

# Mission Repository: Missions in Malmo (14 June 2016)

Missions can be generated by coding them in any of the supported languages, but they are usually generated automatically using .XML files.

- Some .XML files can generate one single mission, if no decorators are used or a fixed seed is specified.
- Most .XML files generate many instances, because there is a generator and a variable seed. Another way of generating several instances is through an xml template.

The instances that are generated, when recording is enabled, can be written in the saved\_data.tgz file, so one can always know which instance has been used. The tasks in the repository below are all accompanied with an instance trace and video in a compressed ".tgz" file with the same name as the task.

Id	Type	Instances	Description	Goal	Rewards	End	Observability	Actions	States	Ergodicity	Noise	#Agents	Breadth	Difficulty	Experimentation	Implementation	Date	Link
default_world_1	Survival	Infinite instances.	Survive and find gold, diamond or redstone. The agent appears in a default Minecraft world, with all possible objects. The agent appears at x=-204 y=81 z=217, which depending on the world that is generated means that it's going to fall initially to touch ground.	The goal must be learnt from rewards. The task instance is considered complete if the agent finds (mines) any special block (diamond_block, redstone_block).	Sparse rewards. Negative if dead (-10000), positive if the target block is touched (+1000) and negative if the time is over (-1000).	Task ends if the timeLimitMs=300000 (300 sec) is reached or agent killed.	Partial observability. Using VideoProducer (480x320)	Possible actions given by "ContinuousMovementCommand" included in "survival mode". Inventory empty initially. So actions are "move", "strafe", "pitch", "turn", "jump", "crouch", "attack", "use" and "hotbar.X". Action "drop" not included <a href="#">[issue #73]</a> .	Infinite no. of states	Non-ergodic	Noisy (stochastic, as there are animals, creepers, clouds...)	Single agent. Animals and creepers are considered NPC.	Very broad.	Difficult because of sensorimotor and conceptual complexity of environment and the unlikely goal. As the goal blocks are very rare, the chance of being killed or starve are high if not an experienced player.	Interesting for general agents using intrinsic motivation or knowledge-seeking utility functions, and humans. No experiments performed yet. The "default_world_test.py" example uses a very similar mission.	Created with a "DefaultWorldGenerator".	2016-06-12	<a href="https://github.com/Microsoft/sample_missions/tree/master/default_world_1.xml">https://github.com/Microsoft/sample_missions/tree/master/default_world_1.xml</a>
default_flat_1	Survival	Infinite instances because of noise, but the positions are fixed. It might inherit things from previous experiments as the "force reset" is not included <a href="#">[issue #73]</a>	Move to a wooden hut in a snow tundra. The agent appears in a snowy flat landscape and has to survive. The agent appears at x=0 y=227 z=0.	The goal must be learnt from rewards. The agent has to reach x=19.5 y=227 z=19.5, which is inside a hut that is relatively near.	Sparse rewards. Positive +100 given if the goal position is reached.	Task ends if the timeLimitMs=100000 (100 sec) is reached or agent killed.	Partial observability (although the hut is visible from the beginning). Using VideoProducer (320x240)	Possible actions given by "ContinuousMovementCommand" included in "survival mode". Inventory empty initially. So actions are "move", "strafe", "pitch", "turn", "jump", "crouch", "attack", "use" and "hotbar.X". Action "drop" not included <a href="#">[issue #73]</a> . Inventory empty initially. Only moving on the flat surface required.	Infinite no. of states	Non-ergodic	Noisy (stochastic, as there are animals, creepers, clouds...)	Single agent. Animals and creepers are considered NPC.	Very broad.	The difficulty relies on sensorimotor skills as the goal is close, the chance of being killed or starve is small.	Interesting for image recognition and very basic navigation (aiming at a visible goal).	Created with a "FlatWorldGenerator" and a "DrawingDecorator".	2016-05-20	<a href="https://github.com/Microsoft/sample_missions/tree/master/default_flat_1.xml">https://github.com/Microsoft/sample_missions/tree/master/default_flat_1.xml</a>
