ANU Agile Digital: Autonomous Exploration & Mining Simulation - Competitors' API

GoldDigger documentation

This is an instruction manual on how to play the GoldDigger game.

The goal of the game is to gather as much gold as possible within the given time frame.

This game is running on an HTTP-server. For the examples in this documentation the server resides at http://localhost:8066/golddigger. The correct url (e.g. IP address of the server) was provided to you when the game started.

The part of the path that says veryVerySecret must be replaced by the password given to you by the game administrator.

Your player can be controlled by sending specific actions or state requests to the web server. Actions makes your player interact with the environment, and state requests asks for the current state of you player.

The server always return HTTP reponse code 400, unless it utterly fails. That is, the responses below always represents the content of the response, and never any HTTP status.

A game

View your nearest surroundings

Line of sight length and tile number of sides is determined by 2 factors:

- Tile number of sides
- Line of sight length

Both of these are recorded in the .field files. If not, they will default to:

- Tile number of sides: 4 (i.e. square)
- Line of sight length: 1

Example 1 - square tiles:

Line of sight length = 2;

Tile = Square tiles;

To view your nearest surroundings, issue the following command to the server:

```
http://localhost:8066/golddigger/digger/veryVerySecret/view
```

The response text looks like this:

```
-----
-wwww
-wb..
-ww2.
-w.w.
```

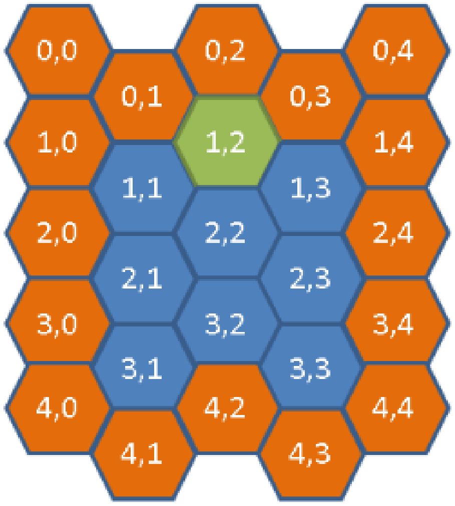
The digger will always be centered, and the line of sight would count from where your digger is, in this case, the digger would be sitting on top of the base (b), since line of sight is 2, and the digger is centered at (1,1) you would be able to see 2 extra moves to east and south.

The w represents walls. Walls limits the area in which the player can move and they cannot be treaded. The outside of the field is
surrounded by dashes(**-**), These dashes are just used as a wrapper to always have the digger clearly centered,

Numbers represents squares where there is GOLD! More about that later.

Example 2 - hexagon

If a map is in 6-sided mode, the view will change to suit based on the following co-ordinate system:



for an easy comparison between these two tile sets consider the following map:

```
wwwww

w.b.w

w...w

w...w

wwwww
```

with the following attributes:

Line of sight length = 2;

Digger Position = 2,2

Tile = Hexagon tiles;

To view your nearest surroundings, issue the following command to the server just as you would for square tiles:

```
http://localhost:8066/golddigger/digger/veryVerySecret/view
```

The response text looks like this:

```
?www?
w.b.w
w...w
w...w
??w??
```

The question marks in the above and following examples represents the tiles that would be seen in four-sided tiles, but because of how the

adjacencies work in six-sided tiles, they are not visible.

Human Hex View

In maps with hexagon tiles, teams can choose to see a view that is easier to read by a human.

```
http://localhost:8066/golddigger/digger/veryVerySecret/view/human
```

```
? w ?
 w w
w b w
 . .
w . w
 . .
w . w
 . .
? w ?
 ? ?
```

Occlusion

In the game, tiles will be blocked from your line of sight if they are obstructed by a blocking tile. A tile is considered blocked from view if you are unable to draw a line from the centre of digger's tile to the centre of the tile in question because the radius of another tile is in the way. Any tiles that are deemed blocked are represented in the output as '?' characters. There are a number of blocking tiles such as hills, walls, mountains ,etc. Each of these tiles blocks the view based on a percentage which varies from tile to tile. The default values for these can be found in a table at the end of this section. Walls will have a blocking percentage of 100%and Hill's(the 'h' on the field) have a percentage of 50%. Since all tiles have a uniform area, these percentages show how large the radius a tile has that will block the digger's view.

