

---

# Описание игры "Монополия"

## Table of Contents

Описание продукта .....	1
Игровая механика и скриншоты работы .....	1
Выводы .....	9

## Описание продукта

Монополия — это классическая игра, в которой вы можете покупать, арендовать и продавать свою собственность! В начале игры участники выставляют свои фишки на поле «Вперед», затем перемещают их по игровому полю в зависимости от выпавшего на кубиках количества очков.

Если вы попадаете на Участок Недвижимости, которая пока еще никому не принадлежит, то вы можете купить эту Недвижимость у Банка. Если вы решаете не покупать ее, она может быть продана на Аукционе другому игроку, предложившему за нее самую высокую цену. Игроки, имеющие Недвижимость, могут взимать арендную плату с игроков, которые попадают на их Участок. При строительстве Домов и Гостиниц арендная плата значительно возрастает, поэтому вам следует вести строительство на как можно большем количестве Участков.

Если вы нуждаетесь в деньгах, вы можете заложить вашу Недвижимость. Но не расслабляйтесь — в некоторых случаях вас могут посадить в Тюрьму.

**Table 1. Функциональность:**

Бросок игровых костей	Симуляция броска игровых костей позволяет игроку совершить ход.
Имущество игрока	Владение недвижимостью дает игроку право взимать арендную плату с любых арендаторов, которые остановились на поле, обозначающем ее. Очень выгодно владеть недвижимостью всей цветовой группы — иными словами, владеть монополией.
Управление имуществом и балансом	На каждый ход игрок может заложить своё имущество, чтобы получить деньги для выкупа себя из тюрьмы или приобретения другого имущества.
Колесо фортуны	Иногда удача может улыбнуться игроку после выхода из тюрьмы и перенести его в очень выгодное положение.

## Игровая механика и скриншоты работы

Вся логика продукта реализована в файле `monopoly.js` и `classicedition.js`, а также использовалась библиотека `jquery-1.11.1.min.js`.

Далее представлен код функции, которая обновляет данные игровых костей во время хода:

```
/**
 * Работа со свойством
 *
 * Функция для обновления цифр на
```

```
* игральной кости
*
*/
function updateDice() {
    let die0 = game.getDie(true);
    let die1 = game.getDie(false);

    $("#die0").show();
    $("#die1").show();

    if (document.images) {
        let element0 = document.getElementById("die0");
        let element1 = document.getElementById("die1");

        element0.classList.remove("die-no-img");
        element1.classList.remove("die-no-img");

        element0.title = "Die (" + die0 + " spots)";
        element1.title = "Die (" + die1 + " spots)";

        if (element0.firstChild) {
            element0 = element0.firstChild
        } else {
            element0 = element0.appendChild(document.createElement("img"))
        }

        element0.src = "images/Die_" + die0 + ".png";
        element0.alt = die0;

        if (element1.firstChild) {
            element1 = element1.firstChild
        } else {
            element1 = element1.appendChild(document.createElement("img"))
        }

        element1.src = "images/Die_" + die1 + ".png";
        element1.alt = die1
    } else {
        document.getElementById("die0").textContent = die0;
        document.getElementById("die1").textContent = die1;

        document.getElementById("die0").title = "Die";
        document.getElementById("die1").title = "Die"
    }
}
```

Код, который обновляет информацию об имуществе игрока

```
/**
 * Работа со свойством
 *
 * Функция для обновления владений игрока
 *
 */
function updateOwned() {
    let p = player[turn];
    let checkedProperty = getCheckedProperty();
```

```

$("#option").show();
$("#owned").show();

let html = "",
    firstProperty = -1;

let mortgageText = "",
    houseText = "";
let sq;

for (let i = 0; i < 40; i++) {
    sq = square[i];
    if (sq.groupNumber && sq.owner === 0) {
        $("#cell" + i + "owner").hide()
    } else if (sq.groupNumber && sq.owner > 0) {
        let currentCellOwner = document.getElementById("cell" + i + "owner")

        currentCellOwner.style.display = "block";
        currentCellOwner.style.backgroundColor = player[sq.owner].color;
        currentCellOwner.title = player[sq.owner].name
    }
}

for (let i = 0; i < 40; i++) {
    sq = square[i];
    if (sq.owner === turn) {

        mortgageText = "";
        if (sq.mortgage) {
            mortgageText = "title='Mortgaged' style='color: grey'"
        }

        houseText = "";
        if (sq.house >= 1 && sq.house <= 4) {
            for (let x = 1; x <= sq.house; x++) {
                houseText += "<img src='images/house.png' alt='' title='House'"
            }
        } else if (sq.hotel) {
            houseText += "<img src='images/hotel.png' alt='' title='Hotel'"
        }

        if (html === "") {
            html += "<table>";
            firstProperty = i
        }

        html += "<tr class='property-cell-row'><td class='propertycellcheckk"

        if (sq.groupNumber === 1 || sq.groupNumber === 2) {
            html += " border: 1px solid grey width: 18px"
        }

        html += "' onmouseover='showdeed(" + i + ")' onmouseout='hidedeed("
    }
}

if (p.communityChestJailCard) {
    if (html === "") {

