

JavaScript Turtle Graphics

Середовище для програмування Черепашачої графіки

Вступ

JavaScript Turtle Graphics - це навчальне середовище для програмування, яке містить базовий набір інструментів для створення Черепашачої графіки (Turtle Graphics), використовуючи об'єктно-орієнтований підхід.

Мета створення середовища JavaScript Turtle Graphics - знайомство з основами програмування на прикладі роботи з Черепашачою графікою у вебпереглядачі, використовуючи мову програмування JavaScript і можливості бібліотеки [p5.js](#).

Підключення

Середовище JavaScript Turtle Graphics можна вважати бібліотекою, яка розширює [p5.js](#).

Щоб використати JavaScript Turtle Graphics у своєму проєкті, слідуйте одному із варіантів:

- у разі використання онлайн-редактора [editor.p5js.org](#) файл бібліотеки JavaScript Turtle Graphics необхідно підключити у файлі `index.html` перед файлом ескізу `sketch.js` ;
- у разі локальної розробки [завантажити](#) зразок проєкту `p5.js` , у який підключити файл бібліотеки JavaScript Turtle Graphics ;
- використати [p5js-widget.pp.ua](#) - спрощену версію онлайн-редактора [editor.p5js.org](#) для мобільних пристроїв.

Завантажити файл бібліотеки JavaScript Turtle Graphics можна у розділі [Початковий код](#).

Огляд можливостей

Початкові налаштування

У блоці `setup()` записуємо функцію `createCanvas(windowWidth, windowHeight)` для створення полотна, де `windowWidth` і `windowHeight` - це ширина і висота вікна полотна відповідно, і викликаємо `myCode()` - це назва блоку, в якому записані інструкції для Черепашки.

```
function setup() {  
  createCanvas(windowWidth, windowHeight);  
  myCode();  
}
```

Блок `myCode()` має такий вигляд:

```
function myCode(){  
  // інструкції  
}
```

Усі інструкції для керування Черепашкою записуємо у блоці `myCode()` . Назву цього блоку можна змінити на свій задум.

Завжди першою інструкцією у блоці `myCode()` є інструкція зі створення об'єкта Черепашки з певним ім'ям:

```
function myCode(){
  t = new Turtle();
}
```

Тепер Черепашка має ім'я `t`.

Ім'я для Черепашки можна обрати на свій задум.

Усі наступні інструкції необхідно записувати у форматі `t.інструкція`.

Мапа

Якщо необхідно створити мапу (координатну сітку) на полотні, додаємо у `myCode()` інструкцію `grid()`:

```
function myCode(){
  t = new Turtle();
  t.grid();
}
```

Інструкція `grid()` використовує три параметри із такими значеннями за замовчуванням:

- крок сітки - `50` пікселів,
- колір сітки - [Platinum](#),
- колір осей - [Cerulean](#).

Значення кольору записується як рядок в одному із форматів: "red" (назва), "#fdfd90" (шістнадцяткове значення), "rgb(11, 156, 78)" (значення червоної, зеленої, синьої складових), "rgba(45, 145, 67, 0.5)" (значення червоної, зеленої, синьої складових і прозорості).

Значення за замовчуванням використовуються тоді, коли `grid()` викликається без аргументів, як у прикладі вище. За потреби `grid()` можна викликати із користувацькими значеннями, зазначивши їх у дужках у вказаному порядку.

Використовуючи інструкцію `setStepGrid(step)` перед малюванням сітки можна окремо встановити крок сітки `step`, а за допомогою інструкції `getStepGrid()` можна отримати поточне значення кроку сітки.

Черепашка і `p5.js`

На полотні поруч з Черепашкою можна створювати зображення, використовуючи засоби бібліотеки `p5.js`. У цьому разі код записуємо у блоці `myDraw()` (за потреби назву блоку можна змінити на іншу) між коментарями `// початок коду` і `// кінець коду`. Також назву блоку `myDraw()` необхідно записати у блоці `draw()`, як показано нижче:

```
function draw() {
  myDraw();
}

function myDraw(){
  push();
  translate(width / 2, height / 2);
```

```
scale(1, -1);  
// початок коду  
  
// кінець коду  
pop();  
}
```

Відображення Черепашки, інформаційна панель та компас

За замовчуванням на полотні відображається інформаційна панель з даними про:

- стан Черепашки (кутову орієнтацію, поточні координати),
- стан олівця (на полотні чи піднятий),
- координати вказівника миші.

