



## **Руководство пользователя**

Программа «Программируемая мультиагентная кибернетическая студия для управления роботами и автономного запуска стратегии (РАстаCS)»



---

## Содержание

|                                  |    |
|----------------------------------|----|
| 1. Общие сведения                | 3  |
| 2. Установка и запуск программы  | 4  |
| 3. Справка по интерфейсу Serviz  | 5  |
| Главное меню                     | 5  |
| Меню телеметрии                  | 7  |
| Графический холст                | 12 |
| 4. Справка по работе через Ether | 13 |
| Draw                             | 15 |
| Telemetry                        | 16 |
| Signals                          | 16 |



## 1. Общие сведения

Программируемая мультиагентная кибернетическая студия для управления роботами и автономного запуска стратегии (РАстаCS) — модульный программный пакет для разработки, тестирования и визуализации стратегий RoboCup Small Size League (SSL). Состав: SerViz (веб-интерфейс визуализации и управления), TransNet (шлюз к инфраструктуре SSL и симуляторам), TelSink (сбор и воспроизведение телеметрии). Взаимодействие модулей осуществляется по внутренней шине сообщений на базе ZeroMQ (каналы Draw, Telemetry, Signals). Основное взаимодействие с программным пакетом происходит через веб-интерфейс, предоставляемый модулем Serviz. При необходимости возможно взаимодействие с программным пакетом через шину данных Эфир (Ether). В настоящем руководстве представлена информация о взаимодействии с программным пакетом посредством перечисленных инструментов.



## 2. УСТАНОВКА И ЗАПУСК ПРОГРАММЫ

1. Перейдите в каталог проекта
2. Установите зависимости и Docker.:  
`make install`
3. Перезапустите устройство
4. Инициализируйте окружение и соберите проект:  
`make init`  
`make build`
5. Запустите сервисы с симулятором grSim:  
`make up-all`  
или без:  
`make up`
6. Откройте веб-интерфейс SerViz: <http://localhost:8000>.

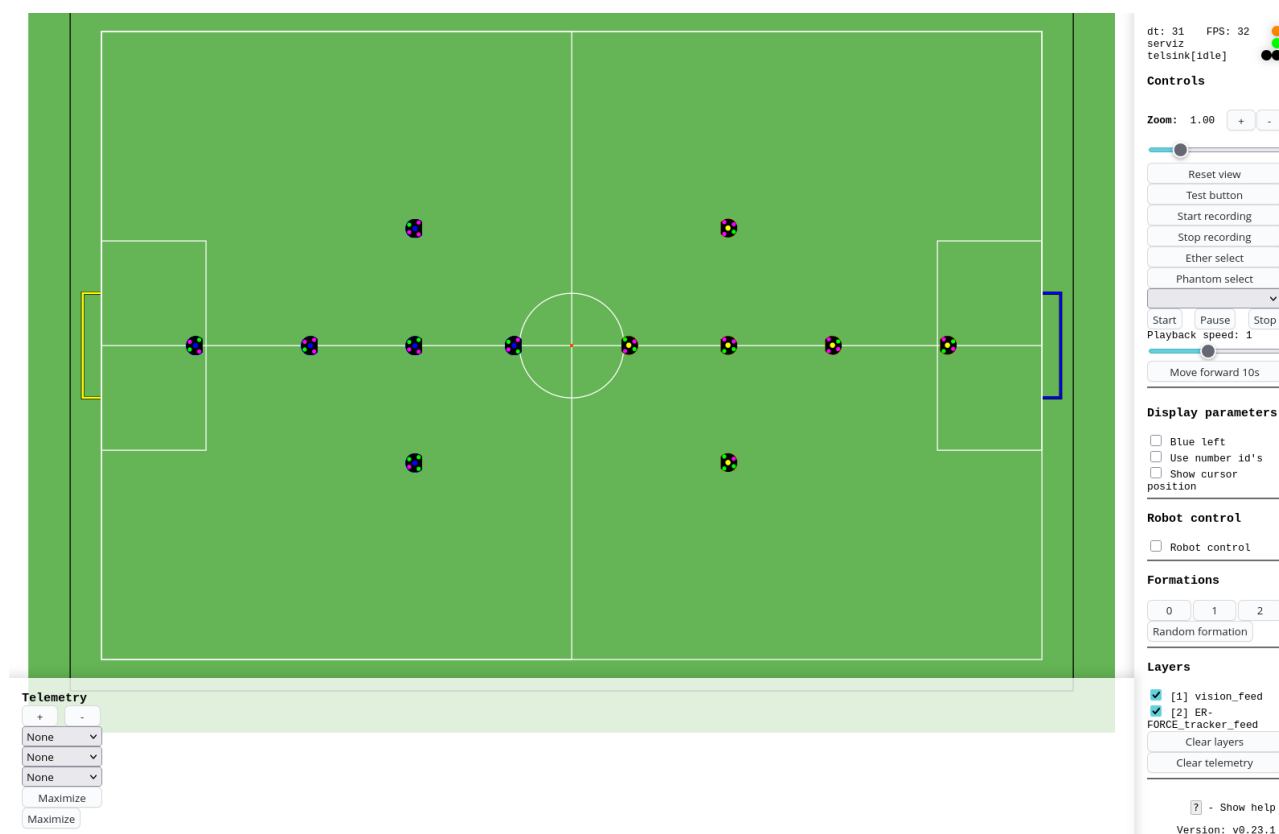
В открытом окне должен появиться интерфейс программы и изображение поля с роботами (в случае запуска симулятора). Если этого не произошло, проверьте правильность установки и, при необходимости, повторите ее.