for example,if we had this field:

```
wwwww
wb..w
w.h.w
w.h.w
wwwww
```

The walls and hills are the blocking tiles that can obstruct the view based on the digger's location. Lets try this with a line of sight that encompasses the entire map i.e. 3. The view command issued by a digger at the (1,1) Position would give the following output:

```
-------
-------
--www??
--wb..w
--w.h.w
--?.h??
--?ww??
```

The Wrapper around the map ("-") are immune to the affects of occlusion as they are not a real tile type.

Default Attributes of Tiles

Tile	Occlusion height	Score cost:
Bank	0	0
City	2	0
Deep Water	0	0
Forest	1	0
Gold	0	0
Hill	2	0
Invisible Teleport	0	0
Mountain	3	0
Road	0	0
Shallow Water	0	0
Visible Teleport	0	3
Wall	Integer Max	0

Make a move

Movement Costs

All tiles are associated with a movement cost that affect how quickly a digger can move onto that tile and then issue commands. Movement commands will initially give an "Ok" if on a successful move but the movement cost of a tile will affect how long (in milliseconds) it takes to traverse onto a tile. While the digger is moving, it will not be able to receive commands.

If a digger was to move onto a tile this would be the immediate output an:

```
OK
```

This represents the digger confirming that it has got the command and is moving onto the tile. While the digger is traversing the tile, it will not be able to receive further commands until finished.

Commands that are sent while the digger is moving will result in:

```
HTTP ERROR: 503

SERVICE_UNAVAILABLE

RequestURI=/golddigger/digger/veryVerySecret/view

Powered by Jetty://
```

This represents the time it takes for the digger to move onto a tile. (This is also the response when too many commands are issued to the digger at a time)

Here are the default movement costs for the tiles:

Tile	**Movement Cost(in milliseconds)**
Bank	100
City	200
Deep Water	500
Forest	300
Gold	100
Hill	175
Invisible Teleport	100
Mountain	300
Road	25
Shallow Water	150
Visible Teleport	100
Wall	100

Square tiles - 4 sided tiles

The player can move in four directions:

direction	url	response
south	http://localhost:8066/golddigger/digger/veryVerySecret/move/south	OK

With the new view:

```
-------
--wwwww
--wb..w
--w....
--w....
--w....
--w.1w.
```

direction	url	response
east	http://localhost:8066/golddigger/digger/veryVerySecret/move/east	OK

With the new view:

```
-------
-wwwwww
-wb..w.
-w....3
-w.....
-w.....
-w.1w.w
```

direction	url	response
north	http://localhost:8066/golddigger/digger/veryVerySecret/move/north	OK

with the new view:

```
-------
-------
-wwwwww
-wb..w.
-w....3
-w.....
-w.....
```

direction	url	response
west	http://localhost:8066/golddigger/digger/veryVerySecret/move/west	OK

with the new view:

```
-------
-------
--wwwww
--wb..w
--w....
--w....
--w....
```

Hex tiles - 6 sided tiles

The player can move in six directions:

direction	url	response
south	http://localhost:8066/golddigger/digger/veryVerySecret/move/south	OK

With the new view:

```
-------
--wwwww
--wb..w
--w....
--w....
--w...?
--?.???
```

direction	url	response
north	http://localhost:8066/golddigger/digger/veryVerySecret/move/north	OK

With the new view:

```
-------
-------
--wwwww
--wb..w
--w....
--w...?
--?.???
```

direction	url	response
south-east	http://localhost:8066/golddigger/digger/veryVerySecret/move/south_east	OK

With the new view:

```
-------
-wwwww?
-wb..w.
-w....3
-w.....
-w.....
-?.1w??
```

direction	url	response

south-west	http://localhost:8066/golddigger/digger/veryVerySecret/move/south_west	OK