а у правому нижньому куті полотна увімкнений компас, який вказує напрямок і значення кутової орієнтації Черепашки.

За відображення вищезгаданих елементів інтерфейсу відповідають інструкції, які записуються у блоці `draw()`

```
function draw() {  
  t.place();  
  t.compass();  
  t.dashboard();  
}
```

і використовуються для таких цілей:

- `place()` - показати Черепашку на полотні у її поточних координатах. Цю інструкцію рекомендується завжди використовувати.
- `compass()` - показати компас.
- `dashboard()` - показати інформаційну панель.

Також на екрані можна відобразити ім'я розробника: у блоці `draw()` розмістити інструкцію `t.creator()`.

Зміна розмірів вікна вебпереглядача/полотна

При зміні розмірів вікна вебпереглядача/полотна усі елементи інтерфейсу (мапа, інформаційна панель, компас тощо) залишаються на своїх місцях і з'являються смуги прокручування. Щоб ці елементи налаштувалися на нові розміри, оновіть вебсторінку. За потреби перед цим збережіть свій код.

Якщо ви працюєте на вебсторінці інтерактивної мапи (p5js-widget.pp.ua/map/), спочатку встановіть розміри вікна вебпереглядача, а потім оновіть вебсторінку, щоб елементи інтерфейсу налаштувались під ці розміри.

Інструкції для Черепашки, згруповані за категоріями

Рух

Рух Черепашки відбувається **без анімації**, тобто виконуються усі записані інструкції і Черепашка відразу розташовується на полотні у точці з кінцевими координатами.

Після запуску Черепашка з'являється на полотні:

- у точці з координатами `[0, 0]` (центр полотна),
- "дивиться" праворуч,
- має початкову кутову орієнтацію, що вимірюється у градусах, `0°`.

Інструкція	Опис
<code>forward(distance)</code>	Перемістити Черепашку вперед на відстань <code>distance</code> , у напрямку, куди "дивиться" Черепашка.
<code>back(distance)</code>	Перемістити Черепашку назад на відстань <code>distance</code> , у протилежний бік напрямку її руху.
<code>goto(x, y)</code>	Перемістити Черепашку в точку з координатами <code>(x, y)</code> .
<code>home()</code>	Перемістити Черепашку в точку з координатами <code>(0, 0)</code> (центр полотна) і встановити початкову кутову орієнтацію.
<code>left(angle)</code>	Повернути Черепашку ліворуч на кут <code>angle</code> , де <code>angle</code> - число (ціле чи з плаваючою крапкою). Якщо значення кута <code>angle</code> додатне, то Черепашка повертається на це значення кута проти годинникової стрілки (кутова орієнтація стає <code>angle</code>), якщо від'ємне - Черепашка повертається за годинниковою стрілкою на це значення кута (кутова орієнтація стає <code>360 - abs(angle)</code>).
<code>right(angle)</code>	Повернути Черепашку праворуч на кут <code>angle</code> , де <code>angle</code> - число (ціле чи з плаваючою крапкою). Якщо значення кута <code>angle</code> від'ємне, то Черепашка повертається на це значення кута проти годинникової стрілки (кутова орієнтація стає <code>angle</code>), якщо додатне - Черепашка повертається за годинниковою стрілкою на це значення кута (кутова орієнтація стає <code>360 - abs(angle)</code>).
<code>setHeading(angle)</code>	Встановити кутову орієнтацію Черепашки на кут <code>angle</code> , де <code>angle</code> - число (ціле чи з плаваючою крапкою). Якщо значення кута <code>angle</code> додатне, то Черепашка повертається на це значення кута проти годинникової стрілки (кутова орієнтація стає <code>angle</code>), якщо від'ємне - Черепашка повертається за годинниковою стрілкою на це значення кута (кутова орієнтація стає <code>360 - abs(angle)</code>).

