

РОЗРОБКА WEB-ДОДАТКІВ НА PHASER

Phaser – це фреймворк, який дозволяє дуже швидко створювати ігри. З його допомогою створити гру дійсно легко і швидко. Ми не відволікаємося на Астор'ів, рендеринг, фізику - фокусуємося на ігровому процесі. За допомогою цього фреймворку можна розробити стильні, модні і молодіжні HTML5 додатки. На відміну від більшості інших фреймворків від самого початку був спроектований для роботи з мобільними пристроями.

Фреймворк Phaser – один з найшвидших движків, який рендерить за допомогою WebGL. А у випадку, якщо WebGL не підтримується — на Canvas.

Основні переваги Phaser:

- Великий FPS.
- Кросбраузерність, підтримка мобільних пристроїв.
- Можливість рендера на Canvas или WebGL.
- Підтримка плагінів.
- Вбудована підтримка роботи зі звуком.
- Підтримка тайлів, анімацій і спрайтів.
- Вбудована фізика, підтримка колізій (зіткнення об'єктів).
- Гнучкість – будь-яка деталь може бути змінена.
- Простота (наприклад, ми можемо переміщувати гравця по карті простими функціями типу moveToXY(), moveUp() і т.д.).

Приводяться платформи, на яких буде працювати фреймворк. Настільні браузері:

Phaser - це бібліотека JavaScript, призначена для роботи на всіх основних настільних браузерах. Це такі браузері:

- Internet Explorer 9 і вище.
- Firefox.
- Chrome.
- Safari.

IE9 не підтримує Canvas і WebGL, а для Phaser потрібна система, яка зможе обробляти DOM рендеринг.

Мобільні браузері:

На мобільному телефоні, Phaser працює на iOS5 і вище, також на Mobile Safari. На Android, він працює починаючи з версії 2.2 і браузері Chrome. Chrome тепер є стандартний Android-браузер, з останніми версіями він навіть почав підтримувати WebGL, а для застарілих пристроїв Android, використовується Canvas.

Ключові особливості.

Простота JavaScript.

Так як в JavaScript не має ООП, масивів ланцюгів, успадкування або компонентів системи, і ви не повинні примушувати себе створювати свої об'єкти в структурі класів. Це не означає, що ви не можете побудувати свою гру в структурованому вигляді. Все це означає, що Phaser не примушує це використовувати.

Дуже легке завантаження Asset файлів.

Phaser має вбудований завантажувач assets, який може обробляти:

- Зображення.
- Текстури атласів (включи текстури Packer, JSON Hash, JSON Array, Flash CS6/CC, і Starling XML формати).
- Звукові файли.
- Файли даних (XML, JSON, звичайний текст).
- Tilemaps (CSV і Tiled map формати).
- Піксельні шрифти.

Assets можуть також бути частково завантажені, кешуватися і витягуватися з різних URL-адрес (для підтримки CDN), і ви навіть можете перетворити будь-який спрайт в прогрес-бар з одного рядка коду.

Рендеринг WebGL і Canvas.

Всередині, Phaser використовує Pixi.js для рендеринга. Pixi є великою, швидкою бібліотекою для рендеринга, яка фокусується на Canvas і WebGL.

Особливості Pixi.js:

- Візуалізації webgl (завдяки їй дуже велика продуктивність).
- Canvas візуалізація.
- Повний граф сцени.
- Супер легкий у використанні API.
- Підтримка атласів текстур.
- Автоматичне визначення, який засіб візуалізації слід використовувати.
- Повна підтримка функції миші та Multi-Touch.
- BitmapFont текст.
- Рендеринг текстур.
- Фільтри.

Для ігор це означає, якщо браузер не підтримує WebGL, то гра побудується на Canvas. WebGL створений на основі OpenGL ES 2.0 з підтримкою API для 3D-графіки. Він використовує елемент canvas з HTML5, а також

взаємодіє з DOM. Автоматичне управління пам'яттю відбувається завдяки мові JavaScript. Шейдери у WebGL запрограмовані безпосередньо на GLSL.

Для того, щоб гра автоматично налаштувалася під екран, потрібно написати такі налаштування для гри:

```
this.scale.scaleMode = Phaser.ScaleManager.SHOW_ALL;  
this.scale.pageAlignHorizontally = true;  
this.scale.pageAlignVertically = true;
```

Аудіо: Web Audio і Legacy Audio.

Phaser використовує Web Audio API, що дозволяє належним чином підтримувати аудіо вузли на основі, з декількома каналами, маршрутизації вузлів і всіх видів ефектів. Таким чином, Phaser повністю підтримує Web Audio. Проте, на Android особливо багато пристроїв, які до цих пір не підтримують його, тому потрібно завантажувати традиційно аудіо файли через спрайти. Але, так як не існує такого формату аудіо файлу, який грав би у всіх браузерах, для цього доречно завантажувати такі два типу аудіо файлів як ogg та mp3. Приклад коду як це виглядає:

```
this.load.audio('death', ['assets/audio/death.ogg', 'assets/audio/death.mp3']);
```

При такому завантаженні, гра сама буде вибирати такий формат, який браузер підтримує.

Вхідні дані: Multi-Touch, клавіатура, покажчик, миша.

За підтримки настільних і мобільних пристроїв є більший різноманітний ряд можливих варіантів введення. Phaser підтримує клавіатуру, мишу, сенсорне натиснення, MSPointer (тепер Pointer під IE11) і їх комбінації. Наприклад, на пристроях Windows, Surface можна переключатися між використанням миші і сенсорним натисненням, або використовувати обидва разом.

Для сенсорного введення, Phaser відмінно справляється як із одним натисненням, так і з декількома. Можемо задати до 10 точок дотику і відстежувати їх незалежно один від одного, використовуючи свої події для обробки Sprite взаємодії, такі як: перетягування, постукування, і зіткнення.

Приклад коду перетягування:

```
this.pet.inputEnabled = true;  
this.pet.input.enableDrag();
```

Фізика, Анімація, і Particles. В ядро Phaser входять бібліотеки ArcadePhysics і система ArcadeParticles. Це прості бібліотеки AABB, які дозволяють застосовувати силу тяжіння і рух до будь-якого Sprite, а потім перевірити на зіткнення і роз'єднання. Якщо така фізика вас не влаштовує, то можна з легкістю змінити її на іншу, наприклад на Box2D або p2.js. Система анімації також вбудована, що дозволяє легко анімувати об'єкти або властивості. Якщо гру зупинити на паузу, то відповідно всі анімації зроблять паузу автоматично і відновляться також.

Приклад коду анімації:

```
let petMovement = this.game.add.tween(this.pet);  
petMovement.to({x: x, y: y}, 700);
```

Цей код означає, що наш спрайт переміститься в координати x, y, за 700 мс.

Phaser також має величезний набір готових класів: SpriteAnimation, TileMap, Timer, GameState і багато іншого. В тому числі, і компоненти фізичного движка: RigidBody, Physics і т.п.

Наявність даних компонентів значно спрощує розробку.

Отже, Phaser - це добрий компроміс між повним набором можливостей і дружелюбністю до розробника. Фреймворк продовжує активно розвиватись і, якщо знаходяться помилки або недоробки, то вони оперативно виправляються.