1UP 000000 HI-SCORE 999999 2UP 000000

ML своими руками на JS

Троицкий Роман, ПАО Сбербанк | ЦЦРР

▶ PRESS START INSERT COIN · HOLYJS 2026

WORLD 0 1 2 3 4 5 6 7 8

Где же ошибка?

SPOT THE DIFF

Expected

Сохранить

VS

Actual

Diff

Сохранить

WHAT?

Или глаза, или скриншот

Скриншот показывает область

Хотим, чтобы агент видел, что именно

Что в результате

P1 WHERE

Playwright: где



CO-OP

P2 WHAT

POWER-UPS ×5

design-dealer: что

0.94	0.71	0.08	0.05	0.03
color	font-weight	letter-spacing	line-height	border

```
playwright test

$ npx playwright test  ✘ 1 failed
button--primary: область 12,40-180,72: color (0.94), font-weight (0.71)
```

WHOAMI

SELECT PLAYER

- Влюблён во фронтенд;
- Senior FE => DSE;
- Студент магистратуры ИТМО;
- Вхожу в ПК HolyJS, Сбера, MoscowCSS;
- I use Arch btw;
- I <3 Vue!



P1 STATS

FRONTEND	■■■■■■■■■■	MAX
ML / DS	■■■■■■■■	
COMMUNITY	■■■■■■■■	
VUE	■■■■■■■■■■	MAX

LVL 1 • Баг в пикселях ♥♥♥ SCORE 000005

EASY MODE



CONTINUE?

1

DIFFICULTY

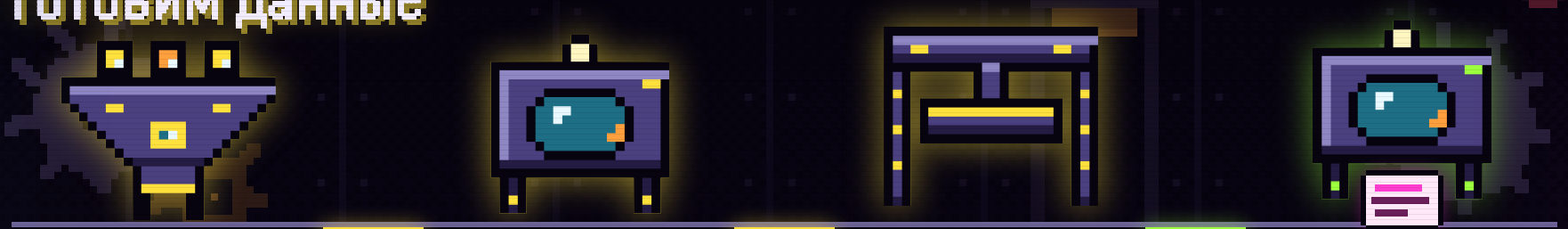
HARD

SAVE

ERR 0X1F

DIFF 3.2

Готовим данные



берём сторибук

генерируем чистые скриншоты

программно искажаем (мутация)

собираем мутированные скриншоты

CLEAN

Заголовок карточки

Основной текст карточки с описанием содержимого.

Обновлено 3 дня назад

≠

MUTANT

Заголовок карточки

Основной текст карточки с описанием содержимого.

Обновлено 3 дня назад

STAGE 3/8

LEVEL 3

ТЕОРИЯ

READY?

Что такое нейросеть



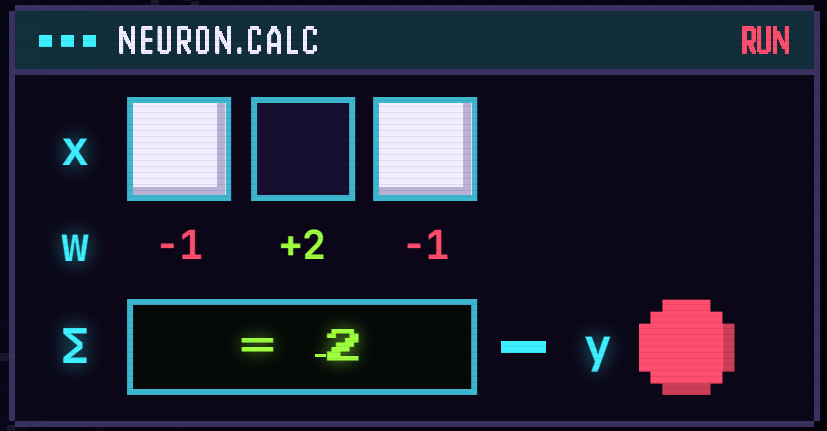
«Великий Человек-паук» (Ultimate Spider-Man), Marvel, 2012

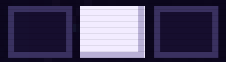
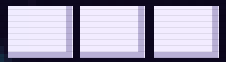
FUNC
Функция с десятками тысяч чисел-весов, которые обучение подгоняет под примеры

NEURON
Нейрон — как нервная клетка: складывает входы с весами и «срабатывает» выше порога

NET
Сеть — нейроны слоями: выход одних — вход для следующих

Как считает один нейрон



1		Σ = 2	горит
2		Σ = 0	молчит
3		Σ = -2	против

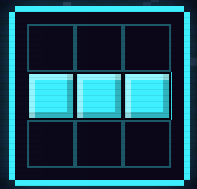
x — **Входы**: пиксели картинки или выходы прошлого слоя

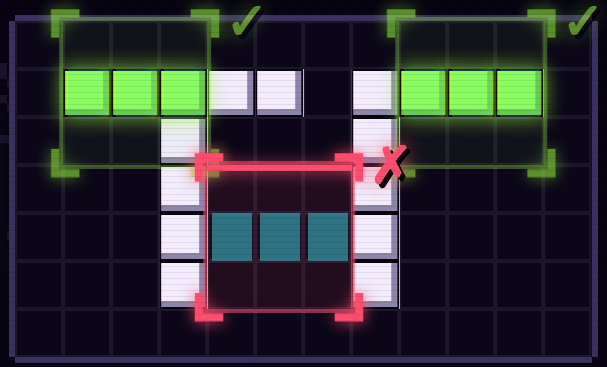
w — **Вес** каждого входа: голос «за», «против» или «всё равно»

