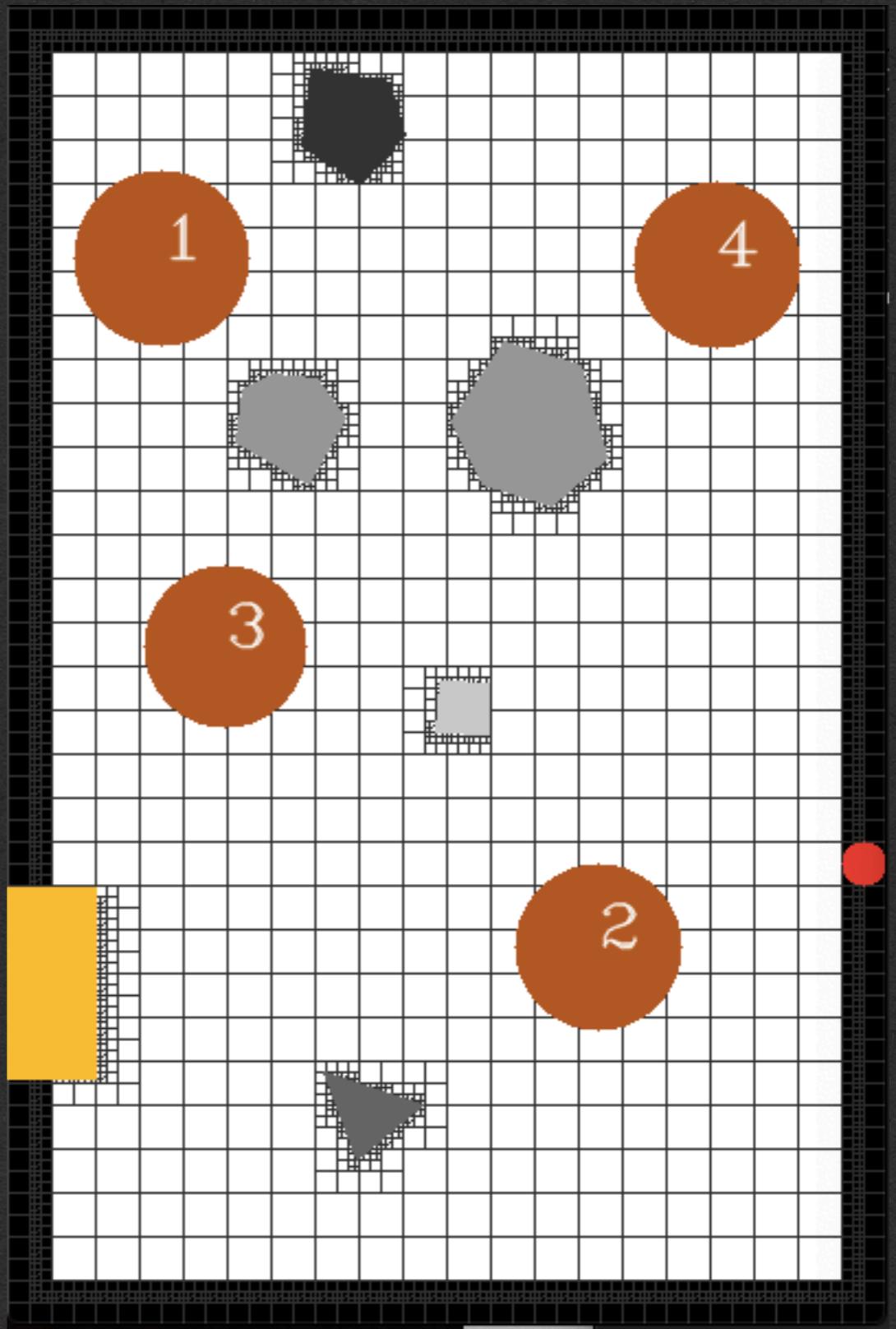


PathPlanning

Laboratory of Applied Robotics

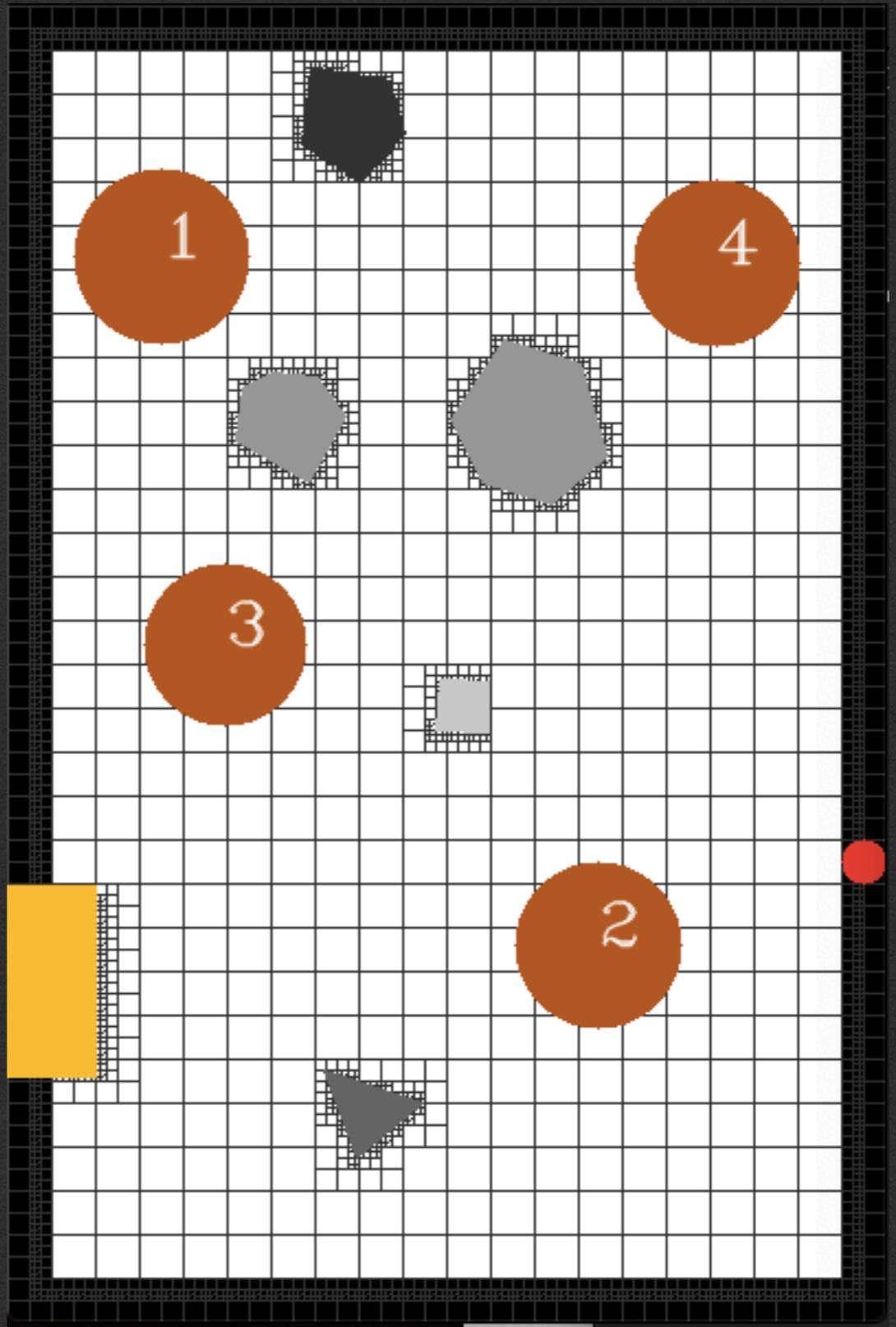
The Foundation

- grid based map
- adaptive mesh
- Character Recognition
- Object Recognition



The Foundation

- ▶ *Map*
- ▶ *Cell*
- LAR
 - ▶ *Arena*
 - ▶ *ExitPoint*
 - ▶ *People*
 - ▶ *Obstacle*



Connecting the dots..