### 3. СПРАВКА ПО ИНТЕРФЕЙСУ SERVICZ

Далее будет описываться запуск с настройками по умолчанию (дивизион Б, роботы в симуляторе) с помощью команды **make up-all** версии v0.23.1

При корректной установке и запуске по адресу <http://localhost:8000> будет доступен веб-интерфейс модуля Serviz.



Пользовательский интерфейс состоит из трех основных областей: меню управления, окно телеметрии и графический холст для отображения игрового поля и визуальной информации.

#### Главное меню

Внешний вид главного меню представлен на рис. 4.3.

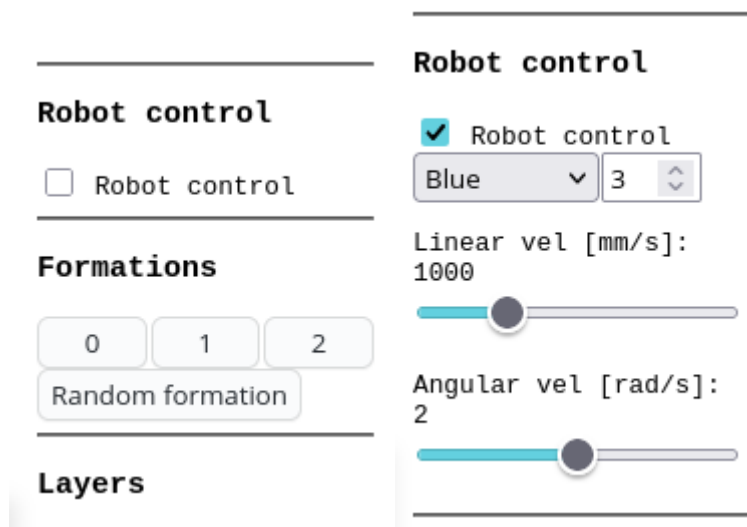
Через меню можно увидеть (сверху-вниз):

- информацию о частоте обновления изображения и данных телеметрии,
- информацию о наличии соединения с бэк-эндом сервиса,
- информацию о статусе работы телсинка
- управление настройками камеры (приближение, отдаление, сброс и др.),
- настраиваемые пользовательские кнопки для отправки сигналов в систему,
- кнопки управления записью и воспроизведением реплеев с помощью телсинка,
- настройки отображения игрового поля,
- настройки ручного управления роботами,



