CLA Summit 2018 Coding Competition

tronview

# Table of contents

[Table of contents 1](#_Toc503824418)

[Game basics 2](#_Toc503824419)

[Game rules 3](#_Toc503824420)

[Competition format 3](#_Toc503824421)

[Technical details 4](#_Toc503824422)

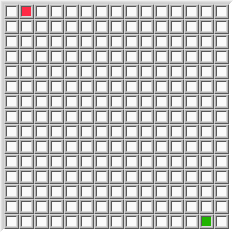
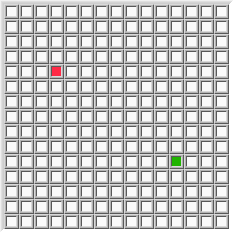
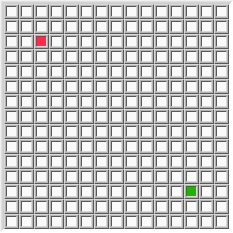
[Quick start 4](#_Toc503824423)

[Bot.vi 4](#_Toc503824424)

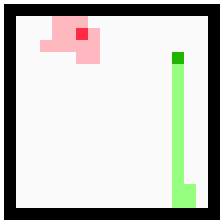
# Game basics

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 randomly at the beginning of the game.
  + Minimum board size is **15x15**, maximum is **33x33**.
  + Board size is always **odd**.
* Bots start at random positions on the board, symmetrically to each other
  + First bot starts at **random position** in the first quarter of the board. Second bot starts at position **symmetrical** to the first with respect to the center of the board.
  + Examples:



* 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 finishes 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. The goal of the coding competition is entertainment – have fun and let others have fun! ☺
2. 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.
3. 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. Treat this limit as a fail-safe if your bot stalls for some reason. Normally your bot should take much less time – please consider that processing your bot may take longer with the competition computer that your super-efficient-38-cores-12-GHz-killer-machine ☺
   2. Your bot must finish all the processing in the time limit. Please do not spawn any continuous processes in the background.
4. Your bot may use provided **BotStorageFolder** to store any data. Maximum size of the stored data must not exceed **20 MB (20 971 520 bytes)**.
   1. This storage is provided to allow you to implement some long-term strategies. Your bot will face the same opponents multiple times, so maybe it’s a good idea to remember something about them…?
   2. You may not store any data outside of provided folder.
   3. Your bot will be provided with **CurrentStorageSize (bytes)**. If current size is larger than **MaxStorageSize (bytes)**, your bot must clean it up in this turn. If the size is still exceeded in the next turn, your bot will **lose current game**.

Competition format

**Competition format is still subject to change (some details might be tweaked – like number of games, etc).**

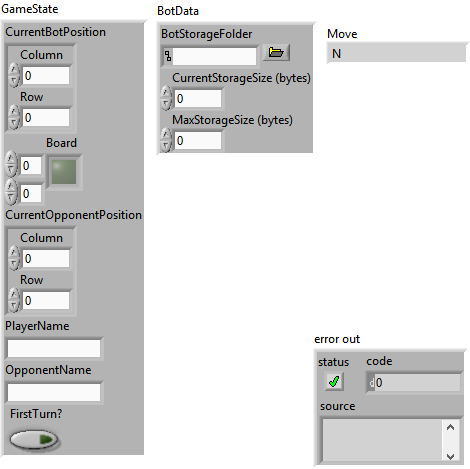
* The competition will be split into two parts: round-robin and knock-out.
* During round-robin:
  + There will be **3 rounds** during which each bot will play **5 games** with each other.
  + Standings will be based on the points gained by bots: winner of each game will gain 2 points, loser 0 points. If the game ends with a draw, each bot will gain 1 point.
* Best 8 bots will advance into knock-out:
  + Bots will be paired and will play **best-of-11 match**:
    - Bots will play 11 games, each game scored as above, and the winner will advance to the next round.
    - In case of a draw after 11 games, bots will play one another game to determine winner (and another if this game will also result in a draw).
  + Process will be repeated until the final between two remaining bots.

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. |
| CurrentStorageSize (bytes) | Current size of your storage folder. |
| MaxStorageSize (bytes) | Maximum size of your storage folder. |
| 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. |