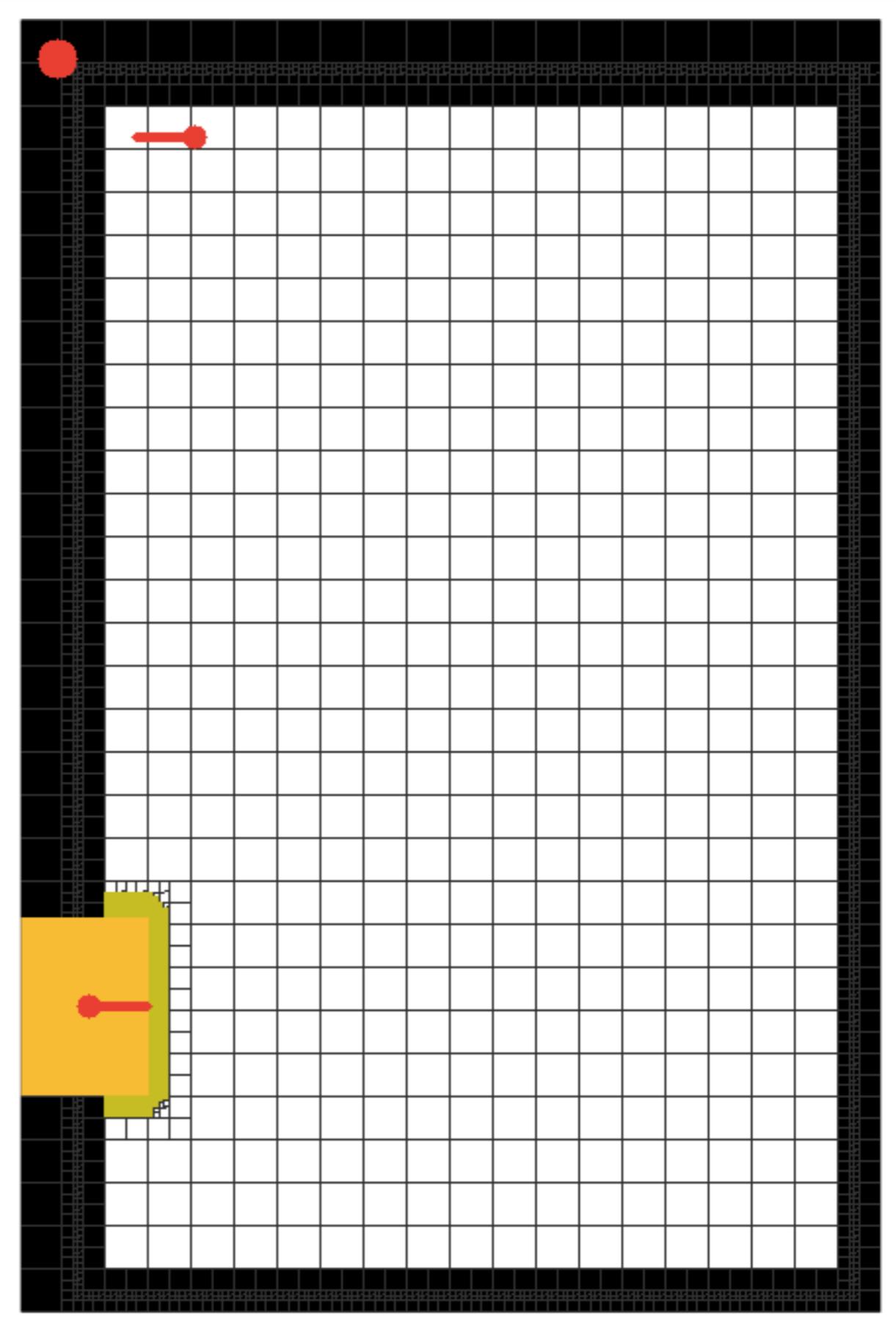
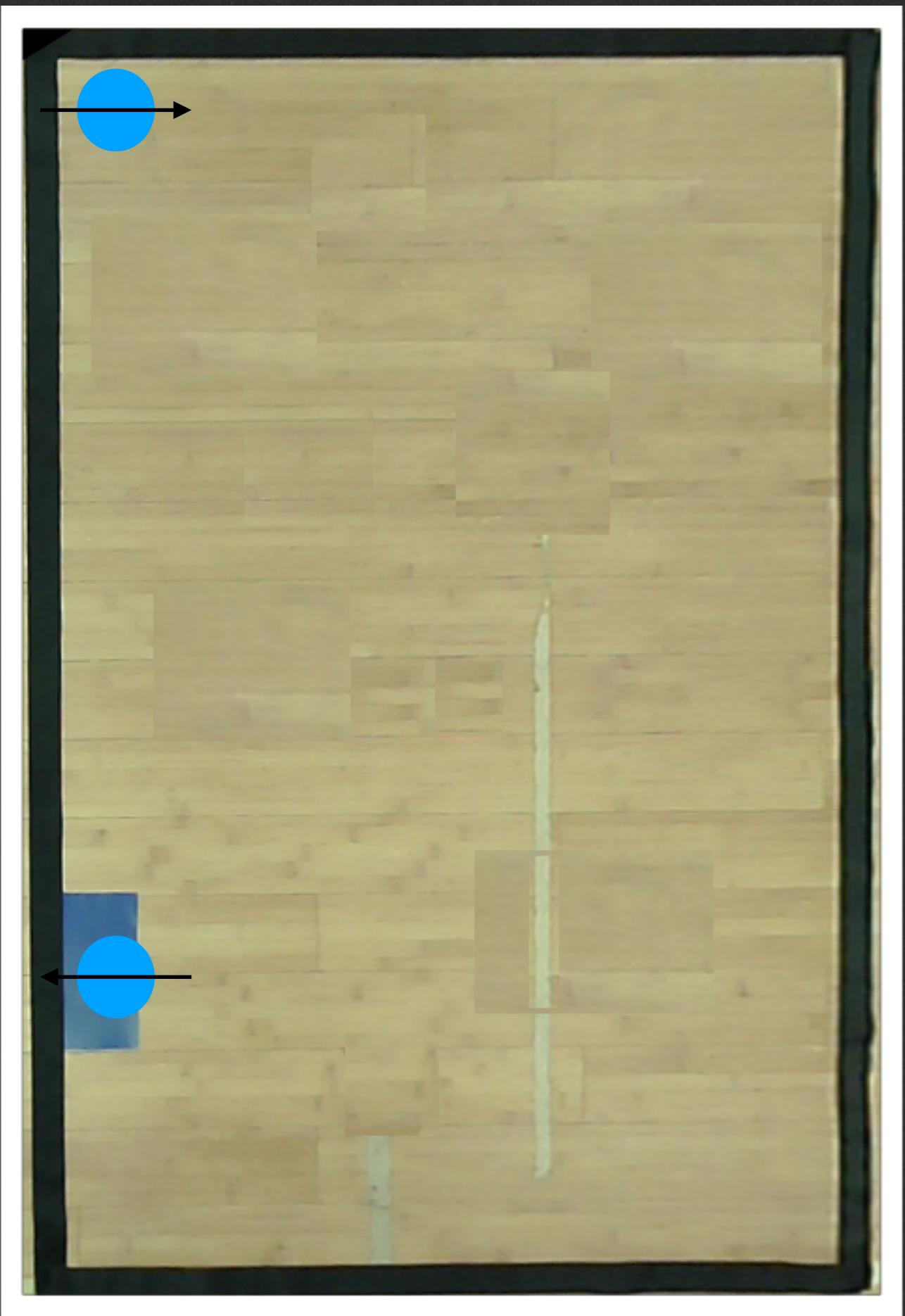
Elements



Position

int	x_coordinate
int	y_coordinate
double	orientation





Connecting the dots..