Σ — **Сумма**: каждый вход умножаем на его вес и складываем

y — **Выход**: сумма после порога, уходит следующим нейронам

Свёртка



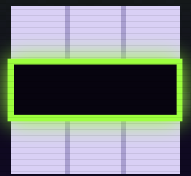


Один и тот же фильтр ищет один и тот же узор (край буквы, линию рамки) в любом месте картинки

Т = 3 ✓

Г = 3 ✓

▪ = 0 ✗



Можно представить, как трафарет

Нейросеть как домик из трафаретов



3 Сохранить

видит ещё крупнее: «буквы стоят слишком далеко друг от друга», «строки разъехались»

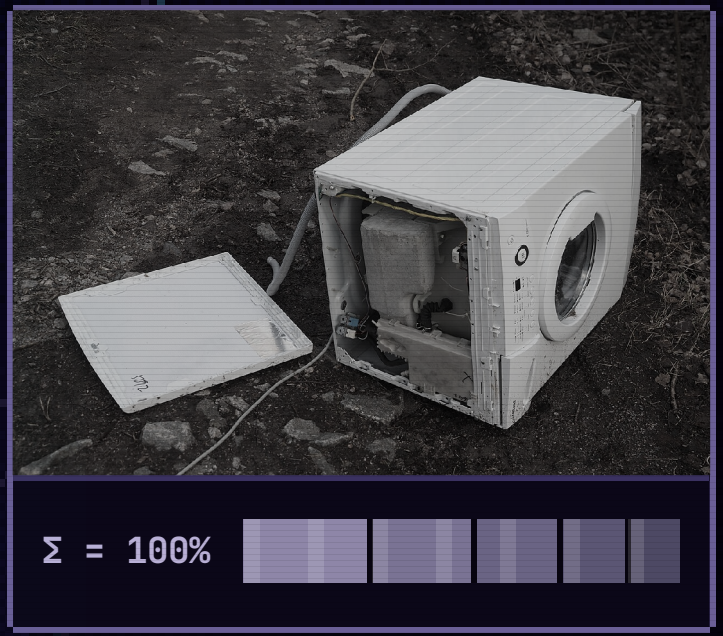
2 Т Г ▭

трафареты к ответам первого этажа: «где палочки стоят рядом?» → буквы и линии рамки

1 | — ⌐ ⌙

много маленьких трафаретов 3×3: палочка, уголок, край буквы

Своя уверенность для каждого свойства



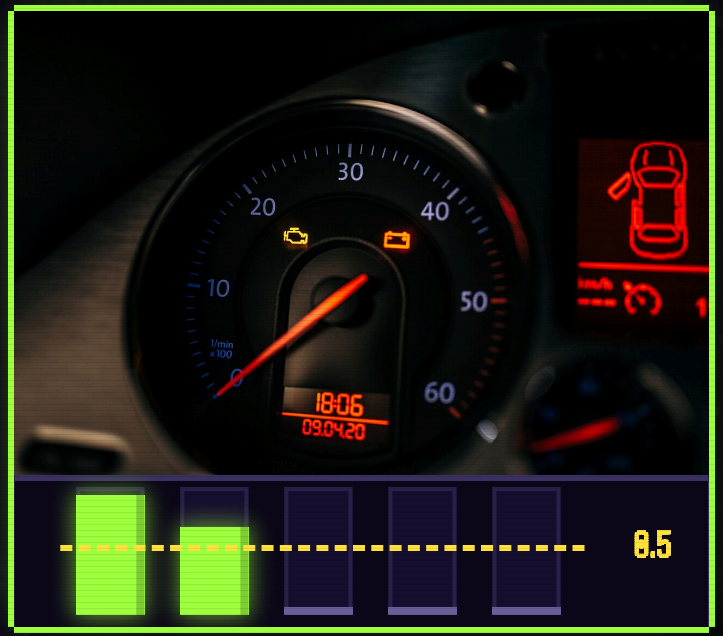


Фото: JIP, CC BY-SA 4.0; Shixart1985, CC BY 2.0 — Wikimedia Commons

model.ts tfjs

```
function buildModel(tf) {
  const conv = (filters) => tf.layers.conv2d({
    filters, kernelSize: 3, padding: 'same', activation: 'relu',
  });

  const shrink = () => tf.layers.maxPooling2d({ poolSize: 2 });

  return tf.sequential({ layers: [
    tf.layers.inputLayer({ inputShape: [64, 64, 7] }),

    conv(16), conv(16), shrink(), // этаж 1: палочки, уголки
    conv(32), conv(32), shrink(), // этаж 2: буквы, рамки
    conv(64), conv(96), // этаж 3: строки разъехались

    tf.layers.globalAveragePooling2d({}), // нашёл где-нибудь?
    tf.layers.dense({ units: 64, activation: 'relu' }),
    tf.layers.dense({ units: 5, activation: 'sigmoid' }), // 5 лампочек
  ] });
}
```