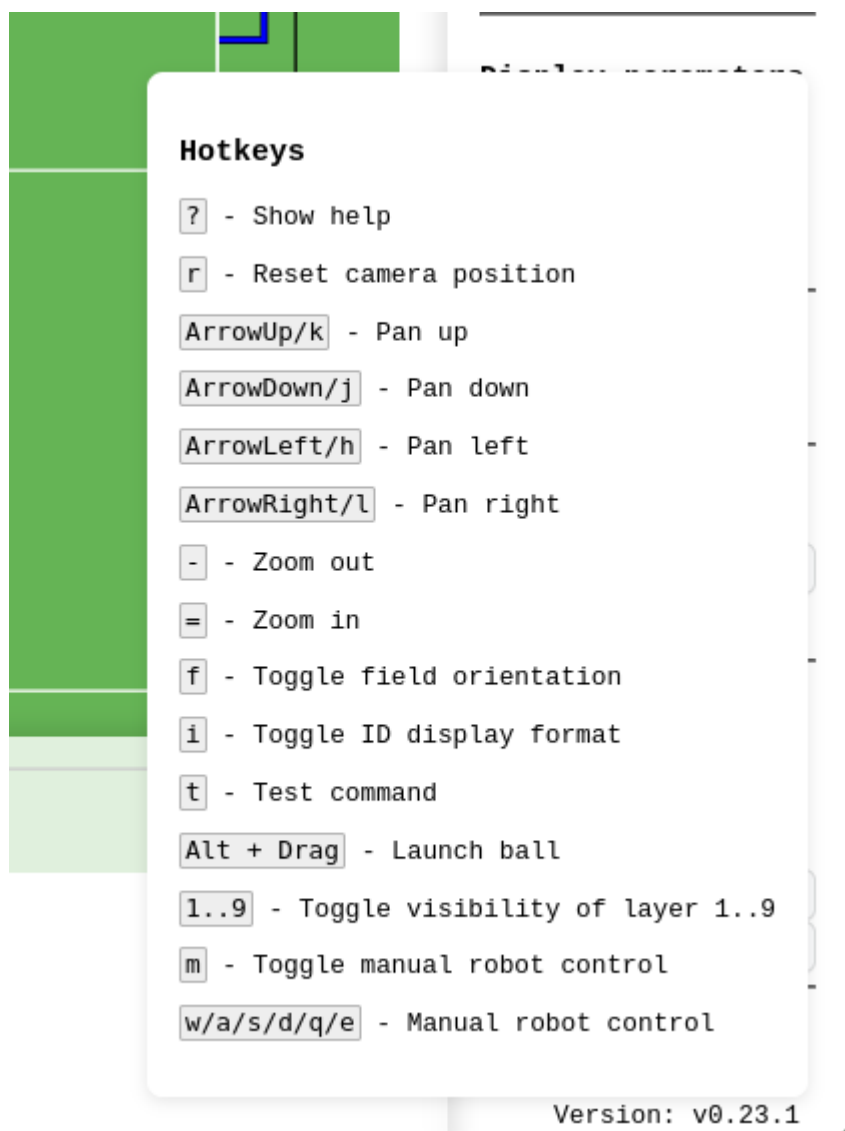
- кнопки для выбора формаций роботов,
- информацию об отображаемых слоях, их названия и настройки видимости,
- кнопки сброса данных отображения слоев и телеметрии,
- помощь по горячим клавишам,
- номер версии программы.

При включении/выключении функции управления роботами открывается дополнительная область интерфейса, позволяющая осуществлять настройки управления (номер и цвет робота, скорости) с клавиатуры.



The screenshot displays the 'Robot control' interface. On the left, there are three sections: 'Robot control' with a checkbox, 'Formations' with buttons for 0, 1, 2, and 'Random formation', and 'Layers'. The main area on the right is titled 'Robot control' and contains a checked checkbox, a dropdown menu set to 'Blue', a numeric input set to '3', a slider for 'Linear vel [mm/s]' set to '1000', and a slider for 'Angular vel [rad/s]' set to '2'.

По набору символа ? можно открыть справку по горячим клавишам, которыми можно взаимодействовать с интерфейсом.

