Modeling robot environment

First draft

1 Robots

1.1 Perception

Robot have the following information about environment from its sensors

- light sensors - adjacency to the boundary (white line),
- camera above the pitch - positions of all objects on the pitch.

1.2 Action

Robot must made the following decisions basing on its and the environment state:

- avoid the boundary,
- direction of movement,
- kick/pass the ball.

It is worth noting, that since there are 2 robots to optimize their collaboration and avoid synchronization problems, it would be beneficial to store the state on a Dice machine which would act as a command server to both the robots. The command center has more computational capabilities than robots themselves and can make decisions basing on the knowledge of both robots states, also it is the first that receives signal from the camera. Robots should have overwrite control only in situations like a risk of crossing the boundary, but it should be communicated back to the command server that it could change plan basing on that information.

1.3 State

The following is a proposition how the state of a game could be represented.

1. The pitch origin $O = (0, 0)$ initialized to one of the pitch corners.
2. Robot position on the pitch as a Point(double, double) from the origin O, please note the point also denotes the distance to the origin as it is measured from $(0, 0)$.
3. Robot orientation as an radian angle relatively to the origin O (*this is redundancy and can be computed using position and the origin*).
4. Enemy robot position as a Point(double, double) relatively to the origin.
5. Enemy robot orientation as an radian angle from measured from the origin (*this is redundancy and can be computed using position and the origin*).
6. (*optional*) Enemy robot velocity, namely tuple of magintude and angle (note this measurement must be based on few previous states, hence it could be represented as history vector with memory of example 10, where top most is the current velocity).
7. Ball position as a Point(double, double).
8. Ball velocity as a tuple of magnitude and radian angle.

9. (*optional*) Robot arm angle.
10. (*optional*) Points enclosing a robot area (convex hull).

1.4 Abstractions

It would be beneficial to use already existing abstraction and to develop it further when necessary, e.g. abstraction of movement, a robot could get information where it should go and in what time and robot would adjust the travel speed.