**Use Case of updateStats**

**Actor**: gameMaster

**Description**:

The gameMaster would track and update info of the robot after it's behavior in the round, based on their behavior, including individual result, the damage, the status, the distance travelled and shots fired. Meanwhile, the gameMaster would always check whether the match has ended. If so, the team result would be updated at the end of match;

**Preconditions:**

After the behavior of each robot;

After click “switch” button;

**Primary Scenario:**

1. Initialization: initial statistics for all robots on the game boards:

* individual result = "survived"
* team result = "-"
* damage inflicted = 0
* damage absorbed = 0
* distance travelled = 0
* shots fired = 0
* Status = "unbehaved";

1. during the behavior of robot A which is alive:

* if A do nothing:
* the gameMaster does nothing;
* the A's status would be updated to "behaved" after click "switch".
* if A is just moving:
* update the distance travelled of A, adding by A's moving spaces;
* if distance travelled of A in current round has been equal to Maximum Movement value of robot A, then robot A cannot move in current round;
* the A's status would be updated to "behaved" after click "switch".
* if A is just shooting at B:

For robot A, we should

* update damage inflicted, adding by A's attacking value;
* update shot fired of A, adding by 1;
* then Robot A cannot shoot at others in this round;
* the A's status would be updated to "behaved" after click "switch".

For robot B, we should:

* update the damage absorbed, adding by A's attacking value;
* If the current damage absorbed is not less than health value of B, then change individual result of B to "destroyed";
* if A is moving and shooting at robot B:

For robot A, we should

* update damage inflicted, adding by A's attacking value;
* update distance travelled of A, adding by A's moving spaces;
* if distance travelled of A in current round has been equal to Maximum Movement value of robot A, then robot A cannot move in current round;
* update shot fired of A, adding by 1;
* the A's status would be updated to "behaved" after click "switch".

For robot B, we should:

* update the damage absorbed, adding by A's attacking value;
* if the current damage absorbed is not less than health value of B, then change individual result of B to "destroyed";

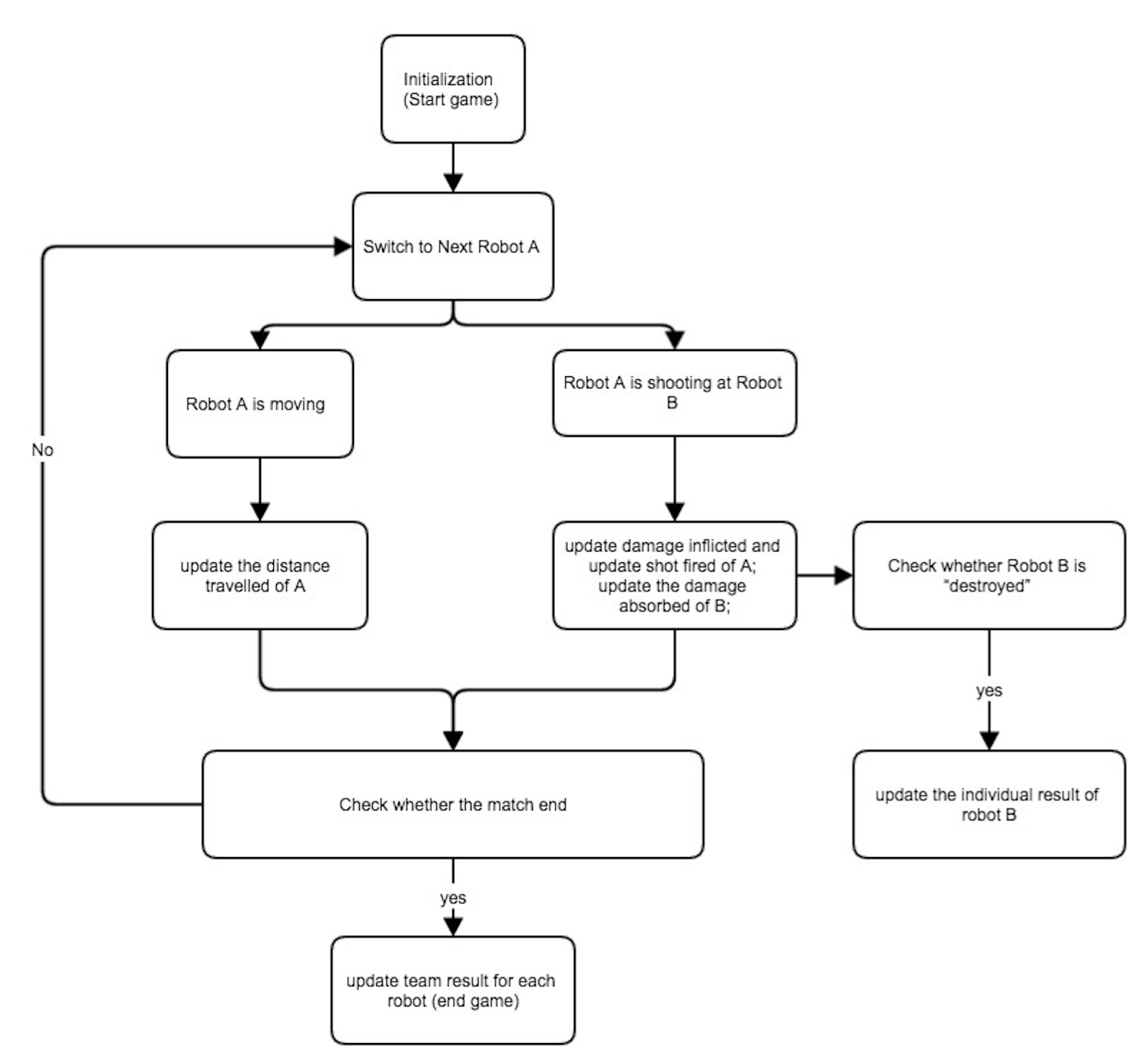
1. For the new turn:

when gameMaster has perceived "new turn", the damage inflicted, distance travelled, Status and shots fired of all survived robots would be initialized again.