With the new view:

```
-------
--wwwww
--wb..w
--w....
--w....
--w...?
--?.???
```

direction	url	response
north-east	http://localhost:8066/golddigger/digger/veryVerySecret/move/north_east	OK

With the new view:

```
-------
-wwwww?
-wb..w.
-w....3
-w.....
-w.....
-?.1w??
```

direction	url	response
north-west	http://localhost:8066/golddigger/digger/veryVerySecret/move/north_west	OK

With the new view:

```
-------
--wwwww
--wb..w
--w....
--w....
--w...?
--?.???
```

...Since some movements won't work depending on how many sides a tile set has, the following text response is outputted:

4 sided tile set examples:

direction	url	response
north-west	http://localhost:8066/golddigger/digger/veryVerySecret/move/north_west	FAILED

direction	url	response
south-east	http://localhost:8066/golddigger/digger/veryVerySecret/move/south_east	FAILED

The same response is also given for north-east and south-west.

6 sided tile set examples:

direction	url	response
west	http://localhost:8066/golddigger/digger/veryVerySecret/move/west	FAILED

direction	url	response
east	http://localhost:8066/golddigger/digger/veryVerySecret/move/east	FAILED

Grab the gold

To grab the gold on the spot you stand, issue the grab command. This is the view before grab where we have centered the player above a square with a number indicating gold(5 in this example):

```
-------
wwwww--
w...w--
.3w5w--
....w--
...ww--
.w..w--
```

... this is the grab command:

direction	url	response
grab	http://localhost:8066/golddigger/digger/veryVerySecret/move/grab	3

The response indicates that three piece of gold was grabbed. The view after looks like this:

```
-------
wwwww--
w...w--
.3w2w--
....w--
...ww--
.w..w--
```

The GoldDigger is greedy and will always try to grab as much as possible, but he cannot carry more than 3 pieces of gold at a single time. Here he is on a square with nine pieces, carrying one piece already:

```
-------
wwwww--
w...w--
.3w2w--
....w--
...ww--
.w..w--
```

When grabbing the gold on that spot, the response is

action	response
grab	0

...and the view should stay the same:

```
-------
wwwww--
w...w--
.3w2w--
....w--
...ww--
.w..w--
```

Check how much you are carrying

Now that you have started picking up gold, you might w ant to see how much you are carrying.

action	url	response
carrying	http://localhost:8066/golddigger/digger/veryVerySecret/carrying	3

Drop carried gold

You cannot carry more than 3 pieces of gold at a time. When you have found and grabbed three pieces, you must drop the gold
somew here. This is the view before:

```
w..
w.7
www
```

action	url	response
drop	http://localhost:8066/golddigger/digger/veryVerySecret/drop	3

The response will tell you how many pieces you dropped. This is the view after:

```
w..
w37
www
```

Maximum number of pieces on one square

A square can hold nine pieces of gold, so you can drop up to that amount on one square. If you carry more than the square can absorb, you keep the difference

```
w..
w37
www
```

action	response
grab	3
carrying	3
move east	ok

Then lets drop the gold there:

action	response
drop	2

The field looks like this:

```
..w
.9w
www
```

...and you are still carrying

action	response
carrying	1

Cash the gold in the bank

To cash the gold, you need to go to a square with a b on it and drop what you are carrying there. The gold will then be credited to your account.

```
www
wb.
w..
```

action	url	response

drop	http://localhost:8066/golddigger/digger/veryVerySecret/drop	3

Check how much you have in the bank

When you have cashed your gold, you can check your balance score like this

action	url	response
score	http://localhost:8066/golddigger/digger/veryVerySecret/score	3

Move to the next field

When you have cashed all the gold in one field you are allowed to move to the next map. If you try to move to the next field before you have picked all gold pieces in the current field you will get a message like this

action	url	response
next	http://localhost:8066/golddigger/digger/veryVerySecret/next	FAIL

...and you will remain in the current field until you have picked them all.
After having picked all gold pieces you will get this response:

action	url	response
next	http://localhost:8066/golddigger/digger/veryVerySecret/next	OK

...and you will be moved to the next field.