PARAMS ≈100K

dense

GAP

3 64 96

2 32 32 ÷2

1 16 16 ÷2

64×64×7

CONV2D 3×3

RELU

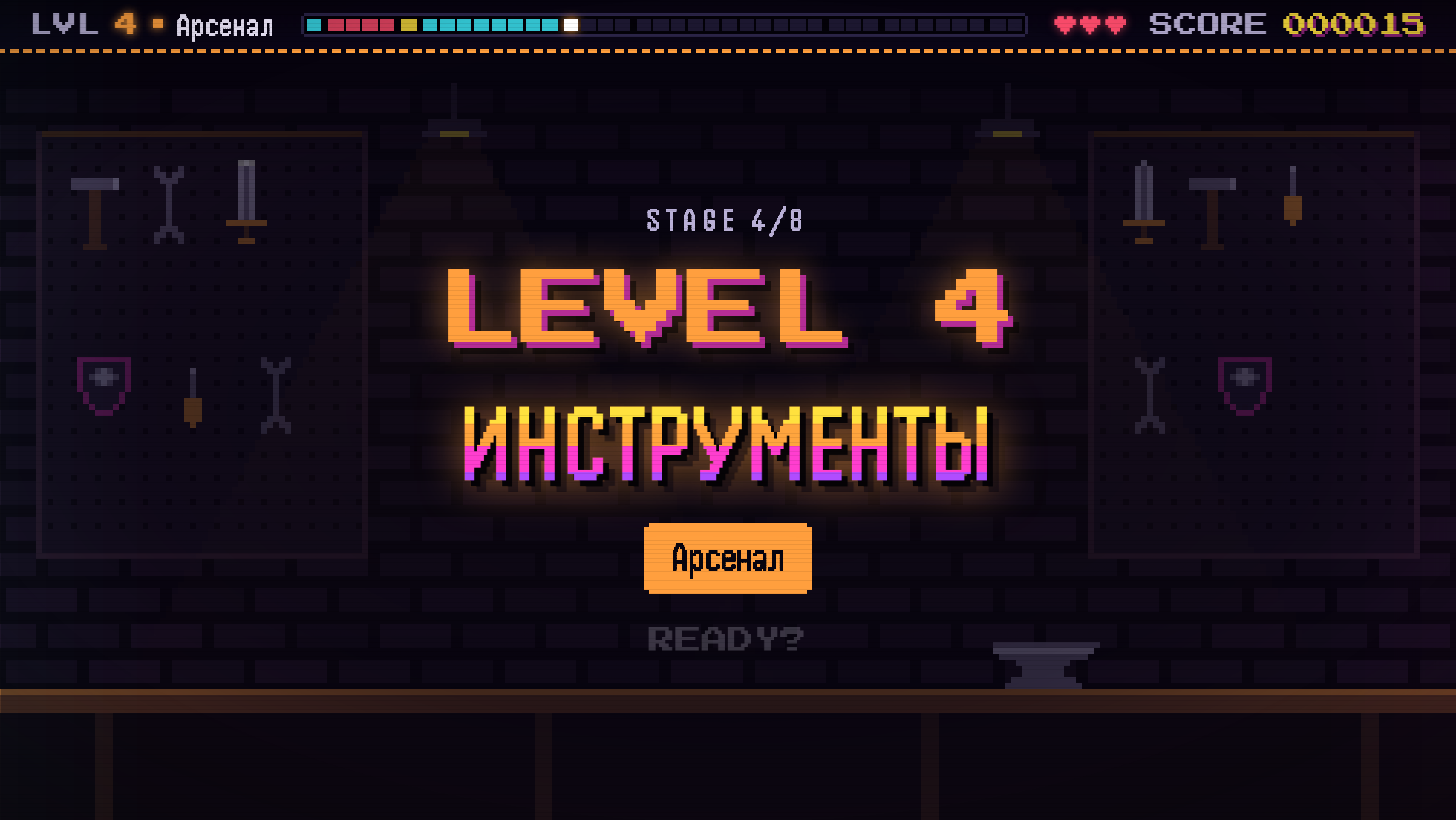
Как сеть учится

Как игра «горячо — холодно»: сеть отвечает, мы говорим, насколько она ошиблась. Это число: **loss**

После каждой подсказки сеть чуть-чуть подкручивает свои числа. Так тысячи раз

Один проход по всем примерам: одна **эпоха**, как прочитать весь учебник




LVL 4 • Арсенал
SCORE 000015
STAGE 4/8
LEVEL 4
ИНСТРУМЕНТЫ
Арсенал
READY?

Кто что делает

PACK

Vue 3 и Storybook; Vuetify — чужие компоненты для проверки

BRAIN

TensorFlow.js, мозг: сама сеть, её обучение и применение



HANDS

Playwright + Chromium: руки и глаза

LENS

pixelmatch + pngjs: игра «найди отличия», кладёт две фотографии одна на другую и обводит, где не совпало