tricky_arena_1	Pattern recognition (colour) on an environment.	One Instance* (This is just one instance that can be generated using the program: "reward_for_mission_end_test.py" example) or changing some parameters of the XML document. It might inherit things from previous experiments as the "force reset" is not included <a href="#">[issue #73]</a>	Mind your step to the redstone! A flat area of 40x40 blocks with some of the blocks being special (goals) or dangerous (lava and water).	The goal must be learnt from rewards. Touching one of the goal blocks (redstone).	Dense rewards. Positive (+100) if agent steps over obidian tiles that haven't been stepped over recently, positive (+400) if goal stone is touched, positive (+100) if out of the arena, negative (-800) if agent falls into the water, negative (-900) when time runs out, negative (-1000) if agent dies (e.g., falling into the lava).	Task ends if the timeLimitMs=15000 (15 sec) is reached or agent killed.	Partial observability. Using VideoProducer (640x480)	Possible actions given by "ContinuousMovementCommand" included in "survival mode". So actions are "move", "strafe", "pitch", "turn", "jump", "crouch", "use" and "hotbar.X". Action "attack" is forbidden with a "deny-list" <a href="#">[issue #73]</a> . Inventory empty initially. Only moving on the flat surface required.	Finite no. of states as getting out of the arena ends the game.	Ergodic, as attack is not allowed. As new stepped blocks give reward there might seem that it is not ergodic, but the reward will be enabled again after some time.	No noise (deterministic), as time, weather and spawning fixed, no clouds	Single agent	Narrow task. Recognising the objects and moving fast may get good results.	The difficulty relies on sensorimotor skills. Image recognition is crucial here to tell some objects from others, and memory	Interesting for image recognition systems, vision systems, etc. No experiments performed yet. The "reward_for_mission_end_test.py" example uses a very similar mission.	Created with a "FlatWorldGenerator" and a "DrawingDecorator"	2016-06-12	<a href="https://github.com/Microsoft/sample_missions/tree/master/tricky_arena_1.xml">https://github.com/Microsoft/sample_missions/tree/master/tricky_arena_1.xml</a>
eating_1	Pattern recognition (colour) on an environment.	One Instance* (This is just one instance that can be generated using the program: "reward_for_items_test.py" example) or changing some parameters of the XML document. It might inherit things from previous experiments as the "force reset" is not included <a href="#">[issue #73]</a>	Eat a healthy diet! A flat world where some food objects are scattered near the starting position. The good ones must be eaten but not the bad ones.	The goal must be learnt from rewards. Eating as much as good food items as possible.	Dense rewards. Positive (+2) if agent eats any of "fish porkchop beef chicken rabbit mutton", positive (+1) if agent eats any of "potato egg carrot", negative (-1) if agent eats any of "apple melon" and negative (-2) if agent eats "sugar cake cookie pumpkin_pie".	Task ends if the timeLimitMs=15000 (15 sec) is reached or agent killed.	Partial observability. Using VideoProducer (480x320)	Possible actions given by "ContinuousMovementCommand" included in "survival mode". So actions are "move", "strafe", "pitch", "turn", "jump", "crouch", "use" and "hotbar.X". Action "attack" is forbidden with a "deny-list" <a href="#">[issue #73]</a> . Inventory empty initially. Only moving on the flat surface required.	Infinite no. of states (the world is not constrained)	Non-ergodic, as "drop" is not allowed. <a href="#">[issue #73 (no drop action)]</a>	No noise (deterministic), as time, weather and spawning fixed, no clouds	Single agent	Narrow task. Recognising the objects and moving fast may get good results.	The difficulty relies on sensorimotor skills. Image recognition is crucial here to tell some objects from others, and memory	Interesting for image recognition systems, vision systems, etc. No experiments performed yet. The "reward_for_items_test.py" example uses a very similar mission.	Created with a "FlatWorldGenerator" and a "DrawingDecorator"	2016-06-12	<a href="https://github.com/Microsoft/sample_missions/tree/master/eating_1.xml">https://github.com/Microsoft/sample_missions/tree/master/eating_1.xml</a>