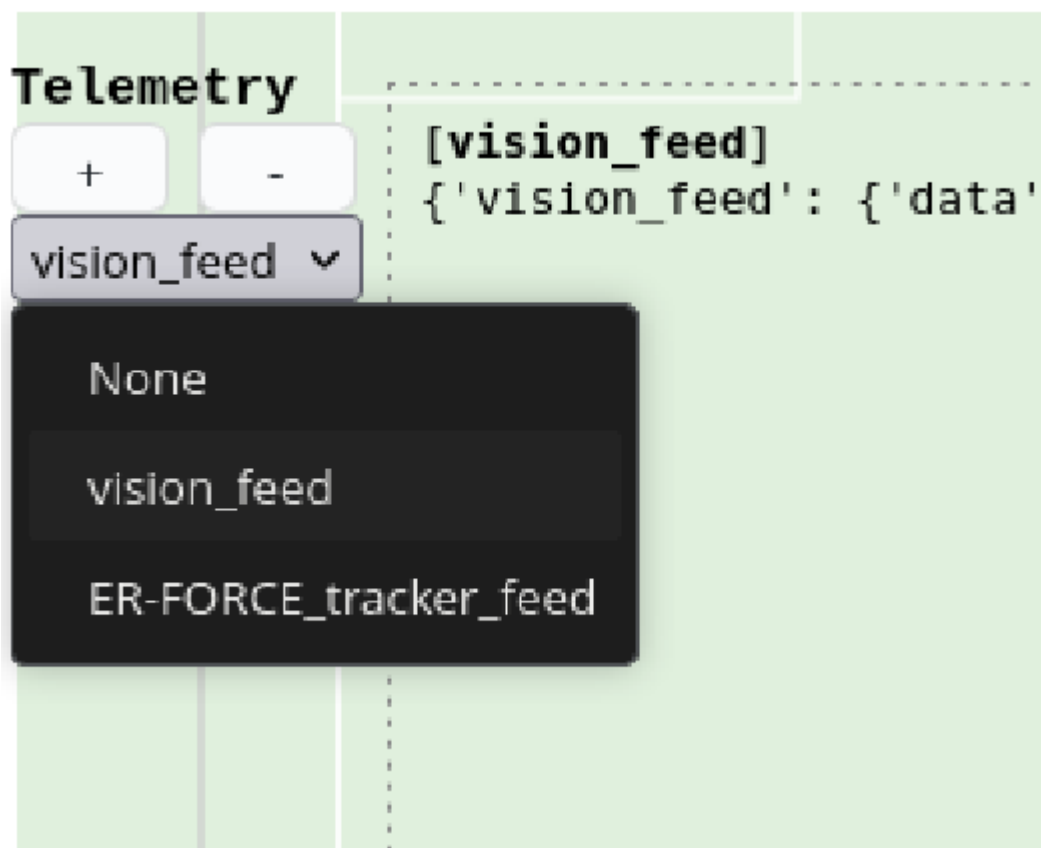


## Меню телеметрии

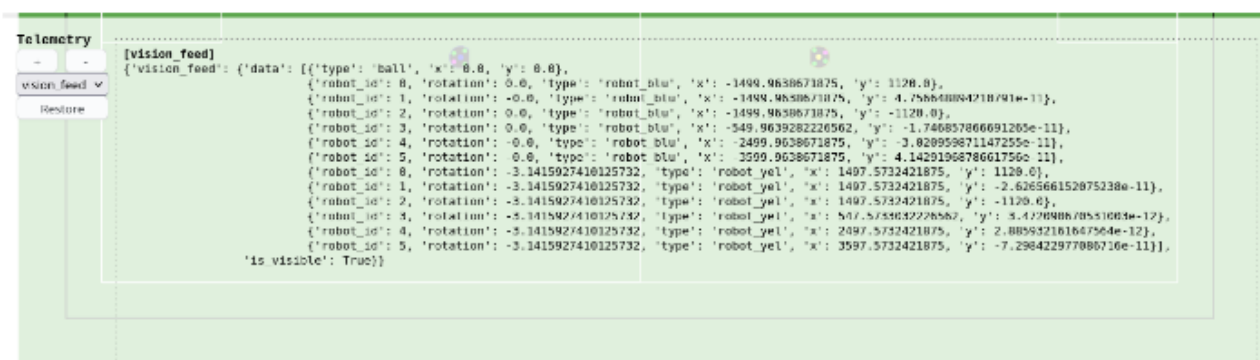
В нижней части окна программы расположено окно телеметрии. Телеметрией в RASMA CS является произвольный поток текстовых данных, обновляющийся в реальном времени. В отличие от логов программы при работе с телеметрией главным является не сохранение прошлых значений данных, а отображение максимально новых значений с минимальной задержкой.