Elements

Position



Line

Position
Position
double

start_point
end_point
length

Position

Position

Connecting the dots..

Elements

Position



Line

Position

Position

double

vector<Point>

start_point

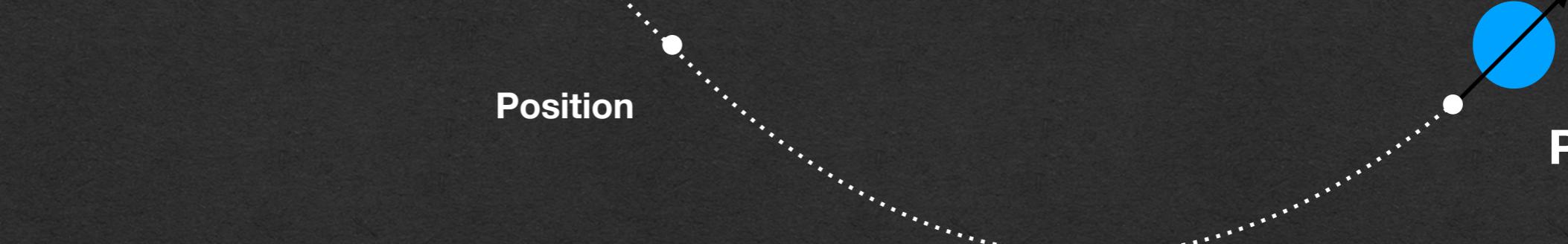
end_point

length

inter_points

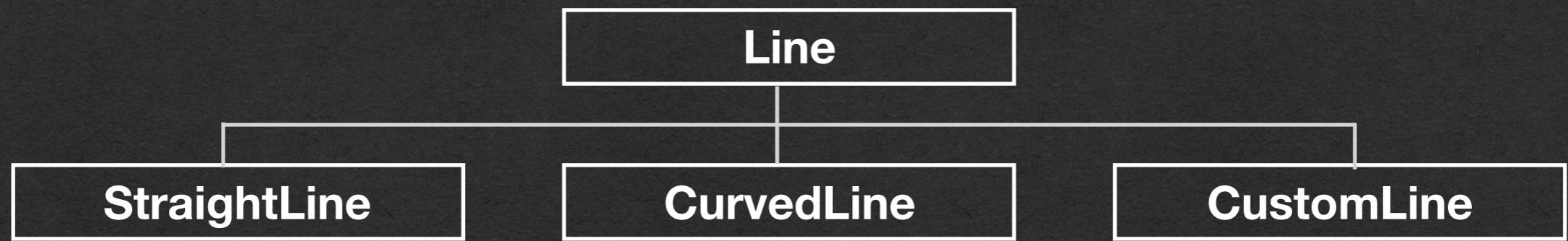
Position

Position



Connecting the dots..

Elements



Connecting the dots..