Teleportation

Advantageous Teleportation

Maps may contain teleportation tiles. These tiles are paired together and require extra commands and gold to use. One Advantageous teleport tile may be picked up(at a cost) by a digger at a time and may be dropped similarly to gold back onto the map. Activating the teleport tile will send the digger to the location of the paired teleport tile. This teleportation is two-way so both ends of the advantageous teleport pair can be activated to reach the other end.

Advantageous teleports appear as a 't' on the map:

```
wwwww
wbt.w
w.h.w
w.htw
wwwww
```

Picking up a teleport tile

Teleports cost 3 gold score to pick up i.e. You must have banked 3 gold.

carrying Teleport	no
score	3

action	url	response
grab	http://localhost:8066/golddigger/digger/veryVerySecret/teleport/grab	OK: Teleport grabbed

carrying Teleport	yes
score	0

These teleport tiles sit on top of regular traversable tiles. So when you pick them up and do a view command, what they were on top of is revealed:

```
www
b..
.h.
```

when one end of an advantageous teleport pair is picked up, the other teleport tile roots itself to the ground and cannot be grabbed by any digger until the paired teleport tile is returned to the map in the instance that another digger/player has grabbed one end of the advantageous teleport pair, and you attempt to grab the other end, the response from the server will be:

action	url	response
grab	http://localhost:8066/golddigger/digger/veryVerySecret/teleport/grab	FAILED: Destination teleport tile being held

If the digger is already carrying a teleport the grab command will result in:

action	url	response
grab	http://localhost:8066/golddigger/digger/veryVerySecret/teleport/grab	FAILED: Already carrying a teleport tile

If the digger is not ontop of a teleport:

action	url	response
grab	http://localhost:8066/golddigger/digger/veryVerySecret/teleport/grab	FAILED: Unit not on teleport tile

If there are not enough funds in the bank(score)

action	url	response
grab	http://localhost:8066/golddigger/digger/veryVerySecret/teleport/grab	FAILED: You can not afford to move this teleport

Dropping a teleport tile

dropping a teleport tile will replace the tile the digger is currently occupying with the teleport tile:

action	url	response
drop	http://localhost:8066/golddigger/digger/veryVerySecret/teleport/drop	OK: Teleport dropped

view:

```
www
bt.
.h.
```

A digger cannot drop another teleport tile onto a teleport tile, so in the following situation:

view:

```
www
bt.
.h.
```

carrying Teleport	yes

A drop would result in:

action	url	response
drop	http://localhost:8066/golddigger/digger/veryVerySecret/teleport/drop	FAILED: Can't drop a teleport tile on a teleport tile

Diggers also cannot drop teleports onto tiles with gold on them in this circumstance the response would be:

action	url	response
drop	http://localhost:8066/golddigger/digger/veryVerySecret/teleport/drop	FAILED: Can't drop a teleport tile on a tile that currently holds gold

in the event that the digger is not holding a teleport tile:

action	url	response
drop	http://localhost:8066/golddigger/digger/veryVerySecret/teleport/drop	FAILED: not carrying a teleport tile

Activating a teleport tile

to activate an advantageous teleport tile:

action	url	response

activate	http://localhost:8066/golddigger/digger/veryVerySecret/teleport/activate	OK

this will send the digger to the location of the paired teleport tile so a view will display a new location:

```
h.w
.tw
www
```

activating again will send the digger back:

action	url	response
activate	http://localhost:8066/golddigger/digger/veryVerySecret/teleport/activate	OK

view:

```
www
bt.
.h.
```

there are a few ways that the activation command can fail:

1. Another digger is occupying the other end of the advantageous teleport pair
2. The other end of the teleport pair is being held by another digger.
3. The digger is not on a teleport tile

in these instances an activation command will result in following respectively:

action	url	response
activate	http://localhost:8066/golddigger/digger/veryVerySecret/teleport/activate	FAILED: Destination currrently blocked

action	url	response
activate	http://localhost:8066/golddigger/digger/veryVerySecret/teleport/activate	FAILED: Destination teleport tile being held

action	url	response
grab	http://localhost:8066/golddigger/digger/veryVerySecret/teleport/grab	FAILED: Unit not on teleport tile