Любая программа может публиковать свои данные телеметрии в RASMA CS для отображения в интерфейсе сервиса. Каждый поток данных характеризуется названием и полезной нагрузкой.

В настройках отображения телеметрии можно выбрать поток данных, который будет отображаться в интерфейсе Serviz.



При выборе он автоматически начинает отображаться в интерфейсе Serviz.



При необходимости можно выбрать несколько потоков для отображения одновременно.





Максимальное количество потоков для отображения - 6.

Для удобства пользователю также предоставляется возможность настраивать отображение текста телеметрии. Доступны настройки цвета текста и его выделения (наклонный, зачеркнутый, подчеркнутый, жирный), а также любые символы юникода. Текст телеметрии, приведенный ниже, будет отображен в сервисе как на рисунке ниже.



<x><r>Bold Red Text</r></x>  
<i><b>Italic Blue Text</b></i>  
<u><g>Underlined Green Text</g></u>

| Emoji                | Unicode      |
|----------------------|--------------|
| :red_circle:         | &#x0001f534; |
| :orange_circle:      | &#x0001f7e0; |
| :yellow_circle:      | &#x0001f7e1; |
| :green_circle:       | &#x0001f7e2; |
| :large_blue_circle:  | &#x0001f535; |
| :purple_circle:      | &#x0001f7e3; |
| :brown_circle:       | &#x0001f7e4; |
| :black_circle:       | &#x0026ab;   |
| :white_circle:       | &#x0026aa;   |
| :red_square:         | &#x0001f7e5; |
| :orange_square:      | &#x0001f7e7; |
| :yellow_square:      | &#x0001f7e8; |
| :green_square:       | &#x0001f7e9; |
| :blue_square:        | &#x0001f7e6; |
| :purple_square:      | &#x0001f7ea; |
| :brown_square:       | &#x0001f7eb; |
| :black_large_square: | &#x002b1b;   |
| :white_large_square: | &#x002b1c;   |



**Telemetry**

None ▾

Restore

**Bold Red Text**  
*Italic Blue Text*  
Underlined Green Text

| Emoji                | Unicode |
|----------------------|---------|
| :red_circle:         | 🔴       |
| :orange_circle:      | 🟠       |
| :yellow_circle:      | 🟡       |
| :green_circle:       | 🟢       |
| :large_blue_circle:  | 🟡       |
| :purple_circle:      | 🟣       |
| :brown_circle:       | 🟤       |
| :black_circle:       | ⬛       |
| :white_circle:       | ⬜       |
| :red_square:         | 🔴       |
| :orange_square:      | 🟠       |
| :yellow_square:      | 🟡       |
| :green_square:       | 🟢       |
| :blue_square:        | 🟡       |
| :purple_square:      | 🟣       |
| :brown_square:       | 🟤       |
| :black_large_square: | ⬛       |
| :white_large_square: | ⬜       |

Полный список поддерживаемых настроек отображения телеметрии можно посмотреть ниже.



