

RoboCup 2v2 Championship — Правила турнира 2026

Студенческий турнир · Факультет ВМК МГУ · 2026

1. О турнире

Соревнование по роботизированному футболу **2 против 2**. Участники разрабатывают программу-агента, управляющую командой из двух роботов. Матчи проводятся автоматически на сервере, результаты отображаются в реальном времени в браузере.

2. Призы

Место	Приз
1 место	Рекомендательное письмо на кафедру ММП (Математические методы прогнозирования)
2 место	Рекомендательное письмо на кафедру НДС (Нелинейная динамика и системы управления)

Итоговое место формируется из **результата турнира** и **оценки эссе** (подробнее в разделе 6).

Важно: ML-решение не требуется. Хорошо запрограммированный алгоритм (геометрия, физика, теория управления) оценивается наравне с нейросетями.

3. Поле и физика симуляции

Параметр	Значение
Размер поля	1080 x 640 px
Ворота	у в диапазоне [210, 430] px, по центру каждой стороны
Радиус робота	28 px
Длительность матча	180 с (2 тайма x 90 с)
Шаг симуляции dt	1/60 с

Физика робота

Каждый шаг агент задаёт ускорение **ax, ay** в **[-1, 1]**. Сервер обновляет состояние:

```
vx += ax * ACCEL * dt      (ACCEL = 900 px/s^2)
vy += ay * ACCEL * dt
vx *= DRAG                 (DRAG = 0.90 -> установившаяся скорость ~150 px/s)
vy *= DRAG
x  += vx * dt
```

```
y += vy * dt
```

Физика мяча

- Drag мяча: 0.97 (скользит дальше, чем роботы).
- Упруго отражается от границ поля.
- При ударе получает импульс вдоль направления от центра робота к центру мяча.
- Kick cooldown: ~0.5 с после каждого удара.

Система координат

- (0, 0) — левый верхний угол поля.
 - **Ваши ворота всегда слева (x = 0). Атакуете вправо (x = 1080).**
 - Наблюдение автоматически зеркалируется для каждой команды.
-

4. Наблюдение (observation)

Каждый шаг симуляции сервер передаёт вашему агенту словарь:

```
{
  "my_robots": [
    {"x": float, "y": float, "vx": float, "vy": float,
     "angle": float, "kick_ready": float}, # 0.0 или 1.0
    { ... } # второй робот
  ],
  "opponent_robots": [
    {"x": float, "y": float, "vx": float, "vy": float, "angle": float},
    { ... }
  ],
  "ball": {"x": float, "y": float, "vx": float, "vy": float},
  "score": {"my": int, "opponent": int},
  "time_remaining": float, # секунды до конца тайма
  "period_seconds": float, # общая длительность тайма
  "field": {
    "x": 0.0, "y": 0.0,
    "width": 1080.0,
    "height": 640.0,
    "goal_y_top": 210.0, # верхний край ворот
    "goal_y_bottom": 430.0, # нижний край ворот
  }
}
```

5. Формат сдачи — .pkl файл

Сдаётся **один файл** .pkl — Python-класс, сохранённый через cloudpickle.

Требования

- Класс должен иметь метод `get_actions(self, observation) -> list`.
- Метод возвращает список из 2 действий: `[[ax0, ay0, kick0], [ax1, ay1, kick1]]`
- Файл сохраняется через `cloudpickle` (не стандартный `pickle`).
- Максимальный размер файла: **512 КБ**.

Запрещено внутри `get_actions`

Запрещены: файловый I/O, сеть, `subprocess`, системные вызовы, бесконечные циклы.

Разрешены: `math`, `numpy`, `scipy`, `torch`, `stable-baselines3`, `scikit-learn`, `numba`, `sympy`, `pandas`, `optuna` и любые чистые вычислительные библиотеки.

Шаблон минимального агента

```
import cloudpickle, math

class MyModel:
    def get_actions(self, obs):
        actions = []
        for robot in obs["my_robots"]:
            dx = obs["ball"]["x"] - robot["x"]
            dy = obs["ball"]["y"] - robot["y"]
            d = math.hypot(dx, dy) or 1.0
            kick = 1.0 if d < 60 and robot["kick_ready"] >= 1.0 else 0.0
            actions.append([dx/d, dy/d, kick])
        return actions

with open("my_model.pkl", "wb") as f:
    cloudpickle.dump(MyModel(), f)
```

Если модель вылетает

Если `get_actions` вызывает исключение — роботы этой команды замирают на месте, матч продолжается. При систематических ошибках (> 10%) засчитывается техническое поражение.

6. Оценка и эссе

Итоговое место = результат турнира + оценка эссе.

Эссе (PDF, до 2 стр.)

Вместе с моделью необходимо предоставить эссе. В нём нужно описать:

- **О себе** — кратко: курс, группа, средний балл.
- **Оценки и предметы** — какие дисциплины понравились и почему.
- **Достижения** — олимпиады, проекты, стажировки, спортивные достижения, личные прорывы, хобби и интересы (если есть).
- **Ожидания от научного руководителя** — что хотите изучать, формат работы.

Описание решения (отдельный раздел или файл)

- Что пробовали: рассматриваемые подходы и причины отказа.
 - Что выбрали: почему этот метод / алгоритм / архитектура.
 - Результаты: счёт в тестовых матчах, против каких соперников.
-

7. Как пользоваться платформой

Регистрация / Вход — кнопки «Войти» и «Регистрация» в правом верхнем углу.

Главная — краткое описание турнира, ссылки на скачивание правил и примера модели.

Лидерборд — рейтинговая таблица и список последних матчей.

Матчи — товарищеские матчи:

- «Ваша модель» — загрузите .pkl кнопкой «Загрузить модель».
- «Быстрый матч» — нажмите «Найти соперника», чтобы войти в очередь.
- «Вызов» — выберите соперника из списка и нажмите «Бросить вызов».
- «Входящие вызовы» — здесь появятся вызовы от других игроков.

Трансляция — выберите идущий матч. Матч отображается в реальном времени.

Турнир — зарегистрируйтесь (имя, фамилия, группа), загрузите турнирную модель. Регистрация открыта до **14 апреля 20:00 МСК**.

8. Система очков

Результат	Очки
Победа	+3
Ничья	+1
Поражение	0

Товарищеские матчи формируют **временную таблицу лидеров** — ориентир, не влияет на итог.

Финальная таблица строится только по матчам финального турнира.

9. Сроки

Этап	Дата
Открытие турнира	3 апреля 2026
Дедлайн загрузки финальных моделей	14 апреля 2026, 20:00 МСК
Начало финального турнира	15 апреля 2026, 10:00 МСК
Дедлайн приёма эссе для топ-5	15 апреля 2026, 23:00 МСК
Итоги и награждение	16 апреля 2026

10. Советы по разработке

- **Разделяйте роли:** один робот атакует, другой защищает ворота.
- **Kick cooldown:** проверяйте `kick_ready == 1.0` перед ударом.
- **Предсказывайте мяч:** используйте `ball.vx` / `ball.vy` для опережающего перехвата.
- **Нет нужды в ML:** чёткий геометрический алгоритм может обыграть нейросеть.
- **Тестируйте локально:** скачайте репозиторий геймастера и запускайте матчи у себя.