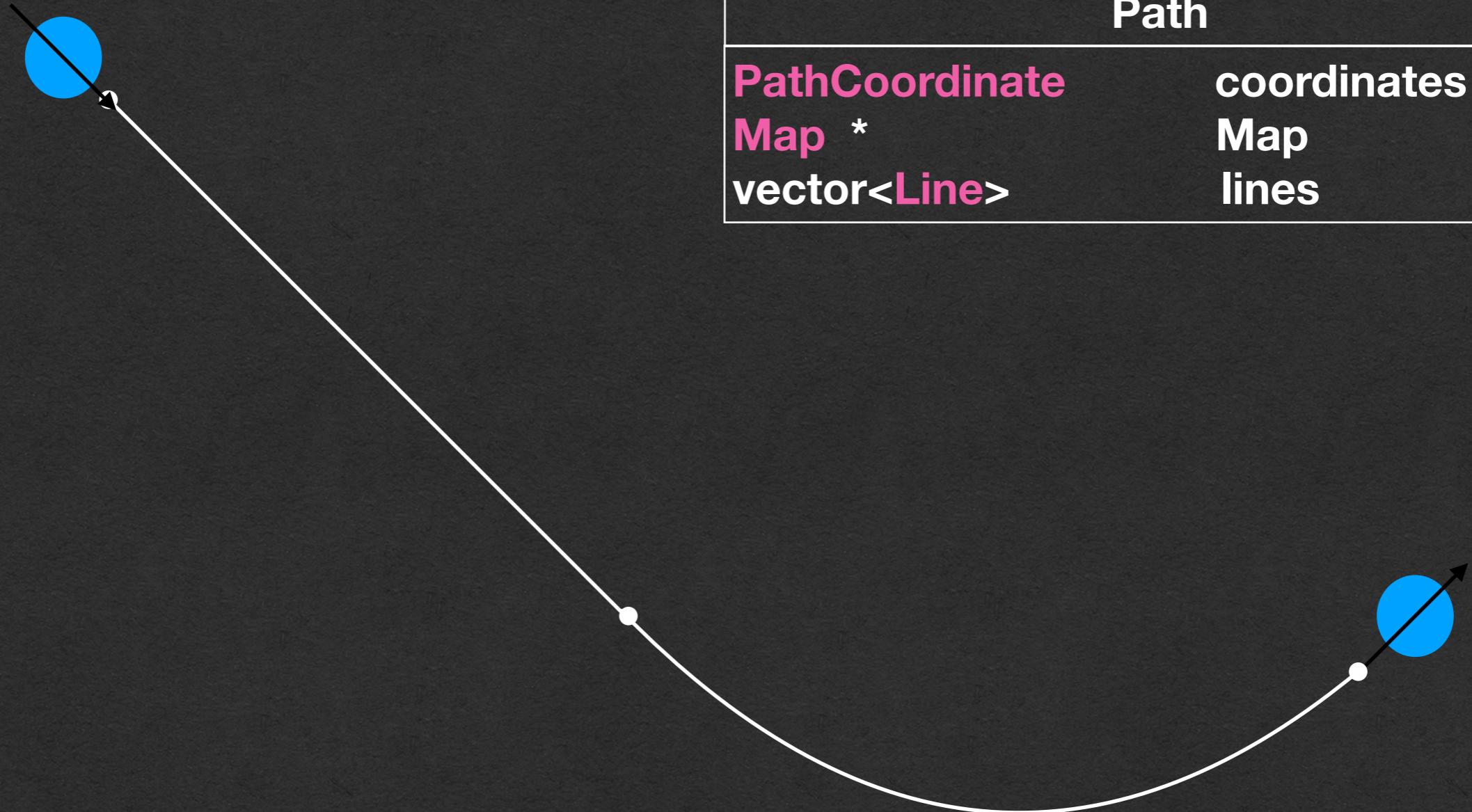
Elements

PathCoordinates

Position	start_point
Position	end_point
double	curvature

Connecting the dots..

2DPath



Connecting the dots..

2DPath



Connecting the dots..