## Цвета

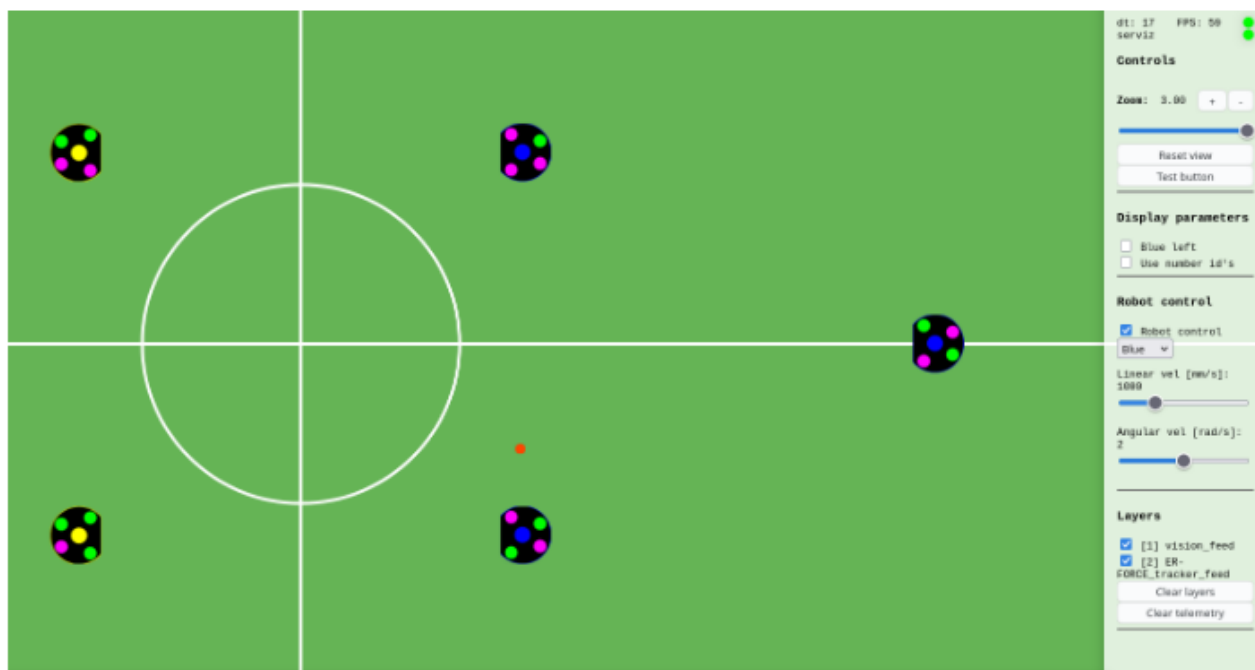
- `<r>` - **красный** (red)
- `<o>` - **оранжевый** (orange)
- `<y>` - **желтый** (yellow)
- `<g>` - **зеленый** (green)
- `<b>` - **синий** (blue)
- `<p>` - **фиолетовый** (purple)
- `<k>` - **черный** (black)
- `<w>` -  (white)
- `<c>` - **циан** (cyan)
- `<m>` - **пурпурный** (magenta)
- `<l>` - **светло-зеленый** (lime)
- `<a>` - **аква** (aqua)
- `<n>` - **синий-синий** (navy)
- `<v>` - **ярко-фиолетовый** (violet)
- `<s>` - **серебро** (silver)

## Курсив, жирность и подчеркивание

- `<x>` - **жирный** (bold)
- `<i>` - *курсив* (italic)
- `<u>` - подчеркнутый (underline)
- `<q>` - ~~зачеркнутый~~ (line-through)

## Графический холст

Центр экрана сервиса занимает графический холст, на котором отображается вся графическая информация всех запущенных программ, которые ее публикуют через API для отображения на экране. Изображение можно перемещать мышкой или стрелками клавиатуры, а также приближать и отдалять. На рисунке ниже приведен пример изображения приближенных роботов на игровом поле.



При необходимости есть возможность изменить режим отображения роботов на поле между реалистичным отображением с отрисовкой маркеров на крыше и минималистичным отображением с их номерами.