Disadvantageous Teleportation

There may be tiles within the map that can automatically teleport the digger once the digger moves onto those tiles. These teleport tiles are invisible to the digger's view and both look and have all the attributes and effects of any other traverse-able tile(city, deep water, ...). So essentially, any tile on the map could have these hidden teleportation tiles beneath it and once a digger moves onto these tiles they will be teleported to the tile's intended destination.

for example,if we had this field with the digger currently positioned on the base and a disadvantageous tile directly to the east of the digger.:

```
wwwww
wb..w
w.h.w
w.h.w
wwwww
```

a view(line_of_sight = 1) command would result in:

```
www
wb.
w.h
```

when the digger moves onto the disadvantageous teleport the response will reveal how far the digger has moved from the position of the teleport tile.

direction	url	response
east	http://localhost:8066/golddigger/digger/veryVerySecret/move/east	OK: Teleported to (2,1)

that will be the response and the digger will now be located(in this instance) 2 tiles south and 1 east. So another view for this particular teleport would result in:

```
h.w
h.w
www
```

Disadvantageous teleports are one-way so moving back onto the tile you were teleported to, won't send you back to the location of the disadvantageous teleport tile.

Multiplayer

Joining a game:

Multiplater maps are divided into 4 states, **Starting, Running**, **Ending** and **Finished**.

Starting State:

In this state, all commands will receive the following output:

```
FAILED: Joining Only
```

however, the **view** command will also output the current state of the map and how much time(in milliseconds) is left in that state. So for example: starting state and 1 minute left:

command	url	response

view	http://localhost:8066/golddigger/digger/veryVerySecret1/view	state: STARTING,1000

Running State:

In a multiplayer game, all the commands, actions, and the basic functionality of everything will remain basically the same. The only difference is that you must contend with other digger's vying for the gold on the map, there are multiple bases and you have a strict time limit until the map ends. (there is also combat, but that'll be shown in the next section).

Dropping gold on another player's base doesn't work. The base you start the game on is your base for the duration of the multiplayer map and is the only one capable of being dropped on. The view command, will print the view regularly but like in the previous section, will also print the current state and time left for a very condensed example of multiplayer in the running state consider this map:

```
wwwww
wb..w
w.9.w
w.9bw
wwwww
```

digger1 view:

```
www
wb.
w.9

state: RUNNING,10000
```

digger2 view:

```
9.w
9bw
www

state: RUNNING, 9999
```

other diggers are represented by the view command to be a 'd' on the map. To illustrate this, say digger2 moved north twice and west once while digger1 moved south once and east once. If both diggers did a view command, it would look like:

digger1 view:

```
bd.
.9.
.9.

state: RUNNING, 9990
```

digger2view:

```
www
b..
.d.

state: RUNNING,9989
```

if either digger wanted to move to the other digger's position:

digger1:

direction	url	response
north	http://localhost:8066/golddigger/digger/veryVerySecret1/move/north	FAILED

digger2:

direction	url	response
south	http://localhost:8066/golddigger/digger/veryVerySecret2/move/south	FAILED

now, if digger 2 moved west onto digger 1's base:

digger2view:

```
www
wb.
w.9

state: RUNNING,9987
```

an attempted to drop gold(assuming that digger2 is carrying gold) the output would be:

action	url	response
drop	http://localhost:8066/golddigger/digger/veryVerySecret/drop	FAILED

Now, even if drop fails, there is a neat thing you can do while occupying an opponent's base. issuing a **score** command will give you how much has been banked and the name of the player who owns that base respectively:

command	url	response
score	http://localhost:8066/golddigger/digger/veryVerySecret1/score	0 player1