2DPath



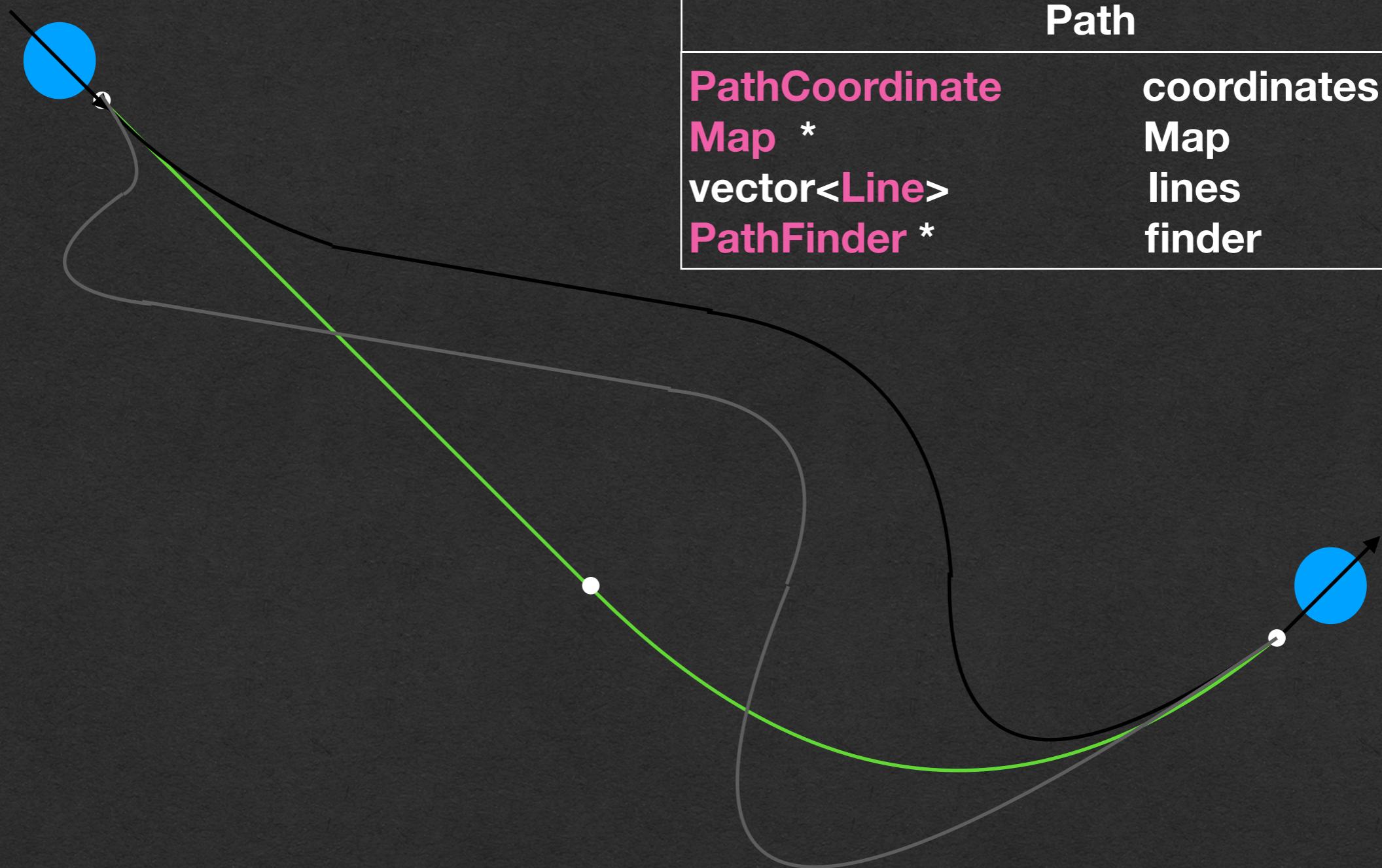
Path

PathCoordinate
Map *
vector<Line>
PathFinder *

coordinates
Map
lines
finder

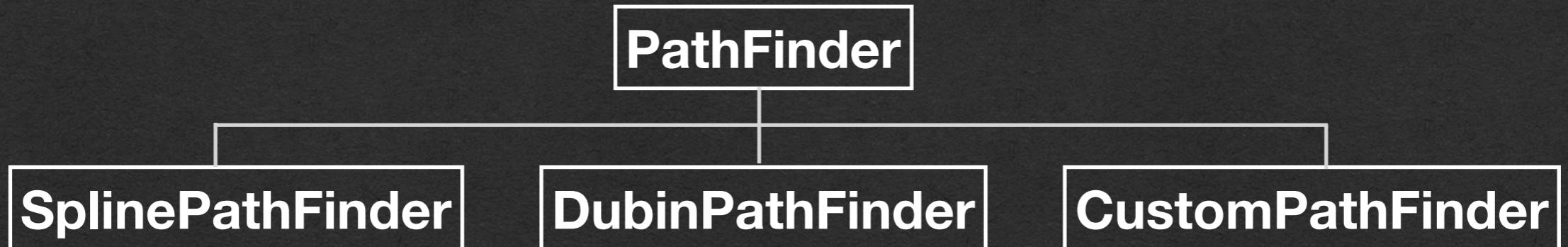
Connecting the dots..

2DPath



Connecting the dots..

2DPath



```
std::vector< Line > shortestPath (std::vector< cv::Point > &alternative_Points)
```

