CIBot Square Agent Description

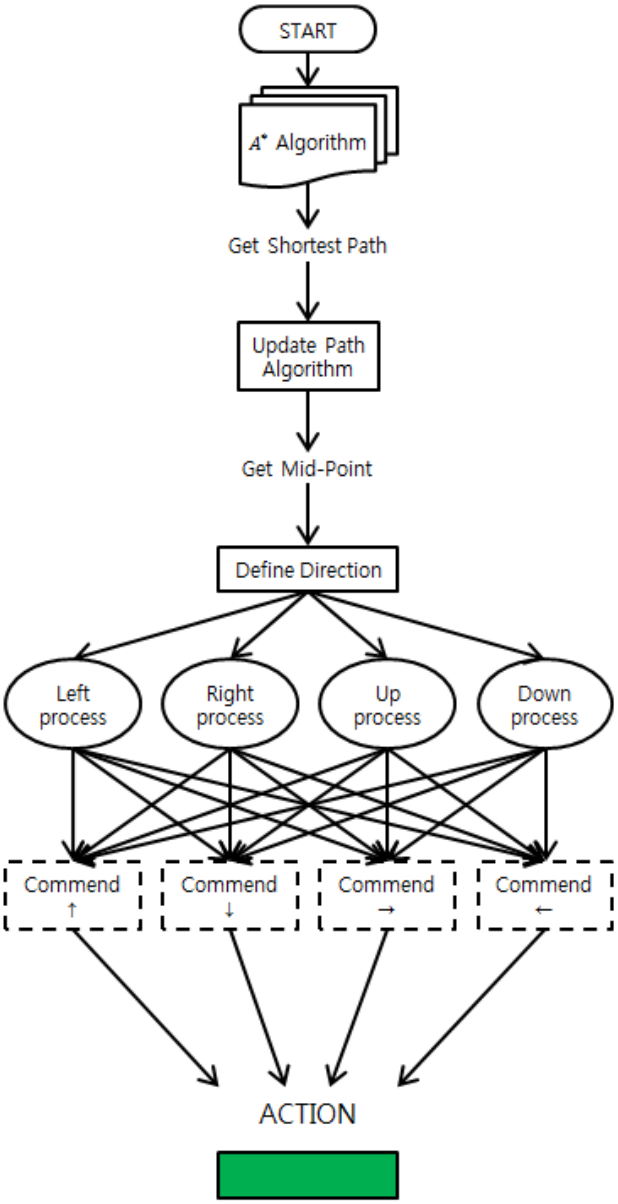

Figure 1. Agent Flow Diagram

The Square Agent uses A-star Algorithm basically. At first, the map is divided into nodes of certain size. The nodes recognize obstacles, agent and diamonds. The start node is agent. The target node is one of diamonds. The agent realizes the shortest path to the diamond using A-star Algorithm. If there are a lot of target nodes(diamonds), A-star Algorithm is performed each target node. So, agent find the nearest target node(diamond) and move to target node along a

path which was obtained by A-star.

A-star Algorithm informs the agent of the optimal path always. But, this path create some problems to apply real game. One of problems is that A-star creates jagged path across the sky. In that case, the agent can't move along the path. We change movement cost from current node to the next node in all directions. $W_{horizon}$ is weight(movement cost) to go to left or right node. W_{up} is weight to go to top node. W_{down} is weight to go to bottom node. Agent is able to get smooth path through changing weights. The changing rule is presented as follows:

$$W_{down} < W_{horizon} < W_{up}$$

That is another problem that agent's speed is very slow if agent only moves along the path, because the agent has to recognize perfectly each node along the path. This method is very inefficient in order to quickly clear the game. So, we create Update-Path Algorithm.

```
// Update-Path Algorithm pseudo code
# Case_A
If ( Agent is Current Node )
    While ( Path.count != 0 ) do
      If ( Agent.y == NextNode.y)
        Mid-point ← NextNode;
      Remove Path(-1);
    End While
# Case_B
If ( Agent's around is Current Node )
    While ( Path.count != 0 ) do
      If (Agent's around is Current Node && CurrentNode.x == NextNode.x)
        Mid-point ← NextNode;
      Remove Path(-1);
    End While
If ( !Case_A && !Case_B) Mid-point ← CurrentNode;
Return Mid-point
```

The agent finds out the mid-point that can go as far as possible from current location using Update-Path Algorithm. If agent gets mid-point, it decides the direction by calculating distance between mid-point and current location. And it performs each process along this direction. Each process output command depending on each situation.