1. For the Match end:

When gameMaster has perceived that there existed only one team on the boards, the match has ended. Then gameMaster updated the team result for all robots in the match, "win" for all robots of last survived team and "lost" for others robots.

**Activity Diagram:**

****

**Extension Scenario:**

1. If A is shooting at one cell which has more than one robots:

for robot A:

* update damage inflicted, adding by (A's attacking value \* the number of robots in target cell);
* if A moved, then update distance travelled of A, adding by A's moving spaces;
* update shot fired of A, adding by 1;
* the A's status would be updated to "behaved" after click "switch".

for other robots in the same cell:

* update the damage absorbed, adding by A's attacking value;
* if the current damage absorbed is not less than its health value, then change its individual result to "destroyed";

1. if robot A and other robots are in same cell, then A shoots at them:

for robot A:

* update damage inflicted, adding by (A's attacking value \* the number of robots in this cell, including itself);
* if A moved, then update distance travelled of A, adding by A's moving spaces;
* update shot fired of A, adding by 1;
* update the damage absorbed, adding by A's attacking value;
* if the current damage absorbed is not less than its health value, then change its individual result to "destroyed";
* the A's status would be updated to "behaved" after click "switch".

for other robots in the same cell:

* update the damage absorbed, adding by A's attacking value;
* if the current damage absorbed is not less than its health value, then change its individual result to "destroyed";

1. if there is no robot left at the end of match, which means the robots of different team are destroyed at the same time:

* Then gameMaster would update the team result "lost" for all robots in the match.

**Post conditions:**

The info for each robot has been updated after behavior of each robot in the round, including: individual result (survived or destroyed), team result (win or lost or unknown), the damage (inflicted and absorbed), status, distance travelled and shots fired;

**Use Case of Switch:**

**Actor:** Player, AI and gameMaster

**Description:**

The player or AI click on "Switch" button, then ending the behavior of current robot and switching to the next team's survied robot which has the highest movement value and does not behave in this turn;

**Precondition:**

Player or AI click on "Switch" Button;

**Primary Scenario:**

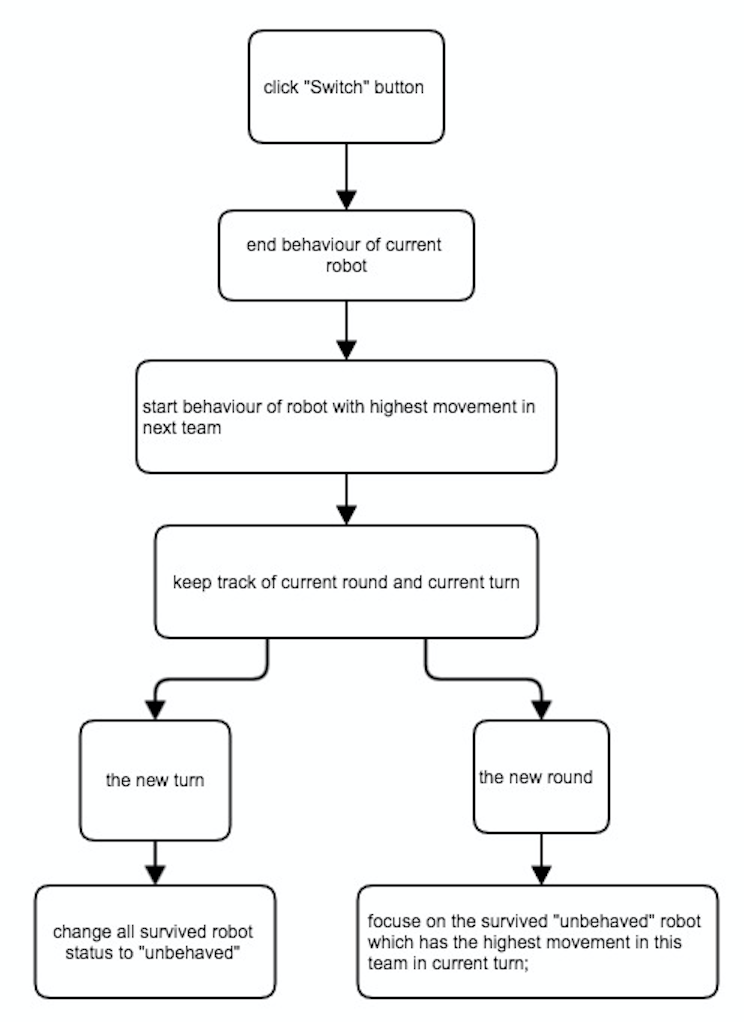
(1) click on the "Switch"

(2) ending the behavior of current robot;

(3) Starting the behavior of next robot which has the highest movement value and does not behave in this turn;

(4) keep track of current round and current turn;

**Activity Diagram:**



**Secondary Scenario:** none;

**Extension Scenario:**

1. If all robot behaved or died in current round in current team, skip this round and switch to next team;
2. It may start the new round or new turn:

* for new round: focusing on the survived "unbehaved" robot which has the highest movement in this team in current turn;
* for new turn: change all survived robot status to "unbehaved"

**Post conditions:**

Ending the behavior of current robot and switching to the next team's survived robot which has the highest movement value and does not behave in this turn;