cliff_walking_1	Navigation	One Instance* (Starting position is fixed: x=4, and y=6 and z=1, but in the "tutorial_6mine.py" several lava blocks are randomly placed inside the arena using the code, making it more difficult. This code might generate three lava blocks in a row, making it necessary to jump, or (unlikely) might even create so many lava holes to make the instance impossible).	Burning level A cliff of size 12x3 sandstone blocks surrounded by burning lava. Agent must go from starting block to destination block without following in the lava.	The goal must be learnt from rewards. The task instance is considered complete if the agent goes from starting position ("cobblestone") and steps over a special destination block (blue "lapiz_block") without following in the lava.	Sparse rewards. Negative if lava (-1000), positive if target block ("lapiz_block") touched (+1000) and negative (-1) for any action done. No (negative) reward if out of time.	Task ends if either when fallen in the lava or the destination block is reached. Or if the timeLimitMs=20000 (20 sec) is reached.	Partial observability for "ObservationFromGrid", the adjacent cell where the agent is located and partial/total for the 3D view ("VideoProducer", screenshot view). Background is a grey room, surrounding the lava. There are no torches around but the lava produces some light. Currently using VideoProducer (640x480)	Possible actions given by "ContinuousMovementCommand" included in "survival mode". So actions are "move", "strafe", "pitch", "turn", "jump", "crouch", "use" and "hotbar.X". Action "attack" is forbidden with a "deny-list" <a href="#">[issue #73]</a> . Inventory empty initially. Only moving on the flat surface required. Jumping might be required if many black lava wholes are created.	Finite no. of states: 12x3 if we consider position only (ignoring where the agent is looking at. Jumping, but the inventory is empty).	Ergodic. Except for the lava and the destination block, any other action is reversible (moving back), so the agent can reach any of the previous states.  If attack is disabled, no block can be broken, so it's ergodic.	No noise (deterministic)	Single agent.	Very narrow task. Specialised agents can be coded or learnt easily.	The difficulty relies on sensorimotor skills. If more lava blocks are added to the arena, the instance difficulty might depend on the number of these lava blocks, their location (closer to the starting block, configuring barriers, etc.). The kind of observations allowed changes difficulty completely. If looking at adjacent is allowed, the task becomes trivial.	Interesting for tabular learners, vision systems, etc. Based on Sutton and Barto. Successfully attempted with G-learning (using many learning episodes, see tutorial_6.py in Python_Examples). Things change significantly depending on the possible observations and actions.	Created with a "FlatWorldGenerator" and a "DrawingDecorator"	2016-05-20	<a href="https://github.com/Microsoft/sample_missions/tree/master/cliff_walking_1.xml">https://github.com/Microsoft/sample_missions/tree/master/cliff_walking_1.xml</a>
maze_1	Maze navigation (using "MazeDecorator")	Zillions of instances	Get a-mazed! The agent appears in a maze of size 20x20. Mazes are within walls.	The goal must be learnt from rewards. The agent appears at starting position (emerald_block) and has to traverse the maze to reach the target position (redstone_block).	Sparse rewards. Negative if dead (-10000), positive if the target block is touched (+1000) and negative if the time is over (-1000).	Task ends if the timeLimitMs=30000 (30 sec) is reached or killed.	Partial observability. Using VideoProducer (640x480)	Possible actions given by "ContinuousMovementCommand" included in "survival mode". So actions are "move", "strafe", "pitch", "turn", "jump", "crouch", "use" and "hotbar.X". Action "attack" is forbidden with a "deny-list" <a href="#">[issue #73]</a> . Inventory empty initially. Only moving on the flat surface required (jumping is useless).	Finite no. of states	Ergodic	No noise (deterministic)	Single agent	Narrow. A simple maze traversal algorithm can solve it.	The difficulty relies on sensorimotor skills, but movements are very constrained. So it depends on how complex the maze is generated (some are straight). Mazes are will walls, so one cannot fall and there are no forks. One cannot die either only from starvation but times are too short for that)	Interesting for maze solvers, planning, general, and humans. No experiments performed yet. The "depth_map_runner.py" example uses a very similar mission.	Created with a "MazeDecorator".	2016-06-12	<a href="https://github.com/Microsoft/sample_missions/tree/master/mazes/maze_1.xml">https://github.com/Microsoft/sample_missions/tree/master/mazes/maze_1.xml</a>