BELT

tsx/esbuild, vitest

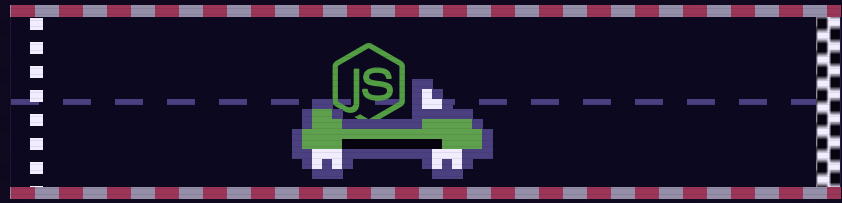
SHIELD

zod проверяет документы на входе и выходе: файл модели, находки, список примеров

Инфраструктура

медленно

JS в Node

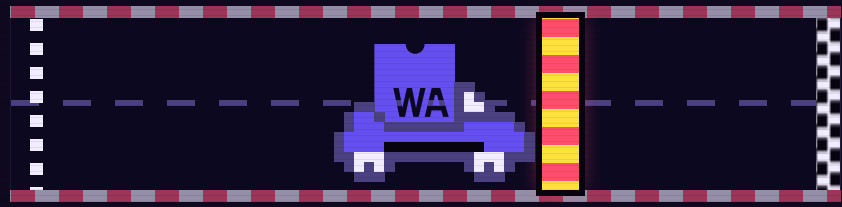


TIME

116 с

неучится

WebAssembly

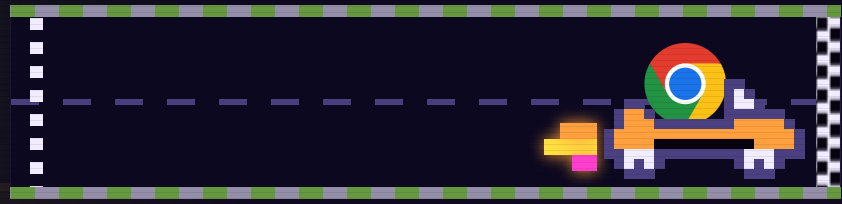


N/A

ТОЛЬКО ОТВЕТЫ

выбрал

Headless Chromium



1ST

<4 с

Инфраструктура







Node
поднимает браузер

Браузер
учит сеть на видеокарте

Веса
возвращаются в Node и ложатся в файлы

2.6 мс применение в Node: на кусочек картинки вместо **61 мс**

LVL 5 • Тренировка ♥♥♥ SCORE 000019

STAGE 5/8

LEVEL 5

КАК ОБУЧАЛИ МОДЕЛЬ



EXP EPOCH 05

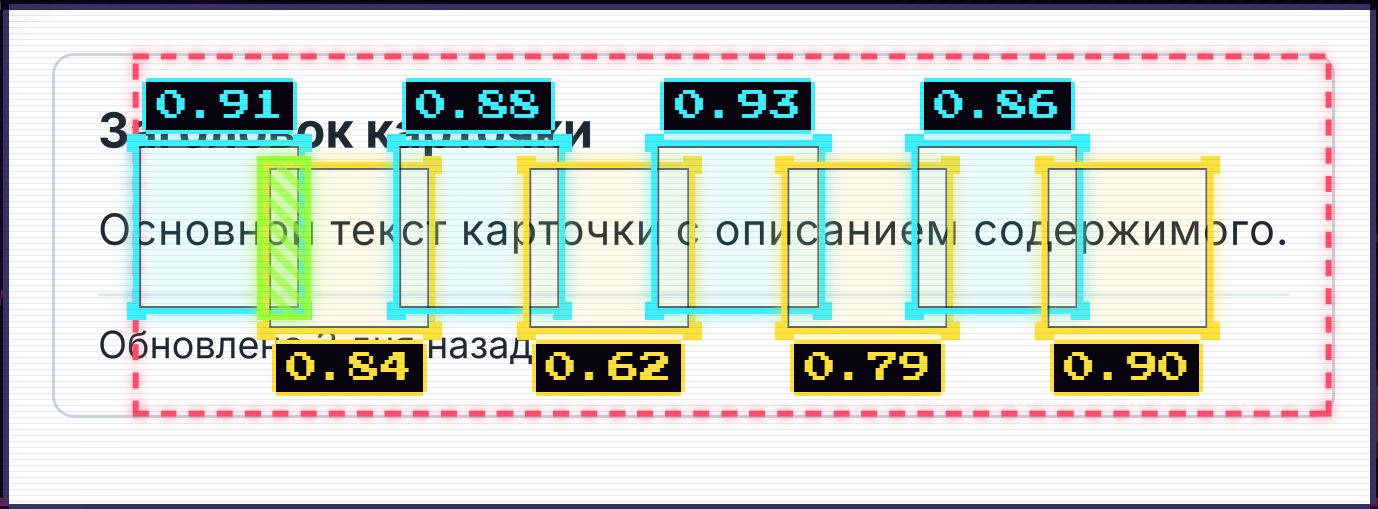
LEVEL UP!

Тренировка

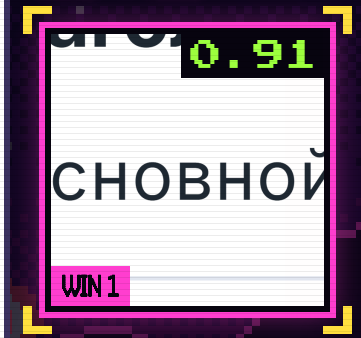
READY?

TRAIN 01 EPOCH 12

Что видит сеть



● SCAN
VOTE → AVG



⌀ 0.84 letter-spacing

Сеть не смотрит на весь компонент

Вокруг каждого отличия вырезаем окошко 64×64 пикселя

Соседние окошки перекрываются на 16 пикселей

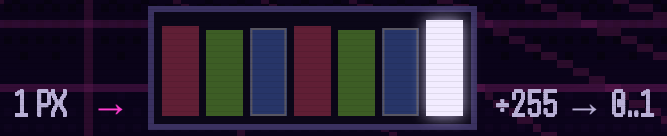
Каждое окошко голосует, ответы окошек одного отличия усредняем

Тензор

Тензор: многомерная таблица чисел, с ними работает TensorFlow.js



Канал	Что в нём
1-3	R, G, B эталона («как надо»)
4-6	R, G, B рендера («как стало»)
= 7	маска диффа: отличается ли этот пиксель



Угадал с первой попытки?

Главная оценка, **top-1**: правда ли самая яркая лампочка указывает на сломанное свойство