Ending State

when the running time is over or when the last piece of gold has been picked up, the map will enter this state which instantly wipes all gold from the map(if any). This state allows the diggers a limited amount of time to bank any gold it is carrying before the map officially ends and the gold that is being carried is lost. All commands should work as intended and the **view** command will again give the current state and time left:

command	url	response
view	http://localhost:8066/golddigger/digger/veryVerySecret1/view	state: ENDING, 100

Finished State

When a map enters the Finished state, all gold from the map and gold being carried by any diggers left are removed. In this state all commands will return:

FAILED: Game Over

except of course for the **view** command:

command	url	response
view	http://localhost:8066/golddigger/digger/veryVerySecret1/view	state: FINISHED,0

Combat

In multiplayer maps you gain the ability to buy ammo for your digger and shoot other diggers if they are within range(2). After getting shot, the enemy digger will drop everything it is carrying and will respawn at it's base.

Buying a cannon

ammo costs a certain amount of gold, in this example they cost 1 gold to buy. This cost is subtracted from your score, so you must have money in the bank to buy ammo:

```
-----
-wwww
-wb..
-w.9.
-w.9b
```

given the following:

ammo	0
carrying	0
score	2

a "buy" command will result in:

action	url	response
buy	http://localhost:8066/golddigger/digger/veryVerySecret/cannon/buy	OK you have 1 rounds left

ammo	1
carrying	0
score	1

another "buy" command at this point(with zero score) would result in:

action	url	response
buy	http://localhost:8066/golddigger/digger/veryVerySecret/cannon/buy	FAILED: Dont have enough cash

if the digger was not on it's base the following would be the response:

action	url	response
buy	http://localhost:8066/golddigger/digger/veryVerySecret/cannon/buy	FAILED: Not on your bank

if another digger was foolish enough to come after you:

```
-----
-wwww
-wb.d
-w.9.
-w.9b
```

other digger's attributes:

ammo	0
carrying	3

Just issue a "shoot" command along with the latitude and longitude of the digger and show it who's boss:

action	url	response
shoot	http://localhost:8066/golddigger/digger/veryVerySecret/camnon/shoot/one/three	FAILED: invalid target

woops looks like the either the longitude or latitude wasn't an integer, luckily an invalid target doesn't expend ammo, attempting to shoot out of range will also not expend ammo:

action	url	response
shoot	http://localhost:8066/golddigger/digger/veryVerySecret/cannon/shoot/1/4	FAILED: out of range

alright, let's try shooting at these co-ordinates:

action	url	response

shoot	http://localhost:8066/golddigger/digger/veryVerySecret/cannon/shoot/1/2	MISSED

oh no! we fired at the the wrong location, FIRE AGAIN!

action	url	response
shoot	http://localhost:8066/golddigger/digger/veryVerySecret/cannon/shoot/1/3	FAILED: out of ammo

oh right....looks like we'll have to buy some more ammo and try again:

action	url	response
buy	http://localhost:8066/golddigger/digger/veryVerySecret/cannon/buy	OK you have 1 rounds left

ammo	1
carrying	0
score	0

alright third time's the charm:

action	url	response
shoot	http://localhost:8066/golddigger/digger/veryVerySecret/cannon/shoot/1/3	HIT

alright! that's more like it, let's check if the digger had any gold on it when it died with a view command:

```
-----
-wwww
-wb.3
-w.9.
-w.9d
```

Now it's just a matter of collecting the spoils of war. Don't worry the other digger isn't really dead, look at the previous view, he's just back at his base....well an exact replica of him is back at his base...

Default Attributes of Tiles

Tile	Map Representation	Movement Costs	Occlusion height	Score cost:
Bank	b	100	0	0
City	c	200	2	0

Deep Water	d	500	0	0
Forest	f	300	1	0
Gold	"."(0 gold) or 1-9(1-9 gold)	100	0	0
Hill	h	175	2	0
Invisible Teleport	N/A	100	0	0
Mountain	m	500	3	0
Road	r	25	0	0
Shallow Water	s	150	0	0
Visible Teleport	t	100	0	3
Wall	w	N/A	Integer Max	0
Occluded Tile	?	N/A	N/A	N/A
Map Wrapper	-	N/A	N/A	N/A