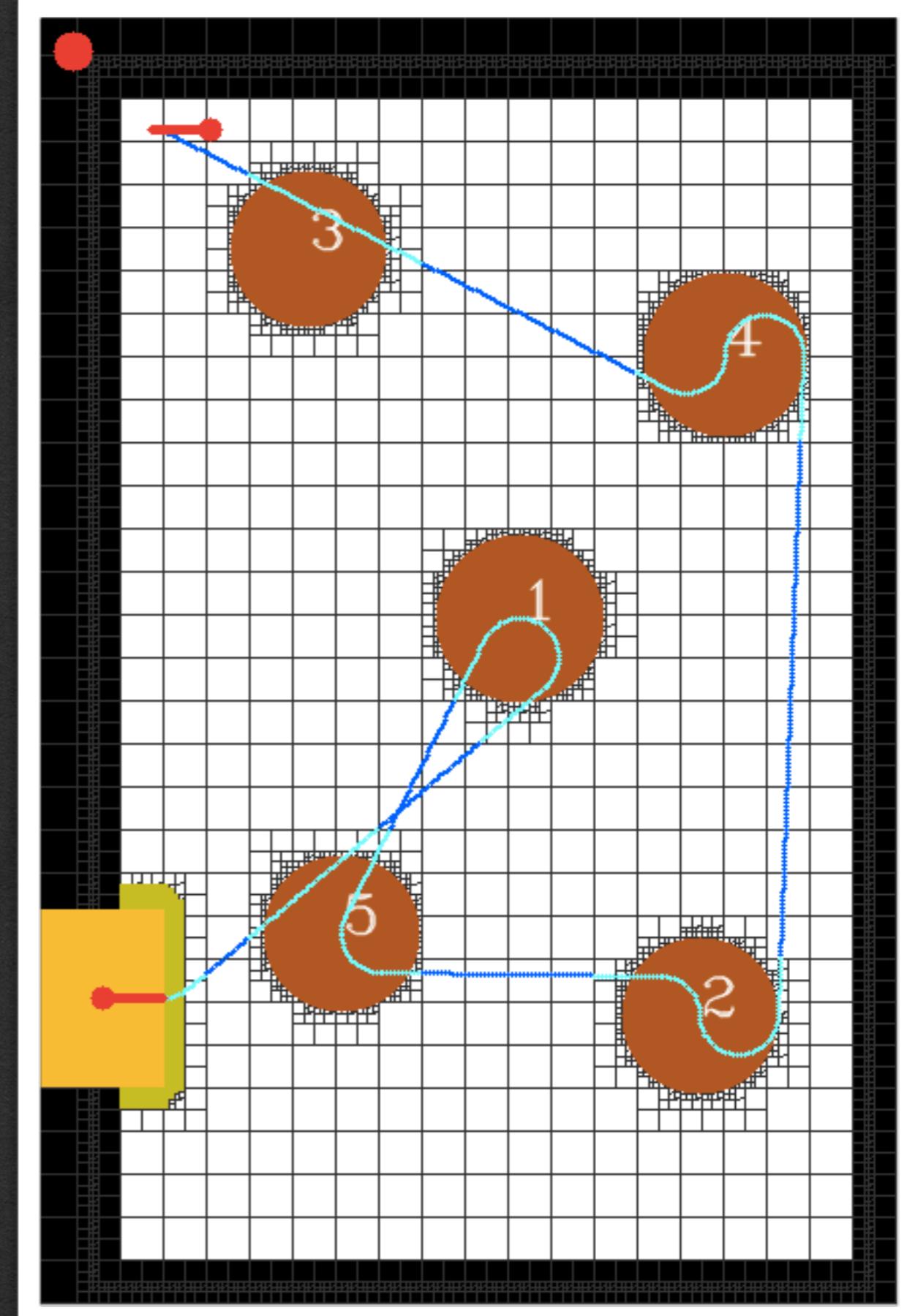
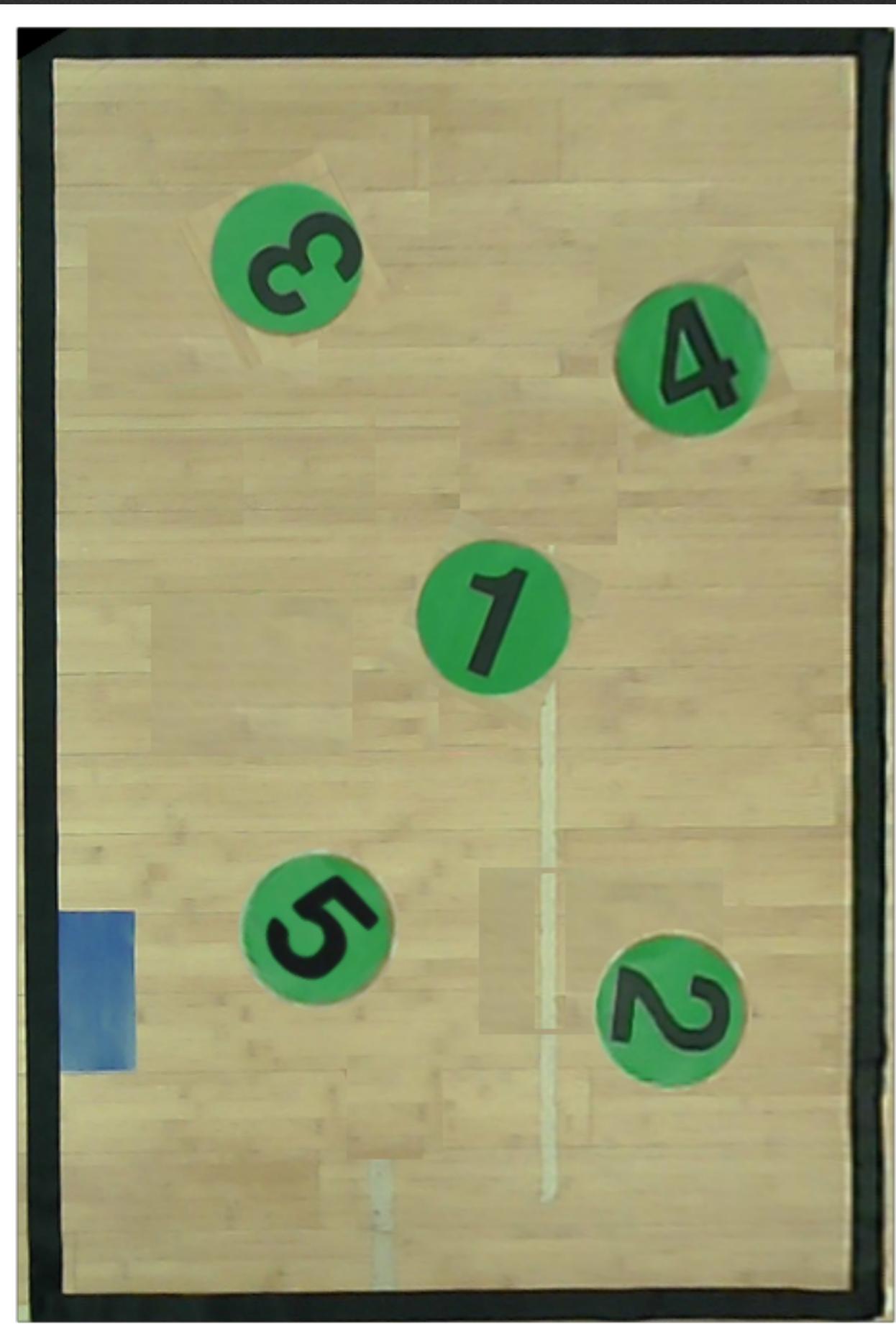
Connecting the dots..