```

```

        firstProperty = 40;
        html += "<table>"
    }
    html += "<tr class='property-cell-row'><td class='propertycellcheckbox'>"
}
if (p.chanceJailCard) {
    if (html === "") {
        firstProperty = 41;
        html += "<table>"
    }
    html += "<tr class='property-cell-row'><td class='propertycellcheckbox'>"
}

if (html === "") {
    html = p.name + ", you don't have any properties.";
    $("#option").hide()
} else {
    html += "</table>"
}

document.getElementById("owned").innerHTML = html;

// Select previously selected property.
if (checkedProperty > -1 && document.getElementById("propertycheckbox" + checkedProperty).checked) {
    document.getElementById("propertycheckbox" + checkedProperty).checked = false;
} else if (firstProperty > -1) {
    document.getElementById("propertycheckbox" + firstProperty).checked = true;
}
$(".property-cell-row").click(function () {
    let row = this;

    // Toggle check the current checkbox.
    $(this).find(".propertycellcheckbox > input").prop("checked", function () {
        return !val
    });

    // Set all other checkboxes to false.
    $(".propertycellcheckbox > input").prop("checked", function (index, val) {
        if (!$.contains(row, this)) {
            return false
        }
    });

    updateOption()
});
updateOption()
}

```

Пример кода, который реализует управление владениями и балансом игрока

```

/**
 * Работа со свойством
 *
 * Функция для обновления настроек игрока
 *
 */
function updateOption() {

```

```
$("#option").show();

let allGroupUnimproved = true;
let allGroupUnmortgaged = true;
let checkedProperty = getCheckedProperty();

if (checkedProperty < 0 || checkedProperty >= 40) {
    $("#buyHouseButton").hide();
    $("#sellHouseButton").hide();
    $("#mortgagebutton").hide();

    let houseSum = 32;
    let hotelSum = 12;

    for (let i = 0; i < 40; i++) {
        s = square[i];
        if (s.hotel === 1)
            hotelSum--;
        else
            houseSum -= s.house
    }

    $("#buildings").show();
    document.getElementById("buildings").innerHTML = "<img src='images/house

    return
}

$("#buildings").hide();
let sq = square[checkedProperty];

buyHouseButton = document.getElementById("buyHouseButton");
sellHouseButton = document.getElementById("sellHouseButton");

$("#mortgagebutton").show();
document.getElementById("mortgagebutton").disabled = false;

if (sq.mortgage) {
    document.getElementById("mortgagebutton").value = "Unmortgage ($" + Matl
    document.getElementById("mortgagebutton").title = "Unmortgage " + sq.name
    $("#buyHouseButton").hide();
    $("#sellHouseButton").hide();

    allGroupUnmortgaged = false
} else {
    document.getElementById("mortgagebutton").value = "Mortgage ($" + (sq.p
    document.getElementById("mortgagebutton").title = "Mortgage " + sq.name

    if (sq.groupNumber >= 3) {
        $("#buyHouseButton").show();
        $("#sellHouseButton").show();
        buyHouseButton.disabled = false;
        sellHouseButton.disabled = false;

        buyHouseButton.value = "Buy house ($" + sq.housePrice + ")";
        sellHouseButton.value = "Sell house ($" + (sq.housePrice * 0.5) + "
        buyHouseButton.title = "Buy a house for $" + sq.housePrice;
```

```
sellHouseButton.title = "Sell a house for $" + (sq.housePrice * 0.5)

if (sq.house === 4) {
  buyHouseButton.value = "Buy hotel ($" + sq.housePrice + ")";
  buyHouseButton.title = "Buy a hotel for $" + sq.housePrice
}
if (sq.hotel === 1) {
  $("#buyHouseButton").hide();
  sellHouseButton.value = "Sell hotel ($" + (sq.housePrice * 0.5)
  sellHouseButton.title = "Sell a hotel for $" + (sq.housePrice *

}

let maxHouse = 0;
let minhouse = 5;

for (let j = 0; j < max; j++) {

  if (square[currentSquare.group[j]].house > 0) {
    allGroupUnimproved = false;
    break
  }
}

let max = sq.group.length;
for (let i = 0; i < max; i++) {
  s = square[sq.group[i]];

  if (s.owner !== sq.owner) {
    buyHouseButton.disabled = true;
    sellHouseButton.disabled = true;
    buyHouseButton.title = "Before you can buy a house, you must
  } else {

    if (s.house > maxHouse) {
      maxHouse = s.house
    }

    if (s.house < minhouse) {
      minhouse = s.house
    }

    if (s.house > 0) {
      allGroupUnimproved = false
    }

    if (s.mortgage) {
      allGroupUnmortgaged = false
    }
  }
}

if (!allGroupUnmortgaged) {
  buyHouseButton.disabled = true;
  buyHouseButton.title = "Before you can buy a house, you must un

}

// Force even building
if (sq.house > minhouse) {
```

```
        buyHouseButton.disabled = true;

        if (sq.house === 1) {
            buyHouseButton.title = "Before you can buy another house, the other player must first buy a house."
        } else if (sq.house === 4) {
            buyHouseButton.title = "Before you can buy a hotel, the other player must first buy a house."
        } else {
            buyHouseButton.title = "Before you can buy a house, the other player must first buy a house."
        }
    }
    if (sq.house < maxHouse) {
        sellHouseButton.disabled = true;

        if (sq.house === 1) {
            sellHouseButton.title = "Before you can sell house, the other player must first buy a house."
        } else {
            sellHouseButton.title = "Before you can sell a house, the other player must first buy a house."
        }
    }

    if (sq.house === 0 && sq.hotel === 0) {
        $("#sellHouseButton").hide()

    } else {
        $("#mortgagebutton").hide()

    }

    // Before a property can be mortgaged or sold, all the properties owned by the player must be improved.
    if (!allGroupUnimproved) {
        document.getElementById("mortgagebutton").title = "Before a property can be mortgaged or sold, all the properties owned by the player must be improved."
        document.getElementById("mortgagebutton").disabled = true
    }

    } else {
        $("#buyHouseButton").hide();
        $("#sellHouseButton").hide()
    }
}
}
```