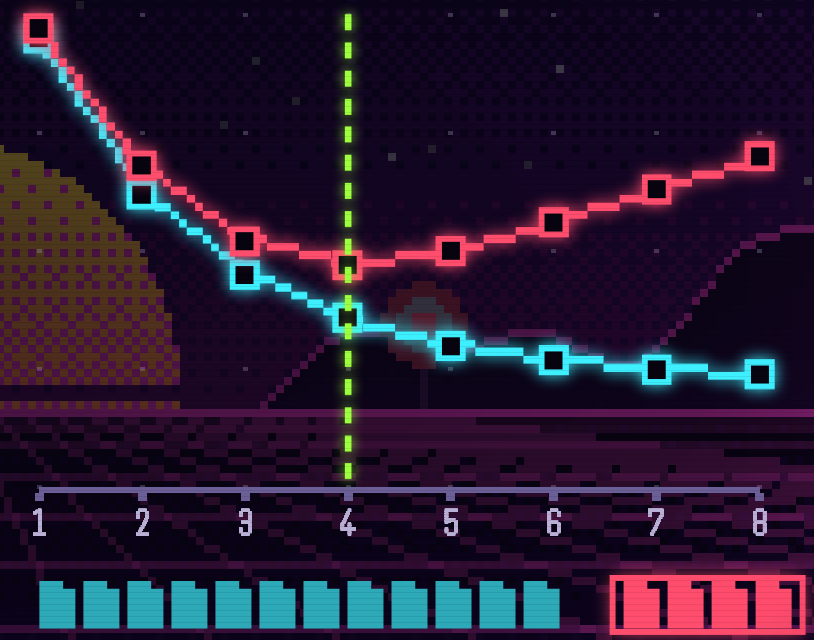
сломан font-weight → ✓

20 % если тыкать наугад

Зубрёжка и контрольная

Главная опасность: зубрёжка. Ученик выучил ответы к задачам из учебника и не решит такую же задачу с другими числами. Это переобучение

Защита: часть задач прячем и не показываем на тренировке. Оцениваем сеть только на них, как контрольную



Проверки

ROUND 1

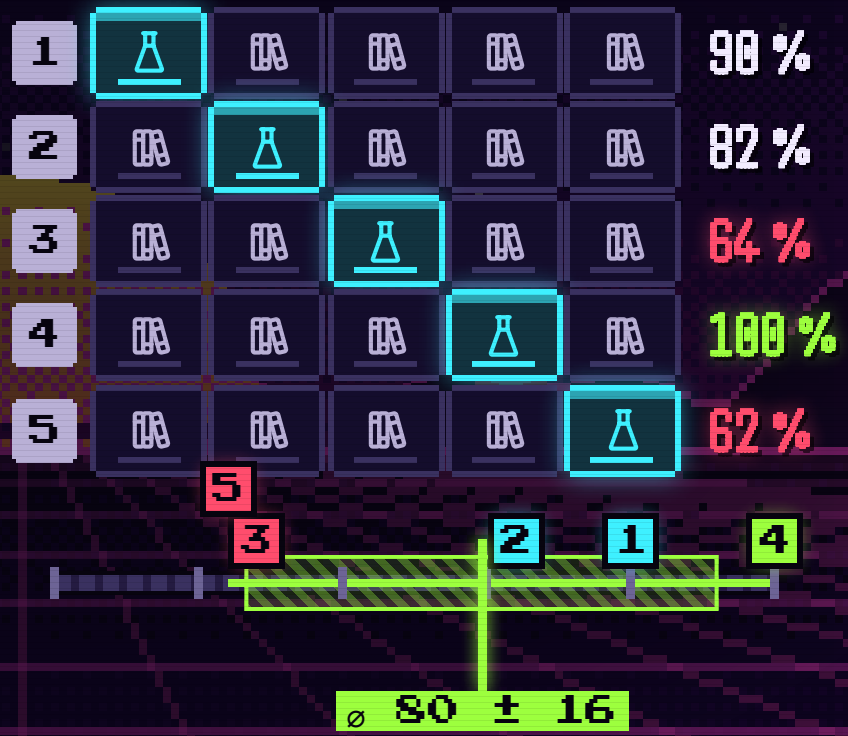
Сначала проверяли сеть на одной story и получили 100 %. Слишком хорошо, чтобы быть правдой

K-FOLD ×5

Делим story на пять кучек. Пять раз учим на четырёх и проверяем на пятой, каждый раз на другой. Это **k-fold**

RESULT

В среднем 80 ± 16 %: разброс между кучками от 62 до 100 %



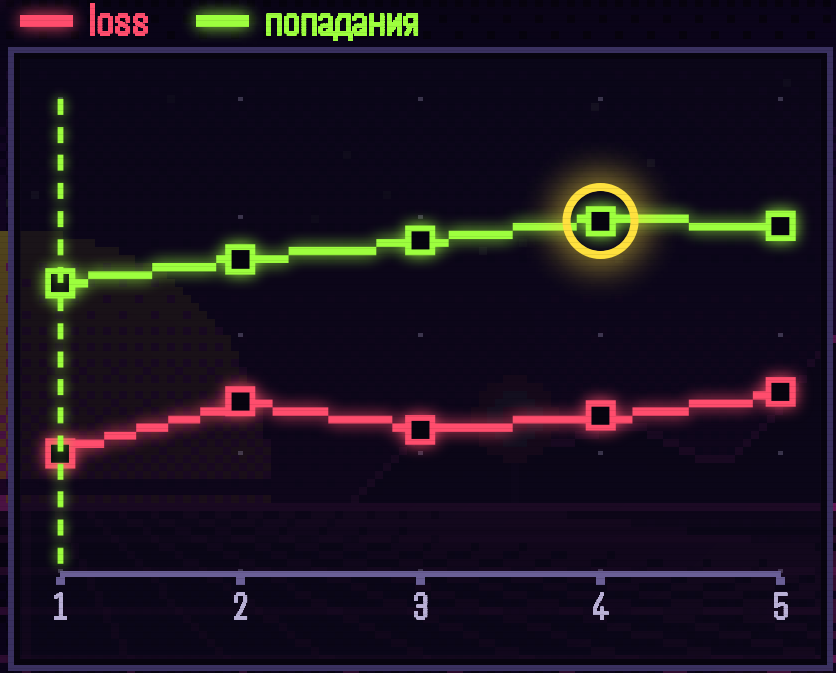
Когда остановить тренировку

ПОПЫТКА 1/3

После каждой эпохи пишем контрольную. Результат три эпохи подряд не стал лучше: стоп, берём лучшую версию. Это **ранняя остановка**

В трёх кучках из пяти лучший loss был после первой эпохи, хотя попаданий становилось больше

loss сильно наказывает за «уверенные ошибки», а попаданию важно только, правильный ли самый яркий ответ

