# Actions

Java agents should extend BW4TAgent which contains methods to perform all the following actions.

1. goTo(X, Y)
   * Use this to travel to a specific location in the environment
2. goTo (ID)
   * The ID can be a room, block or navPoint ID.
3. pickUp(blockId)
   * Can only pick up a block when on that block
4. putDown
5. sendMessage(String message, String receiver)
   * receiver can be either “all” (then the message will be sent to all other agents) or the ID of another agent
   * BW4TAgent has a method that can send a BW4TMessage to a certain receiver, Java agents should use this method to send messages
   * GOAL agents should use the internal send action instead

Differences with BW4T:

* Additional goTo action for going to a specific position in the environment, other goTo is now called goToNavPoint in implementation but will be renamed to goTo to correspond with old version
* pickUp action requires a blockId in BW4T2
* sendMessage arguments are reversed in BW4T2

# Percepts

Percepts are received automatically in the following structure for GOAL agents.

Java agents should extend BW4TAgent and should call getPercepts() to receive all percepts. Getting the parameters of each percept should be handled by the agents themselves.

1. at(ID, X, Y)
   * This percept is sent for each room (**including** the dropzone) in the environment
   * ID = the ID of the room
   * X = the X position of the room in the environment
   * Y = the Y position of the room in the environment
2. block(ID)
   * This percept is sent for each block in the environment
   * ID = the ID of the block
3. room(ID)
   * This percept is sent for each room (**excluding** the dropzone, see dropZone percept) in the environment
   * ID = the ID of the room
4. dropZone(ID)
   * ID = the ID of the dropzone
5. robot(ID)
   * ID = the ID of the robot
6. color(ID, Color, X, Y)
   * This percept is sent for each visible block (**including** the block the agent is holding)
   * ID = the ID of the block
   * Color = the color of the block
   * X = the X position of the block in the environment
   * Y = the Y position of the block in the environment
7. holding(ID)
   * ID = the ID of the block
8. sequence(color1,color2,…,colorn)
   * This represents the sequence of colors that is still needed to complete the mission
   * color1,color2,…,colorn = A color
9. occupied(ID)
   * This percept is sent for each room (**including** the dropzone) that is occupied
   * ID = the ID of the occupied room
10. navpoint(ID, X, Y)
    * This percept is sent for each navPoint
    * ID = the ID of the navPoint
    * X = the X position of the navPoint in the environment
    * Y = the Y position of the navPoint in the environment
11. receiveMessage(sender,message)
    * This percept is sent for each received message
    * sender = the sender of the message (name of the agent)
    * message = the message, can be translated to BW4TMessage by using MessageTranslator.translateMessage
    * GOAL agents should not receive these percepts
12. state(state)
    * This percept contains the state of the robot
    * Can be traveling, collided or arrived

Differences with BW4T:

* Room percept replaces the old place percept (could be renamed to make it identical)
* in(<PlayerID>, <PlaceID>) is not implemented in BW4T2, could be added quite easily
* at(<PlayerID>, <BlockID>) is not implemented yet in BW4T2, could be added if necessary
* color(<BlockID>, <ColorID>) is replaced with color(ID, Color, X, Y) because X and Y position are needed for visualization
* holding(<PlayerID>,<BlockID>) is replaced with holding(ID), as player information is not necessary as it is only sent to that player himself
* player(<PlayerID>) is not yet implemented in BW4T2, could be added easily
* sequenceIndex(<Integer>) is not implemented in BW4T2 as it is implicitly contained in the sequence percept as that list will shrink when blocks have been found
* blocked percept is replaced with state(State) of agent as that can return collided