Черепашка

Інструкція	Опис
<code>getPosition()</code>	Отримати поточні координати Черепашки у вигляді списку <code>[x, y]</code> .
<code>getHeading()</code>	Отримати поточну кутову орієнтацію Черепашки.
<code>setShape(shape)</code>	Встановити форму <code>shape</code> для Черепашки. <code>shape</code> може набувати значень: <code>"blank"</code> (невидимість), <code>"circle"</code> , <code>"square"</code> , <code>"triangle"</code> . За замовчуванням форма Черепашки <code>"triangle"</code> .
<code>setShapeSize(s)</code>	Встановити розмір <code>s</code> для форми Черепашки. За замовчуванням - мінімальний розмір <code>1</code> , максимальне значення <code>20</code> .
<code>showTurtle()</code>	Зробити Черепашку видимою.
<code>hideTurtle()</code>	Зробити Черепашку невидимою.
<code>clr()</code>	Очистити полотно. Черепашка залишається у поточній точці, її

	кутова орієнтація зберігається.
--	---------------------------------

Олівець

Якщо олівець на полотні, при переміщенні Черепашка малюватиме лінію.

Інструкція	Опис
penUp()	Підняти олівець.
penDown()	Опустити олівець на полотно. За замовчуванням - олівець на полотні.
setPenSize(s)	Встановити у пікселях товщину лінії олівця на s. За замовчуванням товщина лінії олівця 1 піксель.

Колір

Значення кольору записується як рядок в одному із форматів: "red" (назва), "#fdfd90" (шістнадцяткове значення), "rgb(11, 156, 78)" (значення червоної, зеленої, синьої складових), "rgba(45, 145, 67, 0.5)" (значення червоної, зеленої, синьої складових і прозорості).

Інструкція	Опис
setColor(c, f)	Встановити колір c для олівця та колір заливки f для зафарбовування фігур. За замовчуванням колір олівця - чорний, колір заливки - прозорий. Для встановлення лише кольору олівця інструкція використовується з одним параметром: setColor(c) .
setBgColor(c)	Встановити колір c для тла полотна. За замовчуванням - білий.
setFillColor(f)	Встановити колір заливки c для зафарбовування фігур. За замовчуванням - заливка прозора.
beginFill()	Увімкнути зафарбовування фігури поточним кольором заливки.
endFill()	Вимкнути зафарбовування фігури поточним кольором заливки.

Фігури

Інструкція	Опис
oval(r, e)	Намалювати коло чи еліпс. Якщо параметри r і e набувають однакових значень, то Черепашка малює коло зі значенням радіуса, що дорівнює r. У разі різних значень r і e - отримуємо еліпс.
polygon([[x1, y1], [x2, y2], [x3, y3], ...])	Намалювати багатокутник за координатами вершин x1, y1, x2, y2, x3, y3, ... (за годинниковою стрілкою).

Текст

Інструкція	Опис
write("txt", {horizontal: A, vertical: B})	Написати текст "txt" у поточній позиції Черепашки. Об'єкт {horizontal: A, vertical: B} використовується для вирівнювання тексту, де A може

A, vertical: B}, {font: C, size: D, style: E})	набувати значень LEFT, CENTER або RIGHT, а в - TOP, BOTTOM, CENTER або BASELINE. Об'єкт {font: C, size: D, style: E} визначає шрифт, розмір тексту і стиль відповідно, а саме: c може набувати значень Arial, Times, Verdana тощо, D - ціле число, E може набувати значень NORMAL, ITALIC, BOLD, BOLDITALIC.
--	--

За замовчуванням текст є порожнім рядком і має такі параметри: вирівнювання {horizontal: CENTER, vertical: CENTER}, шрифту {font: "sans-serif", size: 12, style: NORMAL}.

Розробка

Для розробки середовища JavaScript Turtle Graphics застосовувалась мова програмування JavaScript та інструменти бібліотеки [p5.js](#).

Початковий код

Початковий код JavaScript Turtle Graphics зберігається в єдиному файлі, який можна завантажити [ТУТ](#).

Ліцензія

Середовище JavaScript Turtle Graphics поширюється вільно і безкоштовно.