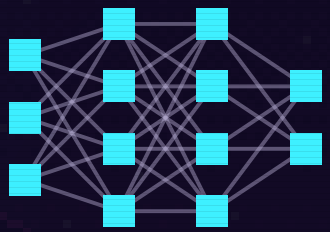


★ 1 2 3 STOP

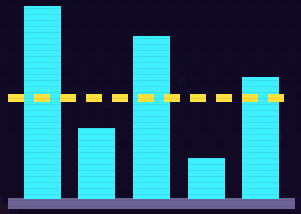
▲ ▲ ★

Против зубрёжки

ПОПЫТКА 2/3

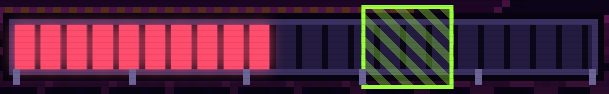


dropout: на тренировке случайно «выключать» часть сети, чтобы она не опиралась на одни и те же числа



L2: штрафовать слишком большие числа

81 % было 85–89 %



Проблема в том, на чём сеть учится

Нейросети как дети

ПОПЫТКА 3/3

ПР▶Меньше строгости, больше разнообразия

12 story

Зубрёжка была оттого, что задачек мало

LEVEL UP!

54 story

18 состояний и у каждого ещё два варианта со своим шрифтом, фоном и размером текста

Aa Aa Aa Aa Aa

Нейросети как дети

= 90 %



Среднее почти не поменялось

÷3



Разброс между кучками сжался втрое: было 68–100, стало 83–93

47 → 81 %



Сеть стала лучше понимать чужие компоненты Vuetify, которых никогда не видела

Распознаём несколько отличий

ANSWER SHEET

✓ ×2

Выход сети с самого начала разрешал несколько галочек, но сеть всегда ставила одну: в каждом примере ломали ровно одно свойство

- ☑ color
- ☑ font-weight
- ☐ letter-spacing
- ☐ line-height
- ☐ border

✓✓

0 → 59 %

Добавили половину примеров с двумя поломками: сеть называет ровно оба свойства, а на обычных хуже не стала

STAGE 6/8

LEVEL 6

МОДЕЛЬ КАК АРТЕФАКТ

Артефакт

READY?

Как устроена сеть и почему она маленькая





97 653

числа подкручиваются при обучении. Для нейросети это мало

Изначально планировали шире (32-…-128), но мы искали минимальную модель, дающую интересные цифры

Как выглядит модель



`model.json`: инструкция по сборке: какие этажи и в каком порядке



`weights.bin`: сами детали: все 97 653 подкрученных числа



`meta.json`: паспорт: какая лампочка какое свойство означает, какого размера окошко, сколько эпох училась



`cv.json`: табель с оценками по пяти кучкам, рядом сами пять проверочных сетей (`folds/0..4/`)

meta.json

```
{
  "registryVersion": "3445fa09…",
  "outputOrder": [
    "color", "font-weight",
    "letter-spacing",
    "line-height", "border"
  ],
  "windowSize": 64,
  "epochs": 6,
  "stopMetric": "top1pair"
}
```

В паспорте есть отпечаток списка свойств (`registryVersion`). Если список поменяли, а сеть старая, это видно сразу

HI-SCORE
1ST 00104
2ND 92111
3RD 85117
4TH 80400
5TH 71230
6TH 64018
7TH 50972
8TH 42001
9TH 30553

PRECISION
0.94
0.81
0.77
0.71
0.62
0.58
0.44
0.31
0.12

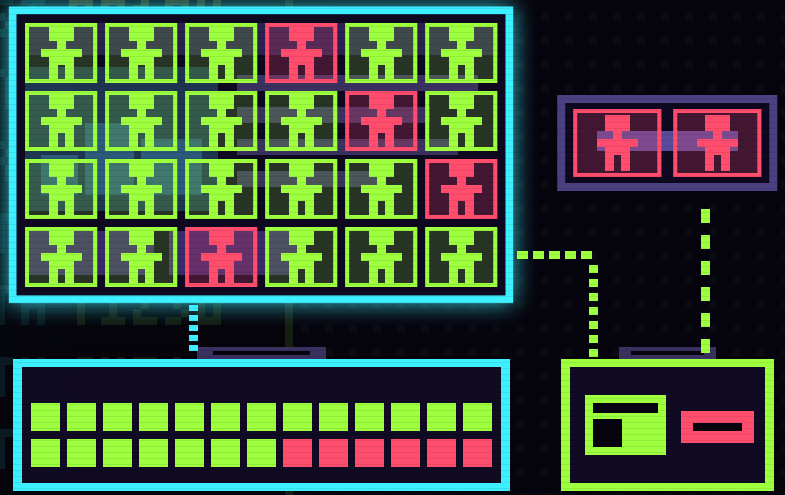
STAGE 7/8

LEVEL 7

МЕТРИКИ

READY?

Top-1: по компонентам или по окошкам?



