CLA Summit 2018 Coding Competition

tronview

# Table of contents

[Table of contents 1](#_Toc501317993)

[Game basics 2](#_Toc501317994)

[Technical details 3](#_Toc501317995)

[Quick start 3](#_Toc501317996)

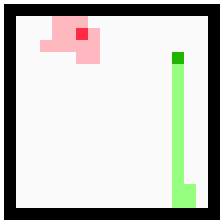
[Bot.vi 3](#_Toc501317997)

[Competition format 4](#_Toc501317998)

# Game basics

tronview is a game where two players move through the board, leaving behind an impenetrable wall. When a player hits one of these walls, they crash and have lost the game. To win, you must outlive your opponent.

* The game is played on a square board which consist of NxN fields. The board is limited by the walls.
  + Board size is decided at the beginning of the game.
  + Minimum board size is 16x16, maximum is 48x48.
* Bots start at random positions on the board.
* The game is played in turns. In each turn each bot makes move (N, E, S or W), leaving the wall on the field it left.
  + Bots move simultaneously.
* The bot loses when:
  + It finish the move in the field which is already occupied by wall.
  + Processing time of the bot takes too much time (see game rules below).
  + Storage limit of the bot is exceeded (see game rules below).
* Game results in a draw when both bots lose in the same turn.
  + Situation when both bots moves into the same field in the same turn is also considered a draw.



Example game board where green bot wins (red bot has crashed into it's own wall)

# Game rules

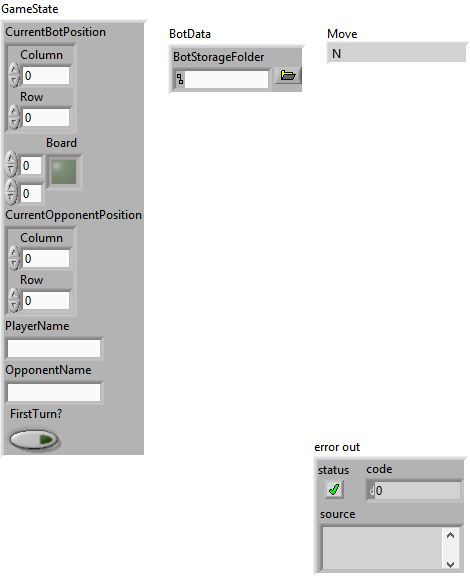
1. Your bot will be called by reference by game engine. Your bot must implement connector pane of the TronVIEWBotAPI.lvlib:Bot.vi to run correctly.
2. Your Bot.vi must take no more than 0.5 second (500 ms) to complete. This limit is needed to ensure that competition will run smoothly.
   1. This limits acts as a fail-safe if you bot stalls for some reason. Normally it should take much less time to complete. Please consider that processing time of your bot may
   2. Your bot must act as simple subVI and finish any processing during allocated time. It may not spawn any continuous processes in background.
3. Your bot may use provided BotStorageFolder to store any data .

Technical details

## Quick start

1. Download the game code.
2. Run the Example\_Game.vi. It will open another VI front panel where the game takes place.
3. Copy the TemplateBot.lvlib and its contents. Rename it to YourName.lvlib (YourName = your name ☺)
4. Implement the Bot.vi
5. In the Example\_Game.vi, replace one of the bots with a path to your bot folder.
6. Have fun!

## Bot.vi



|  |  |  |
| --- | --- | --- |
| INPUTS | | |
| GameState | CurrentBotPosition | Current position of your bot. |
| Board | Current state of the fields on the board.  False = field is not occupied.  True = field is occupied. |
| CurrentOpponentPosition | Current position of opponents bot. |
| PlayerName | Your name. |
| OpponentName | Your opponents name. |
| FirstTurn? | Indicates first turn in current game. |
| BotData | BotStorageFolder | The folder in which the bot may store any persistent data it wants.  This folder will stay the same during entire competition. |
| OUTPUTS | | |
| Move | | The move your bot make in this turn. |
| Error out | | Error returned by your bot. Error will be logged, so you may use it for further investigation during development. |

### Bot processing time

# Competition format

**Competition format is provisional and is subject to change before final release.**

* **The competition will be split into two parts: round-robin round and knock-out round.**
* **In round-robin round each bot will play multiple games with each other.**
  + **Standings will be based on the points gained by bots: winner of each game will gain 2 points, loser 0 points. If the game ends with a draw, each bot will gain 1 point.**
* **Best X bots will advance into knock-out round.**
  + **Bots will be paired and will play multiple games. Winner will advance to next round, and the process will be repeated until the final between two bots.**