maze_2	Maze navigation (using "MazeDecorator")	Zillions of instances	Get more a-mazed! The agent appears in a maze of size 64x64. Mazes are different as they can be labyrinths with walls, or walking on convoluted cliffs surrounded by lava or water.	The goal must be learnt from rewards. The agent appears at starting position (emerald_block) and has to traverse the maze to reach the target position (redstone_block).	Sparse rewards. Negative if dead (-10000), positive if the target block is touched (+1000) and negative if the time is over (-1000).	Task ends if the timeLimitMs=100000 (100 sec) is reached or killed.	Partial observability. Using VideoProducer (640x480)	Possible actions given by "ContinuousMovementCommand" included in "survival mode". So actions are "move", "strafe", "pitch", "turn", "jump", "crouch", "use" and "hotbar.X". Action "attack" is forbidden with a "deny-list" <a href="#">[issue #73]</a> . Inventory empty initially. Only moving on the flat surface required (jumping may be useful in some mazes). Also enabled: <absoluteMovementCommands /> <discreteMovementCommands />	Finite no. of states	Ergodic	No noise (deterministic)	Single agent	Broad for mazes. Many kinds of mazes, falling is challenging for the ones with lava.	The difficulty relies on sensorimotor skills, but movements are very constrained. So it depends on how complex the maze is generated (some are straight). When mazes are labyrinths with walls, observations are poorer but one cannot fall to lava. When mazes are more like convoluted cliffs, falling is the main concern.	Interesting for maze solvers, planning, general, and humans. No experiments performed yet. The "manual_input_test.py" example uses a very similar mission.	Created with a "MazeDecorator".	2016-06-12	<a href="https://github.com/Microsoft/sample_missions/tree/master/mazes/maze_2.xml">https://github.com/Microsoft/sample_missions/tree/master/mazes/maze_2.xml</a>

basic	Navigation (using "ClassroomDecorator")	A few instances. Even if the original position seems to be fixed (Placement says "x=204", "y=61", "z=217") the ClassroomDecorator moves it slightly from X=1 to 5, so making different instances.	Grab the treasure! A room of 7x7x7 surrounded by stone blocks. There is a target block placed somewhere in the room.	The goal must be learnt from rewards. The task instance is considered complete if the agent goes from starting position to the special block (any of "gold_block diamond_block redstone_block").	Sparse rewards. Negative if dead (-10000), although it is very difficult to get killed. Positive if the target block is touched (+1000) and negative if the time is over (-1000). There's a small reward (+20) for touching the target block.	Task ends if target block is reached. Or if the timeLimitMs=20000 (30 sec) is reached.	Partial observability for ("ObservationFromGrid", the adjacent cell where the agent is located) and partial/total for the 3D view ("VideoProducer", screenshot view). Background is a grey room, surrounding the lava. There are no torches around but the lava produces some light. Currently using VideoProducer (320x240)	Possible actions given by "ContinuousMovementCommand" included in "survival mode". So actions are "move", "strafe", "pitch", "turn", "jump", "crouch", "use" and "hotbar.X". Action "attack" is forbidden with a "deny-list" (Issue #70). Inventory empty initially. Only moving on the flat surface required (Jumping is useless).	Finite no. of states: 7x7 if we consider position only (ignoring where the agent is looking at; what it is holding, jumping, etc).	Ergodic. Except when reaching the target block, any other action is revertible (moving back), so the agent can reach any of the previous states ("attack" is disabled, Issue #70).	