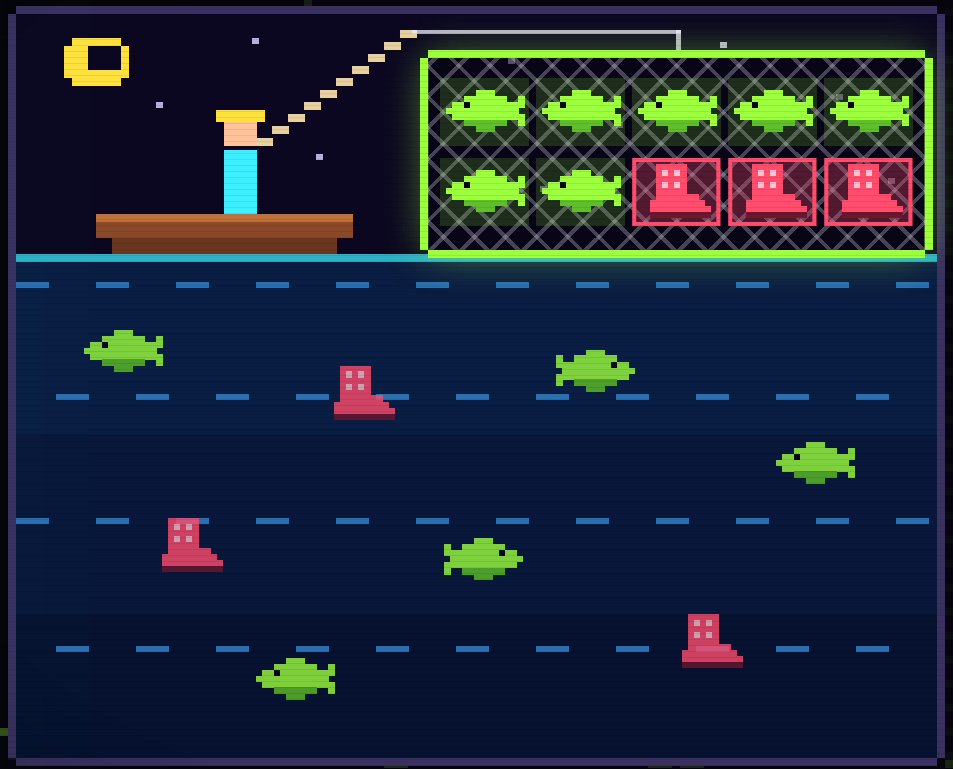
По компонентам: каждый сломанный компонент голосует один раз. **По окошкам**: голосует каждое окошко

Как выбирать старосту голосами классов или голосами всех учеников: если в одном классе 40 человек, а в другом 5, результаты будут разными

Ранний прогон на Vuetify: большая карточка дала **92 %** всех окошек

по окошкам	78 %
по компонентам	53 %

Что поймала наша сеть: precision

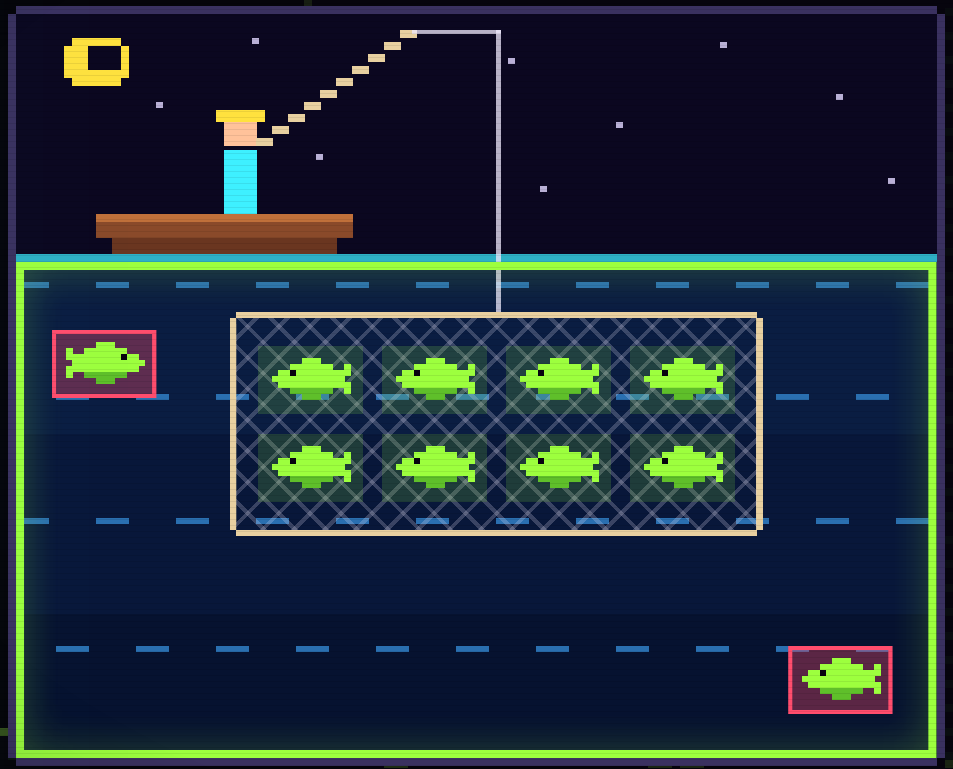


Нашу нейросеть можно мерить как рыбацкую: в озере рыба (настоящие поломки) и всякий мусор

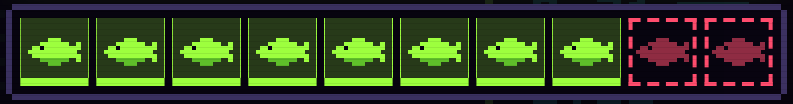
Precision: сколько в улове рыбы, а не старых ботинок. Вытащили 10 штук, рыб 7: precision = 0.7