(а) Реалистичное отображение роботов с маркерами



(б) Минималистичное отображение роботов с их номерами

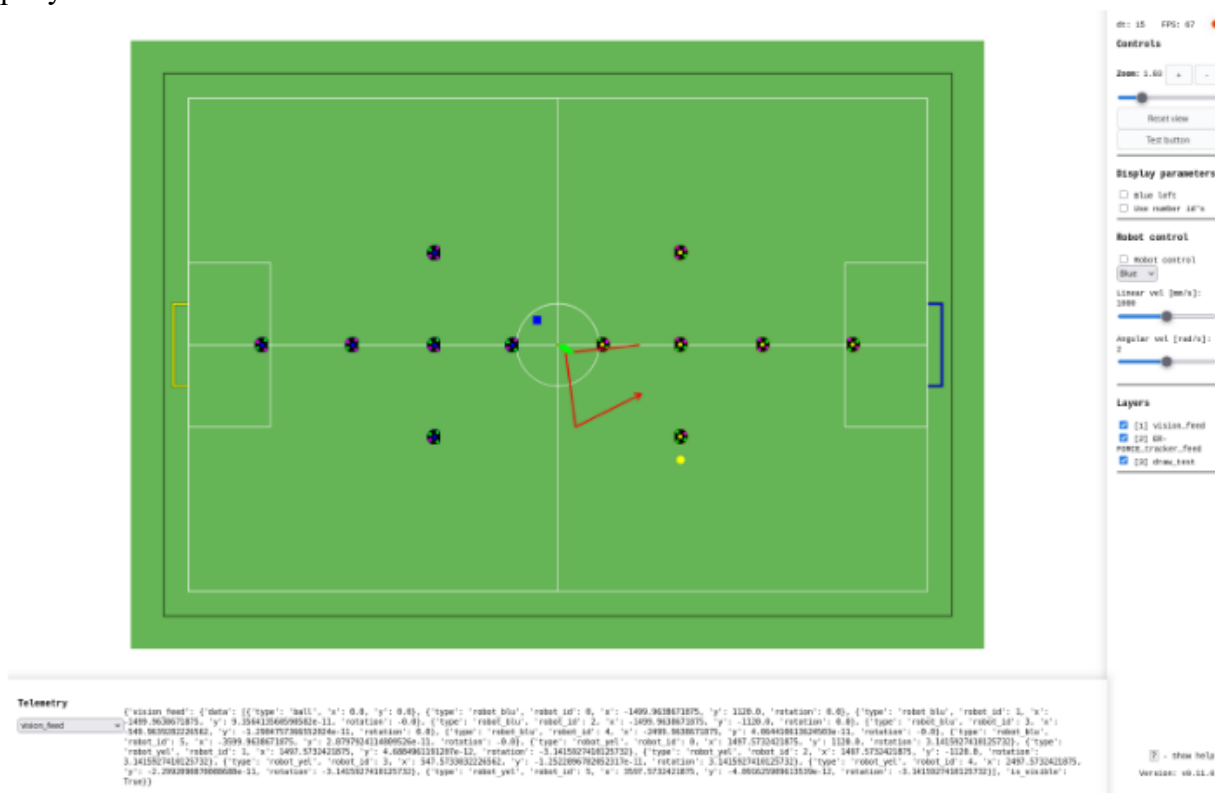
Холст поддерживает ряд графических примитивов для отрисовки, таких как:

