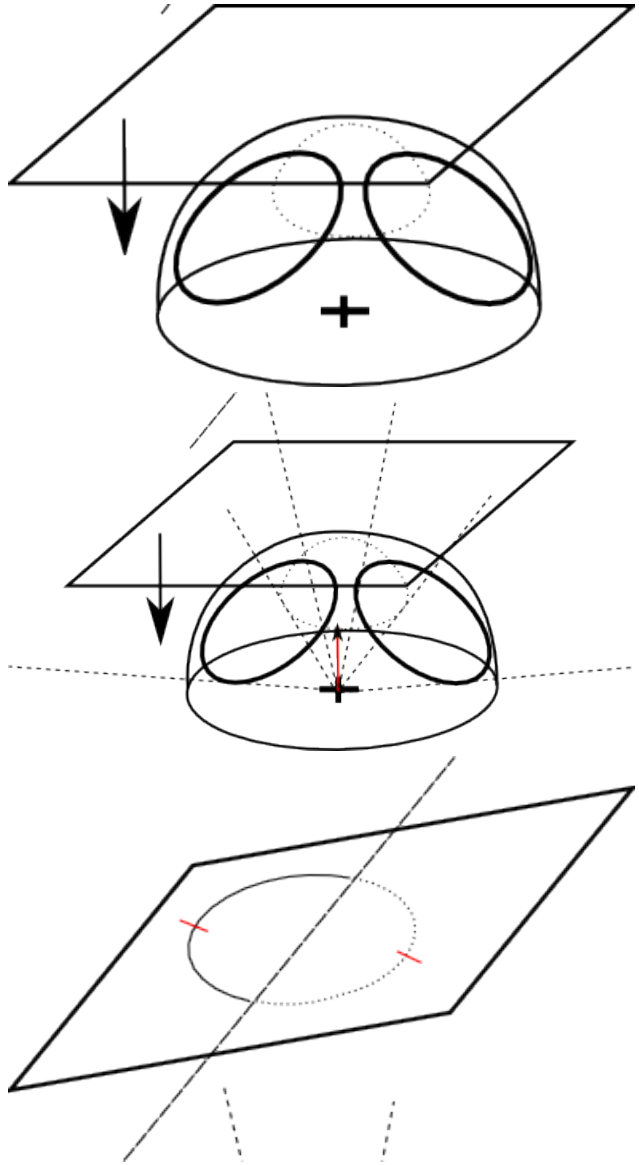
# Detection colision with a plane.

1. We project the center of the module on the plane.
2. We detect if we are in one of the three cones or in other half-space.



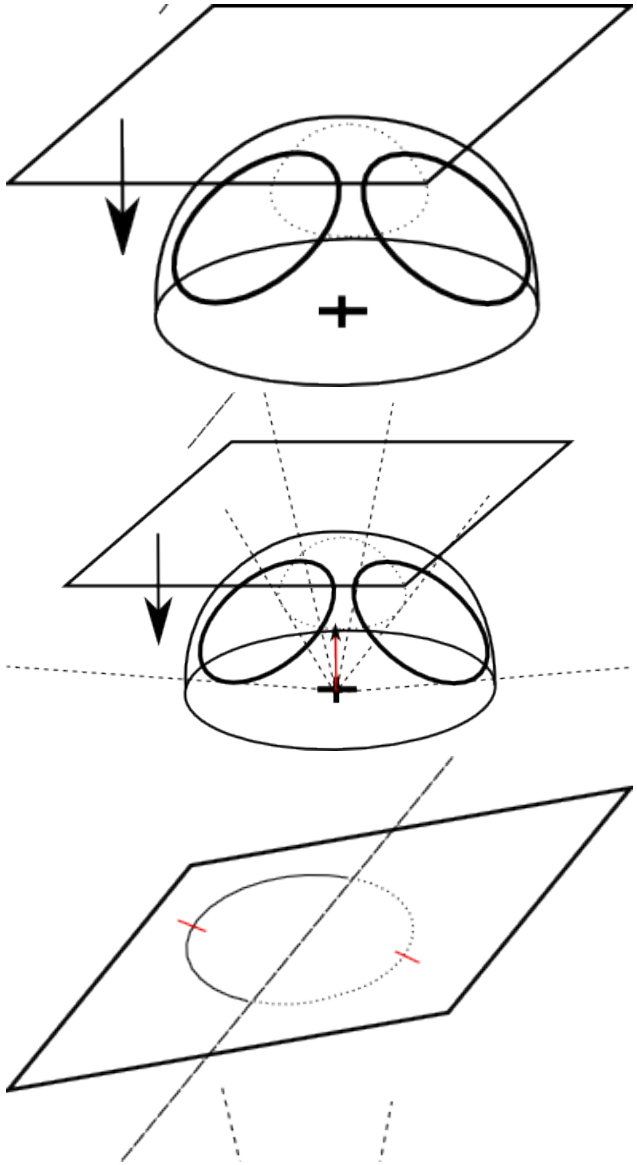
# Detection colision with a plane.