No noise (deterministic). A closed environment with monsters disabled (no spawning).	Single agent.	Very narrow task. Specialized agents can be coded or learnt easily.	The task is trivial even for a random agent.	The use of this task is just for checking whether an agent works. However, it might be useful for the 3D view observation, as very simple vision systems might be able to detect the golden block and direct the agent towards it.	Created with a "FlatWorldGenerator" and a "ClassroomDecorator"	2016-05-20	<a href="https://github.com/Microsoft/ml/blob/master/sample_missions/classroom/basic.xml">https://github.com/Microsoft/ml/blob/master/sample_missions/classroom/basic.xml</a>
obstacles	Navigation (using "ClassroomDecorator")	A few instances. Rooms can change slightly. Even if the original position seems to be fixed (Placement says "x=204", "y=61", "z=217") the ClassroomDecorator moves it slightly from X=1 to 5, so making different instances.	The apartment! Four rooms in a row of approximately the same size connected by openings or doors (with switches to open or not). All rooms are at the same level. There might also be lava and water that have to be surrounded. There is a target block placed somewhere in the last room. Spawning zombies, witches, etc., disabled.	The goal must be learnt from rewards. The task instance is considered complete if the agent goes from starting position to the special block (any of "gold_block diamond_block redstone_block").	Sparse rewards. Negative if dead (-10000), although it is very difficult to get killed. Positive if the target block is touched (+2000) and negative if the time is over (-1000). There's a small reward (+20) for touching the target block.	Task ends if target block is reached. Or if the timeLimitMs=45000 (45 sec) is reached.	Partial observability. Currently using VideoProducer (320x240)	Possible actions given by "ContinuousMovementCommand" included in "survival mode". So actions are "move", "strafe", "pitch", "turn", "jump", "crouch", "use" and "hotbar.X". Action "attack" is forbidden with a "deny-list" (Issue #70). Inventory empty initially. Only moving on the flat surface required and the "use" action needed to open the doors. Jumping is not necessary but useful.	Too many states.	Ergodic. Except when reaching the target block or falling into the lava, any other action is revertible (moving back), so the agent can reach any of the previous states ("attack" is disabled, Issue #70).	No noise (deterministic). A closed environment with monsters disabled (no spawning).	Single agent.	Relatively broad task. Variations in colours, kinds of environments, etc. It's even difficult to code an agent manually to succeed in this task.	The task is of some difficulty as there are different ways of opening the doors, there's the lava. It's relatively easy for humans as there is only one path (no forks)	Interesting for navigation robots, deep reinforcement learning agents, people (e.g., with orientation problems). No experiments reported yet.	Created with a "FlatWorldGenerator" and a "ClassroomDecorator"	2016-05-20	<a href="https://github.com/Microsoft/ml/blob/master/sample_missions/classroom/obstacles.xml">https://github.com/Microsoft/ml/blob/master/sample_missions/classroom/obstacles.xml</a>
simpleRoomMaze	Navigation (using "ClassroomDecorator")	Many instances, depending on the generator.	Eight rooms of approximately the same size are connected by openings or doors (with switches to open or not). All rooms are at the same level. There might also be lava and water around that has to be surrounded. There is a target block placed somewhere in the last room. Spawning zombies, witches, etc., disabled.	The goal must be learnt from rewards. The task instance is considered complete if the agent goes from starting position to the special block (any of "gold_block diamond_block redstone_block").	Sparse rewards. Negative if dead (-10000), although it is very difficult to get killed. Positive if the target block is touched (+4000) and negative if the time is over (-1000). There's a small reward (+20) for touching the target block.	Task ends if target block is reached. Or if the timeLimitMs=45000 (45 sec) is reached.	