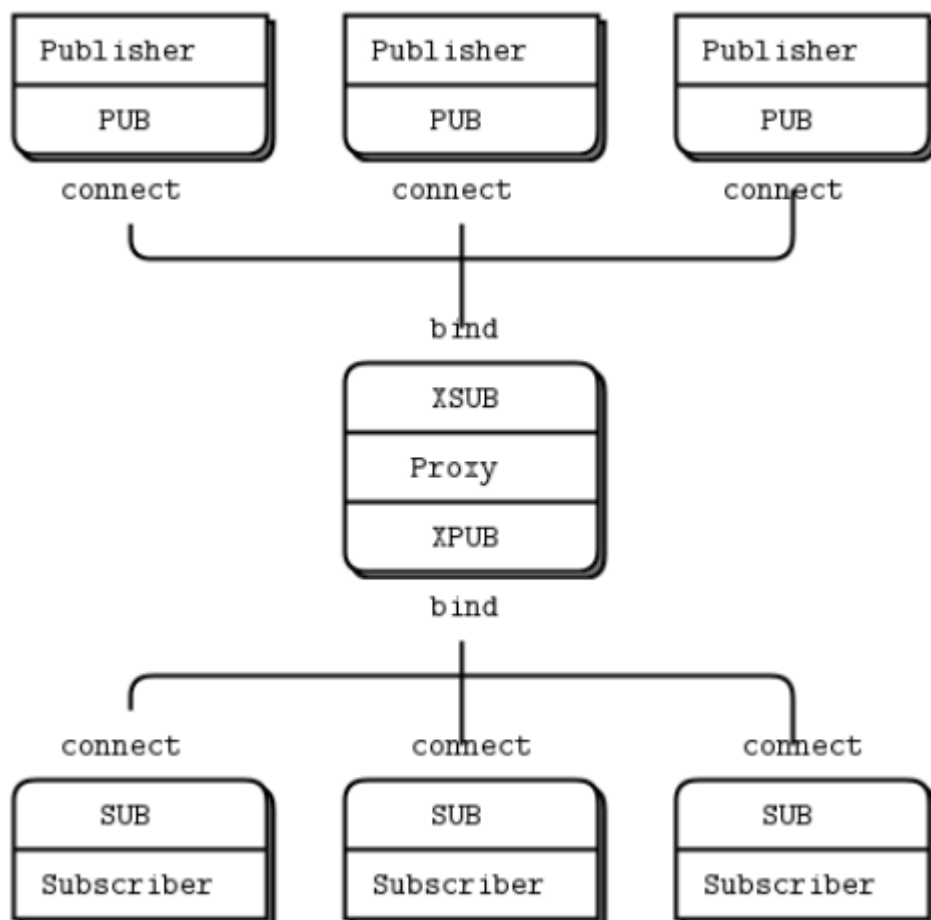
- роботы,
- мяч,
- круги,
- ломанные,



- многоугольники,
- стрелки,
- и др.

Пример использования графических примитивов для отображения на поле можно посмотреть на рисунке ниже.





Такая структура позволяет иметь неограниченное количество модулей, способных писать и читать из одного и того же канала. Модель Pub-Sub гарантирует, что все читающие модули будут получать все сообщения, передающиеся в соответствующий канал. URI каналов определены в файле конфигурации `config.yml` и в актуальной версии их адреса выглядят следующим образом:

ether:

```

api_version: 3
s_draw_pub_url: ipc:///tmp/ether.draw.xpub
s_draw_sub_url: ipc:///tmp/ether.draw.xsub
s_telemetry_pub_url: ipc:///tmp/ether.telemetry.xpub
s_telemetry_sub_url: ipc:///tmp/ether.telemetry.xsub
s_signals_pub_url: ipc:///tmp/ether.signals.xpub
s_signals_sub_url: ipc:///tmp/ether.signals.xsub
  
```

Формат сообщений по каждому из каналов регламентирован в API и описан ниже:

## Draw

В сообщении на отрисовку указывается название слоя, данные для отрисовки и начальное состояние видимости этого слоя.



В одном сообщении можно указать произвольное количество слоев и объектов в каждом из них.

```
{
  "test_vision": {
    "data": [
      <objects to draw>
    ],
    "is_visible": true // Read only on first update
  }
}
```

## Telemetry

В сообщении телеметрии указывается название канала телеметрии и данные телеметрии в виде строки. В одном сообщении можно указать произвольное количество каналов телеметрии одновременно.

```
{
  "test_telemetry": "test telemetry value\nit is just a string"
}
```

## Signals

В сообщении сигнала указывается получатель сигнала, название сигнала и произвольные данные при необходимости.

```
{
  "<recipient>": "<signal_name>",
  "data": {
    /* ... */
  } // optional
}
```

Формат возможных сообщений уникален для всех модулей и может быть произвольным. Существующие виды сообщений приведены в документации, поставляемой вместе с исходным кодом ПО.