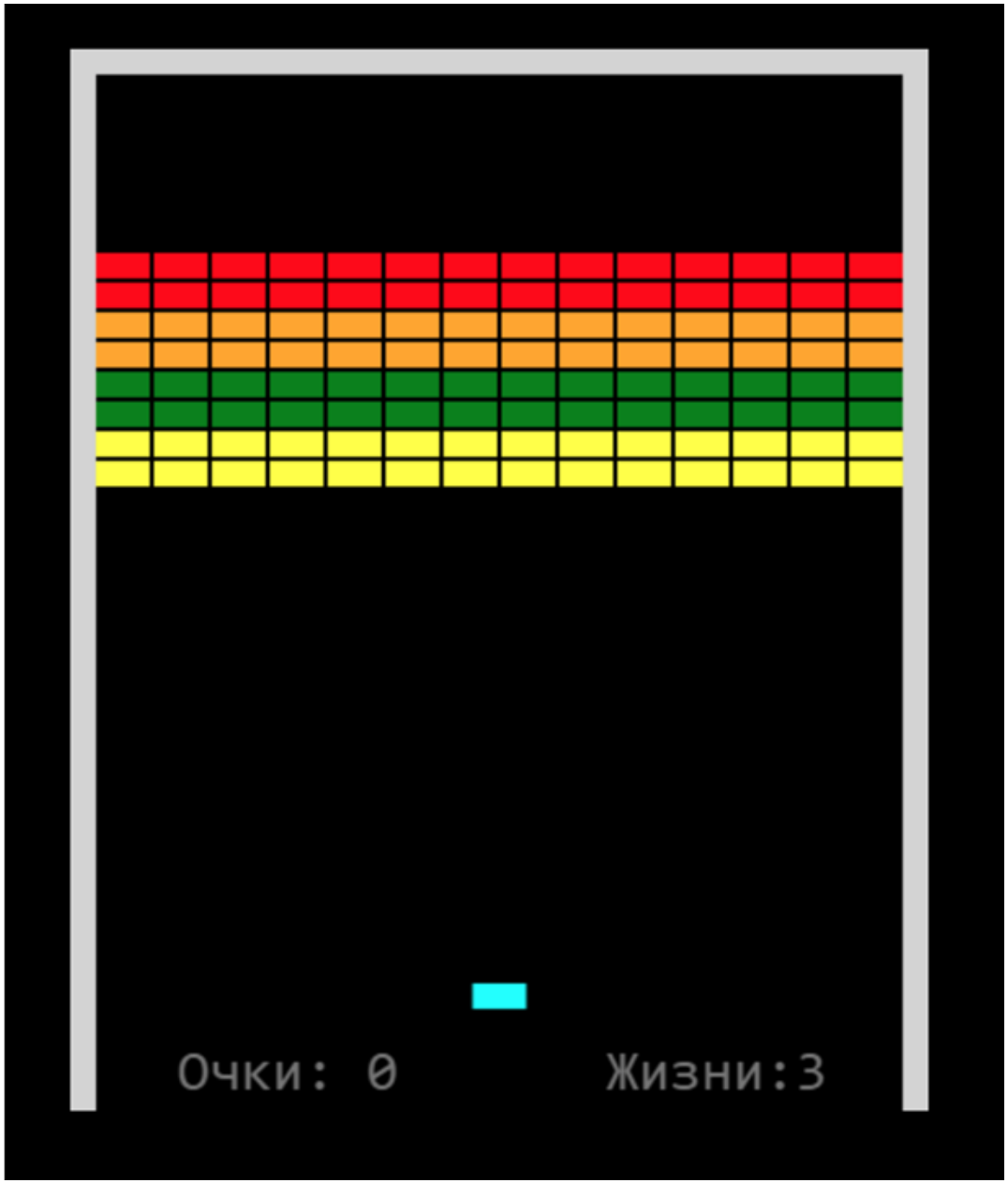
Улучшаем аркаду



Улучшаем арканоид

- Чтобы было интереснее играть, добавим в игру такое:
- подсчёт очков — за кирпич каждого цвета начисляется какое-то количество очков;
- ограниченное количество жизней;
- за каждые 25 набранных очков будем увеличивать платформу на 2 пикселя, чтобы игроку было проще играть дальше;
- за каждые 100 очков выдаём дополнительную жизнь;
- одновременно с увеличением жизни будем ускорять шарик — чтобы было сложнее играть;
- за каждое падение шарика жизнь отнимается;
- очки и жизни будем показывать на экране.