Что поймала наша сеть: recall



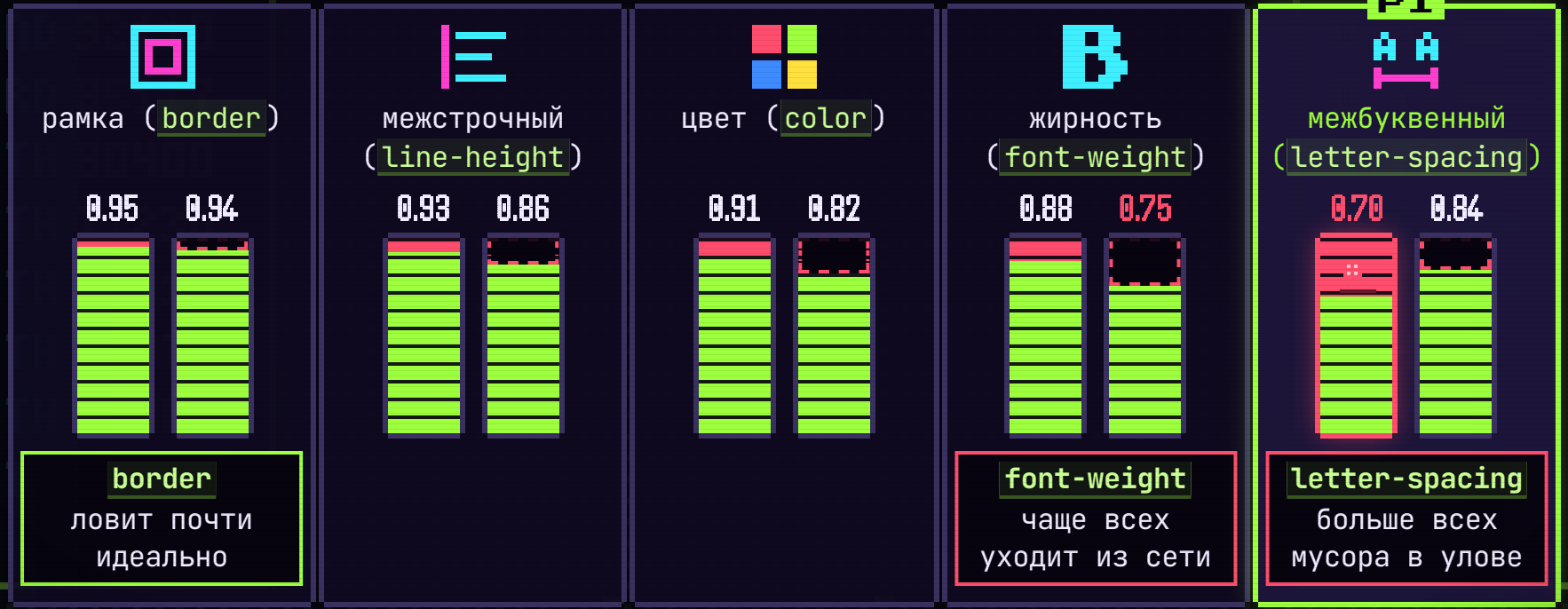
Recall: сколько рыбы из озера поймали.
Рыб было 10, в сеть попало 8: recall = **0.8**



PRECISION
0.58
0.44
0.31
0.12

Precision и recall по свойствам

свойство — **precision**: в улове рыба, а не мусор — **recall**: поймано рыбы из озера



Что в итоге

Задача решена: тест называет, что сломалось

```
color (0.94), font-weight (0.71)
```

Сеть работает: ~90 % попаданий, 2.6 мс на окошко, обучение за пару часов на ноутбуке. Всё на JavaScript

Это кроличья нора: безумно интересно и очень глубоко. Ныряйте!



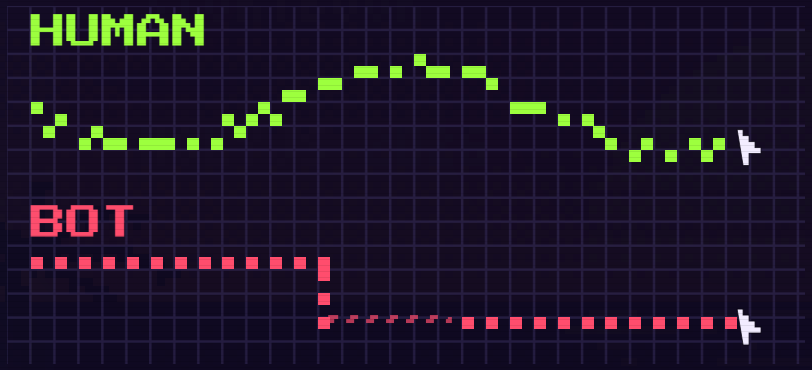
Ваш следующий уровень

SIDE QUESTS

ответы бесплатно → маленькая сеть → честная контрольная → порог по цене ошибки

★★☆ Человек или бот

≈ 1–2 выходных

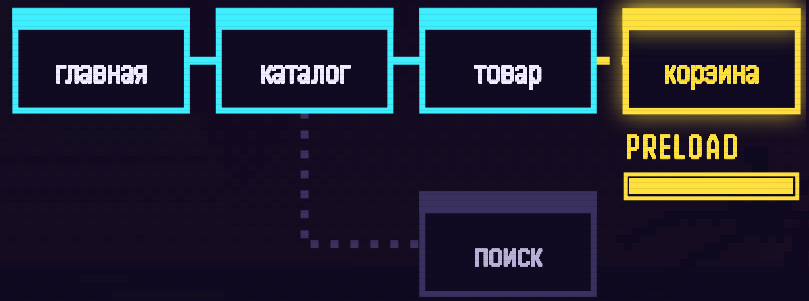


Живые траектории мыши — свои и коллег, ботов бесплатно рисует Playwright

Кучки делим по людям. Заблокировать живого дороже, чем пропустить бота

★★★ Следующий экран

≈ 2–4 недели



Ответы уже в аналитике: куда человек пошёл дальше. Угадали — подгружаем заранее

Учимся на прошлом месяце, проверяем на этом. Промах стоит трафика

LVL 8 · Финал ♥♥♥ SCORE 000040

Спасибо!

RESULTS

1	Баг в пикселях	CLEAR
2	Фабрика данных	CLEAR
3	Теория	CLEAR
4	Арсенал	CLEAR
5	Тренировка	CLEAR
6	Артефакт	CLEAR
7	Метрики	CLEAR
8	Финал	CLEAR

8/8 ★★★

BONUS CODE

Контакты — на vzhyx.io

PLAYWRIGHT

CONTINUE?

9

← SCAN

FREE PLAY

WORLD 0 — 1 — 2 — 3 — 4 — 5 — 6 — 7 — 8 ALL CLEAR