Partial observability. Currently using VideoProducer (320x240)	Possible actions given by "ContinuousMovementCommand" included in "survival mode". So actions are "move", "strafe", "pitch", "turn", "jump", "crouch", "use" and "hotbar.X". Action "attack" is forbidden with a "deny-list" (Issue #70). Inventory empty initially. Only moving on the flat surface required and the "use" action needed to open the doors. Jumping is not necessary but useful.	Too many states.	Ergodic. Except when reaching the target block or falling into the lava, any other action is revertible (moving back), so the agent can reach any of the previous states ("attack" is disabled, Issue #70).	No noise (deterministic). A closed environment with monsters disabled (no spawning).	Single agent.	Relatively broad task. Variations in colours, kinds of environments, etc. It's even difficult to code an agent manually to succeed in this task.	The task is of some difficulty as there are different ways of opening the doors, there's the lava. It's relatively easy for humans as there is only one path (no forks)	Interesting for navigation robots, deep reinforcement learning agents, people (e.g., with orientation problems). No experiments reported yet.	Created with a "FlatWorldGenerator" and a "ClassroomDecorator"	2016-05-20	<a href="https://github.com/Microsoft/ml/blob/master/sample_missions/classroom/simpleRoomMaze.xml">https://github.com/Microsoft/ml/blob/master/sample_missions/classroom/simpleRoomMaze.xml</a>
attic	Navigation (using "ClassroomDecorator")	One instance (agent always appears in X=5, Y=55 and Z=1 even if Placement says "x=204", "y=81", "z=217" because the seed is fixed (214923).	Two rooms (one above the other) of different sizes connected by a stair of blocks (where stairs are not consecutive and one can fall). Rooms are at different levels. There are some small walks with lava around that have to be jumped. There is a target block placed somewhere in the upstairs room. Spawning zombies, witches, etc., disabled.	The goal must be learnt from rewards. The task instance is considered complete if the agent goes from starting position to the special block (any of "gold_block diamond_block redstone_block"). Because of the seed, it is always "gold_block".	Sparse rewards. Negative if dead (-10000), although it is very difficult to get killed. Positive if the target block is touched (+1000) and negative if the time is over (-1000). There's a small reward (+20) for touching the target block.	Task ends if target block is reached. Or if the timeLimitMs=30000 (30 sec) is reached.	Partial observability. Currently using VideoProducer (320x240)	Possible actions given by "ContinuousMovementCommand" included in "survival mode". So actions are "move", "strafe", "pitch", "turn", "jump", "crouch", "use" and "hotbar.X". Action "attack" is forbidden with a "deny-list" (Issue #70). Inventory empty initially. Only moving and jumping are necessary.	Too many states.	Ergodic. Except when reaching the target block or falling into the lava, any other action is revertible (moving back), so the agent can reach any of the previous states ("attack" is disabled, Issue #70).	No noise (deterministic). A closed environment with monsters disabled (no spawning).	Single agent.	Narrow task (only one instance). An agent can be hardwired to solve it.	Going upstairs using the "staircase" is hard even for humans. Precise jumping is needed.	Interesting for navigation robots, deep reinforcement learning agents, people (e.g., with orientation problems). No experiments reported yet.	Created with a "FlatWorldGenerator" and a "ClassroomDecorator"	2016-05-20	<a href="https://github.com/Microsoft/ml/blob/master/sample_missions/classroom/attic.xml">https://github.com/Microsoft/ml/blob/master/sample_missions/classroom/attic.xml</a>