Функция, которая рассчитывает шанс на везение после выхода из тюрьмы

```
/**
 * Работа со свойством
 *
 * Функция, которая даёт шанс игроку
 * попасть на поле Общественного фонда
 */
function chanceCommunityChest() {
    let p = player[turn];

    // Community Chest
    if (p.position === 2 || p.position === 17 || p.position === 33) {
        let communityChestIndex = communityChestCards.deck[communityChestCards.deck.length - 1];

        // Remove the get out of jail free card from the deck.
    }
}
```

```

    if (communityChestIndex === 0) {
        communityChestCards.deck.splice(communityChestCards.index, 1)
    }

    popup("<img src='images/community_chest_icon.png' style='height: 50px w
        communityChestAction(communityChestIndex)
    });

    communityChestCards.index++;

    if (communityChestCards.index >= communityChestCards.deck.length) {
        communityChestCards.index = 0
    }

    // Chance
} else if (p.position === 7 || p.position === 22 || p.position === 36) {
    let chanceIndex = chanceCards.deck[chanceCards.index];

    // Remove the get out of jail free card from the deck.
    if (chanceIndex === 0) {
        chanceCards.deck.splice(chanceCards.index, 1)
    }

    popup("<img src='images/chance_icon.png' style='height: 50px width: 26p
        chanceAction(chanceIndex)
    });

    chanceCards.index++;

    if (chanceCards.index >= chanceCards.deck.length) {
        chanceCards.index = 0
    }
} else {
    if (!p.human) {
        p.AI.alertList = "";

        if (!p.AI.onLand()) {
            game.next()
        }
    }
}
}
}

```

Ниже

представлен

скриншот

старта

игры

Выберите количество игроков





Далее

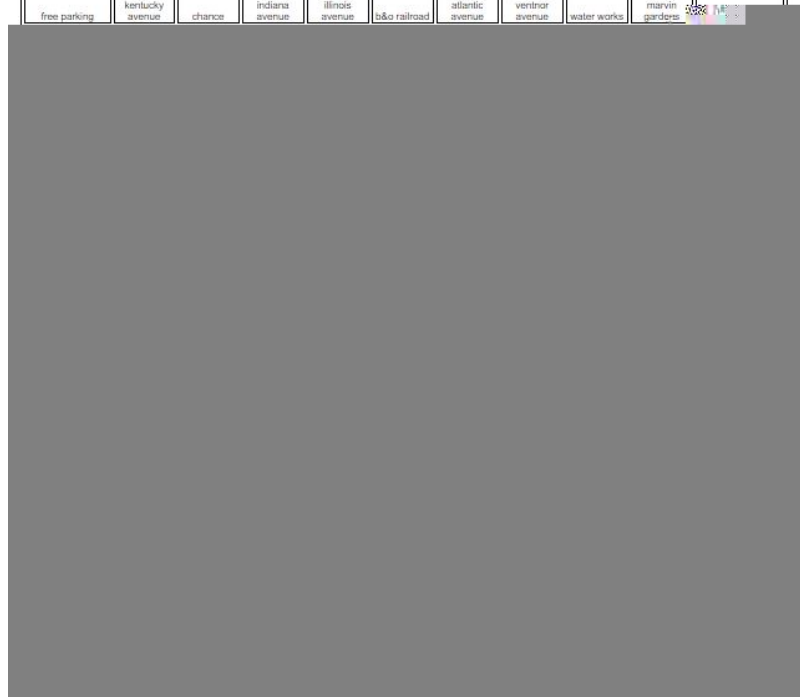
представлен

скриншот

карты

монополии

free parking	kentucky avenue	chance	indiana avenue	illinois avenue	b&o railroad	atlantic avenue	ventnor avenue	water works	marvin gardens	go to jail
--------------	--------------------	--------	-------------------	--------------------	--------------	--------------------	-------------------	-------------	-------------------	------------



## Выводы

В течение двух месяцев была разработана игра "Монополия" с использованием JavaScript, CSS, HTML. Игра получилась забавной, местами непредсказуемой.