

# Snake Drift Rush

Браузерная аркадная гонка змей (текущий этап: офлайн MVP + расширенная отладка).

## Текущее состояние

- Рендер и игровой цикл на `Phaser 3`.
- Фоновая музыка на первой трассе (`canyon_loop`): `assets/sounds/Formula1.mp3`.
- Переключаемые офлайн-режимы:
- Классика: 1 игрок + 3 бота.
- Отладка: 4 бота на автопилоте.
- Экраны: `Main Menu`, `Snake Select`, `Track Select`, `Race HUD`, `Results`.
- 7 уникальных трасс.
- Добавлены новые трассы:
- `volcano_spiral` — плотные дуги и волнообразные связи.
- `glacier_chicane` — длинные прямые с быстрыми шиканами.
- 4 типа змей с разными характеристиками.
- 4 основных пикапа: `BOOST`, `SHIELD`, `OIL`, `BOMB`.
- Активная способность: плевок ядом (`Space` для игрока, автоприменение у ботов).
- Логика поверхности:
- `road` — обычное движение.
- `offroad` — заметное замедление (усилено).
- `outside` — сброс на последний чекпоинт + штраф по времени.
- Фазы заезда: `Отсчет` -> `Гонка` -> `Финиш`.
- Локальная таблица лучших времен по трассам (`localStorage`).

## Новые механики тела змеи

- У каждой змеи есть сегментированное тело.
- Стартовая длина у всех змей: 8 сегментов.
- Визуальный размер сегментов тела увеличен на 50% для лучшей читаемости.
- Тело извивается в движении.
- Для каждого типа змеи свой профиль извивки:
- разный шаг между сегментами;
- разная амплитуда волны;
- разная частота и скорость волны;
- разная "толщина" хвоста.
- Правило истощения:
- каждые 12 секунд без еды змея теряет 1 сегмент;
- если сегментов меньше 4, змея движется очень медленно;
- каждый шаг голода дополнительно снижает скорость;
- после 3 шагов голода подряд змея не выбывает, а переходит в режим ползком (держит минимальный ход);
- кактусы и укусы хвоста уменьшают длину, но не увеличивают счетчик истощения.

## Дополнительные случайные объекты на трассе

- Добавлены случайные яблоки и кактусы.
- Во время гонки они спавнятся в случайных точках на дороге и респавятся после подбора.

- Эффект подбора:
- яблоко увеличивает длину тела на  $+1$  сегмент;
- кактус уменьшает длину тела на  $-1$  сегмент.
- Яблоко также сбрасывает счетчик истощения.
- Поведение ботов: по ходу движения боты стараются подбирать яблоки, если они находятся впереди по траектории и в достижимом радиусе.
- Чем голоднее бот (выше счетчик голода), тем сильнее его тянет к яблокам: он агрессивнее смещает траекторию к яблоку и готов брать более дальние цели.
- BOMB ослаблена: уменьшены радиус поражения и длительность дебафа.
- После попадания BOMB действует короткая защита от повторного попадания подряд, чтобы избежать «вечного залипания».
- Спавн яблок/кактусов разнесен (без плотных наложений и без близкой постановки к чекпоинтам/пикапам), чтобы боты не клинили в одной точке.

## Таблица: объекты, бафы и дебафы на трассе

Название	Форма	Цвет	Размер	Эффект	Время эффекта
BOOST	Ромб (пикап)	Ярко-зелёный	Подбор: radius=12 (визуал ромб ~ 14x14)	Баф скорости и разгона (speed $\times 1.34$ , accel $\times 1.18$ )	2600 ms
SHIELD	Ромб (пикап)	Небесно-голубой	Подбор: radius=12 (визуал ромб ~ 14x14)	Щит: $1$ заряд, блокирует 1 негативный эффект/удар	До 6500 ms или до срабатывания
OIL	Ромб (пикап)	Янтарно-оранжевый	Подбор: radius=12 (визуал ромб ~ 14x14)	Дебаф управления/разгона (turn $\times 0.64$ , accel $\times 0.82$ )	2200 ms
BOMB	Ромб (пикап) + радиус взрыва	Кораллово-розовый	Подбор: radius=12, зона взрыва: radius=86	Накладывает BOMB_SLOW (speed $\times 0.84$ , accel $\times 0.86$ , turn $\times 0.93$ )	1450 ms + защита от повторного попадания 1200 ms
APPLE	Яблоко	Красно-розовый	Объект: radius=11	$+1$ сегмент тела, сброс шага голода	Мгновенно
CACTUS	Кактус	Травяной зелёный	Объект: radius=11	$-1$ сегмент тела	Мгновенно