vertical	Navigation (using "ClassroomDecorator")	Many instances, depending on the generator.	Three rooms (each above the other) of similar size connected by stairs (with a corner) and a straight vertical ladder. The three rooms are at different levels. There is a target block placed somewhere in the topmost room. Spawning zombies, witches, etc., disabled.	The goal must be learnt from rewards. The task instance is considered complete if the agent goes from starting position to the special block (any of "gold_block diamond_block redstone_block").	Sparse rewards. Negative if dead (-10000), although it is very difficult to get killed. Positive if the target block is touched (+1000) and negative if the time is over (-1000). There's a small reward (+20) for touching the target block.	Task ends if target block is reached. Or if the timeLimitMs=45000 (45 sec) is reached.	Partial observability. Currently using VideoProducer (320x240)	Possible actions given by "ContinuousMovementCommand" included in "survival mode". So actions are "move", "strafe", "pitch", "turn", "jump", "crouch", "use" and "hotbar.X". Action "attack" is forbidden with a "deny-list" (Issue #70). Inventory empty initially. Only moving (including pitching up for climbing ladders) and jumping are necessary.	Too many states.	Ergodic. Except when reaching the target block or falling into the lava, any other action is revertible (moving back), so the agent can reach any of the previous states ("attack" is disabled, Issue #70).	No noise (deterministic). A closed environment with monsters disabled (no spawning).	Single agent.	Relatively broad task. Variations in colours, kinds of environments, etc. It's even difficult to code an agent manually to succeed in this task.	Going upstairs might be hard even for humans. Precise jumping is needed.	Interesting for navigation robots, deep reinforcement learning agents, people (e.g., with orientation problems). No experiments reported yet.	Created with a "FlatWorldGenerator" and a "ClassroomDecorator"	2016-05-20	<a href="https://github.com/Microsoft/ml/blob/master/sample_missions/classroom/vertical.xml">https://github.com/Microsoft/ml/blob/master/sample_missions/classroom/vertical.xml</a>
complexity_usage	Navigation (using "ClassroomDecorator")	Many instances, depending on the generator.	Several rooms of different sizes connected by doors (with switches to open or not) and stairs or ladders. Rooms are at different levels. There might also be lava and water around and some small walks that have to be jumped. There is a target block placed somewhere in one room. Spawning zombies, witches, etc., disabled.	The goal must be learnt from rewards. The task instance is considered complete if the agent goes from starting position to the special block (any of "gold_block diamond_block redstone_block").	Sparse rewards. Negative if dead (-10000), although it is very difficult to get killed. Positive if the target block is touched (+1000) and negative if the time is over (-1000). There's a small reward (+20) for touching the target block.	Task ends if target block is reached. Or if the timeLimitMs=30000 (30 sec) is reached.	Partial observability. Currently using VideoProducer (320x240)	Possible actions given by "ContinuousMovementCommand" included in "survival mode". So actions are "move", "strafe", "pitch", "turn", "jump", "crouch", "use" and "hotbar.X". Action "attack" is forbidden with a "deny-list" (Issue #70). Inventory empty initially. Only moving (including pitching up for climbing ladders) and jumping are necessary.	Too many states.	Ergodic. Except when reaching the target block or falling into the lava, any other action is revertible (moving back), so the agent can reach any of the previous states ("attack" is disabled, Issue #70).	No noise (deterministic). A closed environment with monsters disabled (no spawning).	Single agent. No help/demo from other agents.	Relatively broad task. Variations in colours, kinds of environments, etc. It's even difficult to code an agent manually to succeed in this task.	Orientation is key and what has been visited. There are a few forks.	Interesting for navigation robots, deep reinforcement learning agents, people (e.g., with orientation problems). No experiments reported yet.	Created with a "FlatWorldGenerator" and a "ClassroomDecorator"	2016-05-20	<a href="https://github.com/Microsoft/ml/blob/master/sample_missions/classroom/complexity_usage.xml">https://github.com/Microsoft/ml/blob/master/sample_missions/classroom/complexity_usage.xml</a>