Нам понадобятся две новые переменные — для подсчёта очков и жизней, и две служебные:

```
// количество набранных очков за одну попытку
score = 0;

// количество жизней на старте
lives = 3;

// сколько очков нужно набрать до очередного увеличения платформы
score_paddle = 25;

// сколько очков нужно набрать до получения дополнительной жизни
score_lives = 100;
```

Начисляем очки за кирпичи

Так как у нас вся механика очков, бонусов и увеличения сложности завязана на сбивание кирпичей, то сделаем так:
Найдём в программе строчку, которая отвечает
за проверку касания кирпича шариком.
2.Добавим туда вызов функции touchdown().
3.Создадим выше эту функцию и наполним её нужными командами.
За проверку, коснулся ли шарик кирпича,
отвечает такой фрагмент кода:
if (collides(ball, brick)) {}

Добавим внутри этого условия вызов функции touchdown () и передадим в качестве параметра кирпич, которого коснулись. Так мы сможем понять, какого он был цвета, чтобы получить нужные очки:

```
touchdown(brick);
```

Теперь всё остальное, что связано с очками и бонусами, сделаем в этой новой функции. Создадим её сразу после раздела с переменными:

```
// в эту функцию мы поместим всё, что связано с касанием кирпичей
 function touchdown(t_brick) {

// начисляем очки в зависимости от цвета кирпича
        switch(t_brick.color) {
                case "yellow" : score += 1; break;
                case "green" : score += 2; break;
                case "orange" : score += 3; break;
                case "red" : score += 4;
}
```

```
        // за каждые 25 очков — увеличиваем размер платформы на
2 пикселя
if (score > score_paddle) {
  paddle.width += 2;
  // следующее увеличение — через 25 очков
  score_paddle += 25;
}
```

```
// а за каждые 100 очков в одной попытке — прибавляем ещё одну жизнь
if (score > score_lives){
  lives += 1;
  // и усложняем игру — увеличиваем скорость шарика
  ball.speed += 1;
  // сразу меняем скорость движения по осям
  if (ball.dx > 0) { ball.dx = ball.speed}
    else {ball.dx = -1 * ball.speed};
  if (ball.dy > 0) { ball.dy = ball.speed}
    else {ball.dy = -1 * ball.speed};
  // следующее увеличение жизни — через 100 очков
  score_lives += 100;
}
}
```

Выводим очки и жизни на экран

Сам код вывода очков и жизней поместим в самый конец нашего главного игрового цикла loop ():

```
// Цвет текста — серый
context.fillStyle = "#777777";
// Задаём размер и шрифт
context.font = "20pt monospace";
// Сначала выводим рекорд
context.fillText('Очки: ' + score, 50, 490);
// Затем — набранные очки
context.fillText('Жизни:'+ lives, 250, 490);
```

Обрабатываем падение шарика

Последнее, что нам осталось сделать, — убирать жизни при падении и остановить игру, когда они закончились.

Находим строчку, которая проверяет, упал ли шарик вниз:

```
// перезагружаем шарик, если он улетел вниз, за край игрового поля
if (ball.y > canvas.height) {

  // уменьшаем количество жизней
  lives -= 1;
```

```
 // обнуляем набранные очки
score = 0;
score_paddle = 25;
score_lives = 100;

// возвращаем прежний размер платформы
paddle.width = brickWidth;
```

```
 if (lives <= 0){

// рисуем чёрный прямоугольник посередине поля
    context.fillStyle = 'black';
    context.globalAlpha = 0.75;
    context.fillRect(0, canvas.height / 2 - 30, canvas.width, 60);

// пишем надпись белым моноширинным шрифтом по центру
    context.globalAlpha = 1;
    context.fillStyle = 'white';
    context.font = '36px monospace';
    context.textAlign = 'center';
    context.textBaseline = 'middle';
    context.fillText('GAME OVER!', canvas.width / 2, canvas.height / 2);
```

```
    // останавливаем игру
    return; };

  ball.x = 130;
  ball.y = 260;
  ball.dx = 0;
  ball.dy = 0;
}
```