1. We project the center of the module on the plane.
2. We detect if we are in one of the three cones or in other half-space.
3. If in the cone we calculate the deepest and highest point of the face circle.



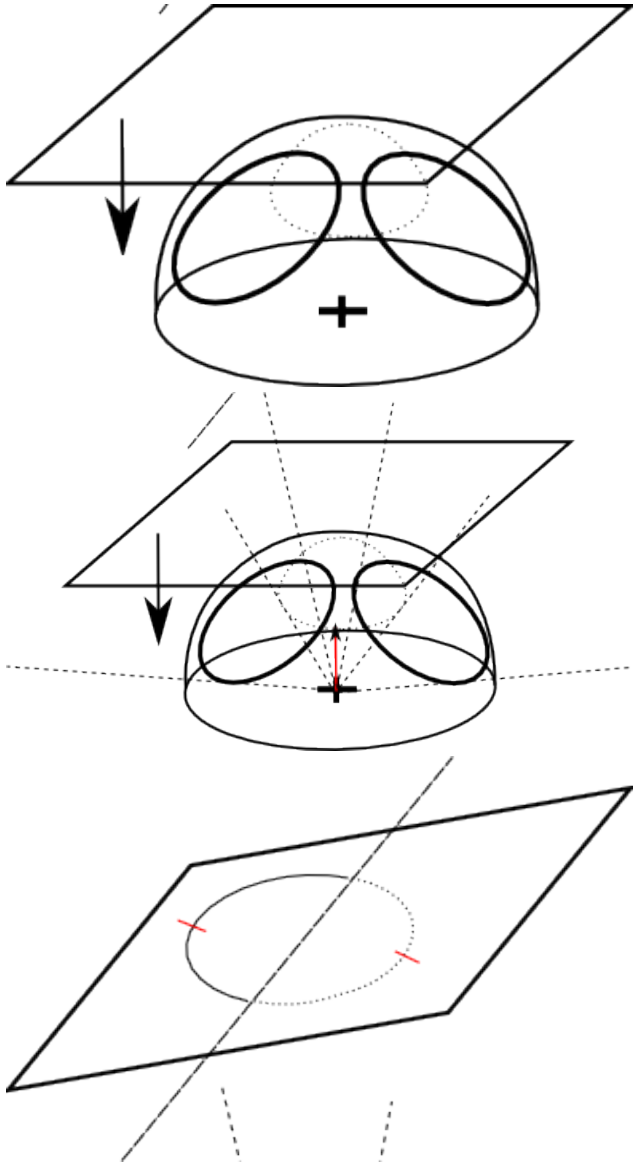
# Detection colision with a plane.

1. We project the center of the module on the plane.
2. We detect if we are in one of the three cones or in other half-space.
3. If in the cone we calculate the deepest and highest point of the face circle.
    1. If the highest is under the plane, we give back three point on the circle.



# Detection colision with a plane.

1. We project the center of the module on the plane.
2. We detect if we are in one of the three cones or in other half-space.
3. If in the cone we calculate the deepest and highest point of the face circle.
    1. If the highest is under the plane, we give back three point on the circle.
    2. If the highest is over the plane, then we just send back the deepest point of the circle.



# Work done

- Mainly have coded and debugged most of my framework.

# Work done

- Mainly have coded and debugged most of my framework.
- Attemp some first result with Powell algorithm and simple CPG this week.

# Questions ?