2DPath

Constructing a Path
using 2 complete path objects

```
Path(Path path1, Path path2);
```

1. start pos = path1.start_pos
2. end pos = path2.end_pos
3. add lines
4. check for G1 continuity



Connecting the dots..

2DPath

Constructing a new Path object

```
template <class T>
    explicit Path(PathCoordinates coordinates,
                  Map *map,
                  T* pathFinder,
                  bool complex = true);
```

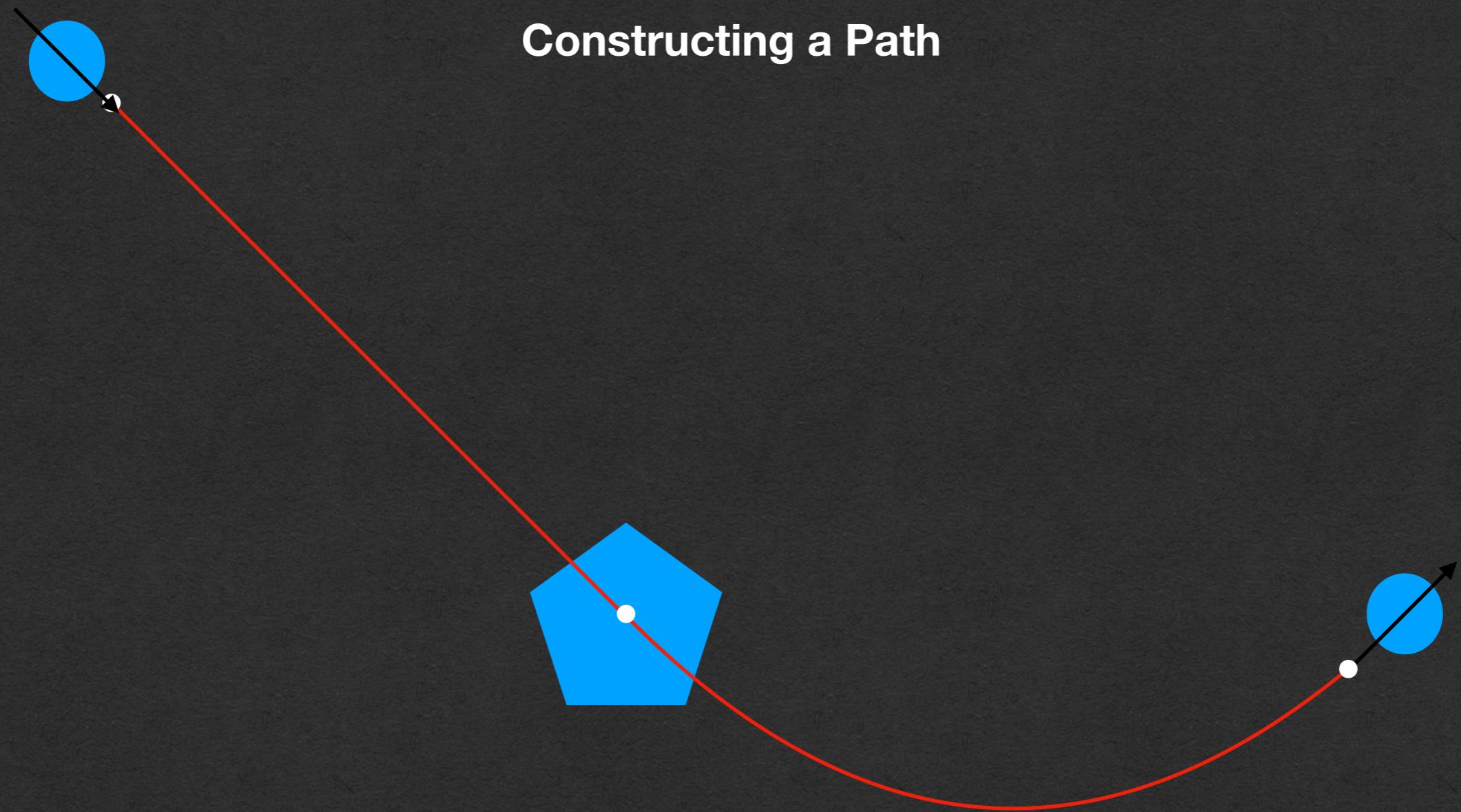
1. assigns start & end points
2. assigns curvature
3. assigns pointer to map
4. assigns a PathFinder class type

complex flag decides if the finder should continue to look for solutions if no path was found

Connecting the dots..

2DPath

Constructing a Path



Connecting the dots..

2DPath