## Контакт с телами змей

- Переползание через тело другой змеи замедляет.
- handler при переползании через чужое тело, наоборот, ускоряется.
- trickster не имеет иммунитета к замедлению от переползания.
- speedster не может переползать через чужие тела (получает блок и отталкивание).
- bully при контакте отталкивает соперника примерно на один сегмент и сам не замедляется от этого столкновения.
- trickster может откусить  $1$  сегмент при касании хвоста другой змеи (с небольшим кулдауном между укусами).
- Усилено антизалипание: если змея теряет прогресс и почти стоит, включается авто-вывод вперед по траектории.
- Для speedster улучшен выход из блокировки на чужом теле, чтобы не зависать в правых поворотах.

# Спрайты змей

- Поддерживается подключение спрайтов головы и сегментов тела для каждого типа змей.
- Путь к файлам:
  - assets/sprites/snakes/speedster/head.png
  - assets/sprites/snakes/speedster/segment.png
  - assets/sprites/snakes/handler/head.png
  - assets/sprites/snakes/handler/segment.png
  - assets/sprites/snakes/bully/head.png
  - assets/sprites/snakes/bully/segment.png
  - assets/sprites/snakes/trickster/head.png
  - assets/sprites/snakes/trickster/segment.png
- Если у типа змеи спрайтов нет, автоматически используется текущий векторный рендер (игра не ломается).

## Влияние длины тела на эффекты

- При увеличении тела усиливаются полезные бафы (например, BOOST).
- При уменьшении тела ослабляются вредные эффекты (OIL, BOMB\_SLOW).
- Текущая длина тела отображается в HUD рядом с активным эффектом.

## Перки, бафы и дебафы (полный список)

- speedster: самая высокая скорость; не переползает чужие тела; минимальная дальность яда.
- handler: лучший контроль в поворотах; ускоряется при переползании через чужое тело; средняя дальность яда.
- bully: тяжёлый корпус и отталкивание соперника; сильный контактный контроль; дальность яда выше средней.
- trickster: может откусывать сегмент хвоста соперника; дальний яд.
- BOOST: временный баф скорости/разгона.
- SHIELD: 1 заряд щита, блокирует один негативный эффект/удар.
- OIL: дебаф управления и разгона.
- BOMB: радиусный дебаф замедления (ослаблен; добавлена короткая защита от повторного цепного попадания).
- VENOM\_SLOW (яд): змеи могут плевать ядом; у каждого типа своя дальность/кулдаун/сила замедления.
- яблоко: +1 сегмент тела, сброс шага голода.
- кактус: -1 сегмент тела.
- Истощение: каждые 12 секунд без яблока -1 сегмент; змея не умирает от голода, а всё сильнее замедляется и после 3 тиков ползёт на минимальной скорости.

## Таблица: перки типов змей

Название	Форма	Цвет	Размер	Эффект	Время эффекта
speedster	Змея (голова + сегменты)	Ярко-бирюзовый	База тела в гонке: 8 сегментов	Самая высокая скорость; не переползает через чужие тела (получает блок/отталкивание); яд с наименьшей дальностью (range=128)  Лучший контроль; при	Пассивно всю гонку; яд: 1450 ms, КД 2650 ms