medium	Navigation (using "ClassroomDecorator")	Many instances, depending on the generator.	14 rooms of different sizes connected by doors (with switches to open or not) and stairs or ladders. There might also be lava and water around and some small rooms are also accessible by jumping from blocks to blocks. There is a target block placed somewhere in one room. Spawning zombies, witches, etc., disabled.	The goal must be learnt from rewards. The task instance is considered complete if the agent goes from starting position to the special block (any of "gold_block diamond_block redstone_block").	Sparse rewards. Negative if dead (-10000), although it is very difficult to get killed. Positive if the target block is touched (+16000) and negative if the time is over (-1000). There's a small reward (+20) for touching the target block.	Task ends if target block is reached. Or if the timeLimitMs=90000 (90 sec) is reached.	Partial observability. Currently using VideoProducer (320x240)	Possible actions given by "ContinuousMovementCommand" included in "survival mode". So actions are "move", "strafe", "pitch", "turn", "jump", "crouch", "use" and "hotbar.X". Action "attack" is forbidden with a "deny-list" (Issue #70). Inventory empty initially. Only moving (including pitching up for climbing ladders) and jumping are necessary.	Too many states.	Ergodic. Except when reaching the target block or falling into the lava, any other action is revertible (moving back), so the agent can reach any of the previous states ("attack" is disabled, Issue #70).	No noise (deterministic). A closed environment with monsters disabled (no spawning).	Single agent. No help/demo from other agents.	Relatively broad task. Variations in colours, kinds of environments, etc. It's even difficult to code an agent manually to succeed in this task.	Orientation is key and what has been visited. There are a few forks.	Interesting for navigation robots, deep reinforcement learning agents, people (e.g., with orientation problems). No experiments reported yet.	Created with a "FlatWorldGenerator" and a "ClassroomDecorator"	2016-05-20	<a href="https://github.com/Microsoft/ml/blob/master/sample_missions/classroom/medium.xml">https://github.com/Microsoft/ml/blob/master/sample_missions/classroom/medium.xml</a>
hard	Navigation (using "ClassroomDecorator")	Many instances, depending on the generator.	32 rooms of different sizes connected by doors (with switches to open or not) and stairs or ladders. There might also be lava and water around and some small walks that have to be jumped, also some rooms are also accessible by jumping from blocks to blocks. There is a target block placed somewhere in one room. Spawning zombies, witches, etc., disabled.	The goal must be learnt from rewards. The task instance is considered complete if the agent goes from starting position to the special block (any of "gold_block diamond_block redstone_block").	Sparse rewards. Negative if dead (-10000), although it is very difficult to get killed. Positive if the target block is touched (+32000) and negative if the time is over (-1000). There's a small reward (+20) for touching the target block.	Task ends if target block is reached. Or if the timeLimitMs=120000 (120 sec) is reached.	Partial observability. Currently using VideoProducer (320x240)	Possible actions given by "ContinuousMovementCommand" included in "survival mode". So actions are "move", "strafe", "pitch", "turn", "jump", "crouch", "use" and "hotbar.X". Action "attack" is forbidden with a "deny-list" (Issue #70). Inventory empty initially. Only moving (including pitching up for climbing ladders) and jumping are necessary.	Too many states.	Ergodic. Except when reaching the target block or falling into the lava, any other action is revertible (moving back), so the agent can reach any of the previous states ("attack" is disabled, Issue #70).	No noise (deterministic). A closed environment with monsters disabled (no spawning).	Single agent. No help/demo from other agents.	Relatively broad task. Variations in colours, kinds of environments, etc. It's even difficult to code an agent manually to succeed in this task.	Orientation is key and what has been visited. There are a few forks.	Interesting for navigation robots, deep reinforcement learning agents, people (e.g., with orientation problems). No experiments reported yet.	Created with a "FlatWorldGenerator" and a "ClassroomDecorator"	2016-05-20	<a href="https://github.com/Microsoft/ml/blob/master/sample_missions/classroom/hard.xml">https://github.com/Microsoft/ml/blob/master/sample_missions/classroom/hard.xml</a>