handler	Змея (голова + сегменты)	Салатовый	База тела в гонке: 8 сегментов	переползании через чужое тело ускоряется ( $\times 1.08$ , до $1.18$ от $\max \text{speed}$ ); яд средней дальности ( $\text{range}=168$ ) При столкновении отталкивает соперника ( $\sim 8$ px) и сам не получает обычный штраф столкновения; яд выше средней дальности ( $\text{range}=182$ )	Пассивно всю гонку; яд: 1650 ms, КД 2450 ms
bully	Змея (голова + сегменты)	Лососевый	База тела в гонке: 8 сегментов	Может откусывать хвост ( $-1$ сегмент, КД укуса 900 ms); самый дальний яд ( $\text{range}=212$ )	Пассивно всю гонку; яд: 1800 ms, КД 2520 ms
trickster	Змея (голова + сегменты)	Лавандовый	База тела в гонке: 8 сегментов		Пассивно всю гонку; яд: 1900 ms, КД 2200 ms

## Управление

- W/A/S/D или стрелки — движение (в режиме Классика).
- Space — плевок ядом.
- R — быстрый рестарт текущей трассы.

## Размер игрового окна

- Игровая область (`race-stage`) увеличена еще сильнее: на десктопе почти на весь экран (порядка 95% ширины), на мобильных — максимально крупно без ручного zoom.

## Docker Compose стек

Проект собран в переносимый стек сервисов:

- web — фронтенд (Nginx, статика игры).
- match-server — Colyseus authoritative match-server.
- nakama — backend для leaderboard/RPC.
- postgres — база данных Nakama.
- Поток результатов: `match-server -> nakama RPC submit_race_time -> leaderboard`.
- Для офлайн `userId` match-server генерирует детерминированный `owner_id` (UUID), чтобы Nakama корректно записывал результаты.
- postgres работает только во внутренней сети docker-compose (порт наружу не пробрасывается).
- Данные Postgres пишутся в папку проекта `./_data/postgres` (bind mount), поэтому не теряются при перезапуске Docker/контейнеров.

## Запуск через Docker (рекомендуется)

1. Скопировать переменные окружения:

```
cp .env.example .env
```

По умолчанию БД хранится в папке `./_data/postgres` внутри проекта.

## 2. Поднять весь стек:

```
docker compose up --build
```

## 3. Сервисы будут доступны:

- **Игра:** `http://localhost:5500` (дополнительно доступна на `http://localhost:8080`)
- **Match-server health:** `http://localhost:2567/healthz`
- **Nakama API:** `http://localhost:7350`
- **Nakama Console:** `http://localhost:7351`
- **Логин в Nakama Console:** из `.env` (`NAKAMA_ADMIN_USER` / `NAKAMA_ADMIN_PASSWORD`)

## 4. Остановить стек:

```
docker compose down
```

Полный сброс с удалением тома БД:

```
docker compose down -v
```

Для `bind mount` (`POSTGRES_DATA_DIR`) команда выше не удаляет файлы БД в папке хоста. Для `bind mount` `./_data/postgres` команда выше не удаляет файлы БД в папке проекта.

Если на macOS видишь ошибку `dependency postgres failed to start`, проверь, что `./_data/postgres` пустая при первом запуске (без `.gitkeep` и прочих файлов), затем подними Postgres заново:

```
rm -f ./_data/postgres/.gitkeep
docker compose up -d postgres
docker compose logs --tail=80 postgres
```

# Локальный запуск без Docker (опционально)

Этот режим нужен только для быстрого офлайн-прогона фронтенда без серверной части.

1. Открыть терминал в корне проекта (`.../_hackathon`).
2. Запустить сервер одним из способов:

```
python -m http.server 5500
```

или

```
npx serve . -l 5500
```

3. Открыть в браузере `http://localhost:5500`.
4. Нажать `Play Offline`.
- 5.

5. Выбрать режим (Классика или Отладка) и змею.
6. Нажать Старт.

Быстрый вариант для черновой проверки: можно открыть `index.html` напрямую, но для стабильной работы спрайтов и ассетов лучше использовать сервер.

## Правило по документации

- Все новые игровые механики и заметные изменения фиксируются в этом `README.md` сразу после реализации.