

# H5动画与游戏开发



河北师范大学软件学院  
Software College of Hebei Normal University

# 课程范围概述

---

- H5 图形图像及动画基础 ( 15% )
  - Canvas绘图及相关API、Canvas图像操作及动画
- Cocos2d-JS ( 60% )
  - 节点、精灵、场景、层
  - 动作、动画、粒子、音效
  - 用户交互、数据存储、网络交互
- Cocos Creator ( 25% )
  - Cocos Creator工作流程 ( 资源工作流程、场景制作工作流程 )
  - 脚本开发、预制体、UI系统、物理系统、多平台发布

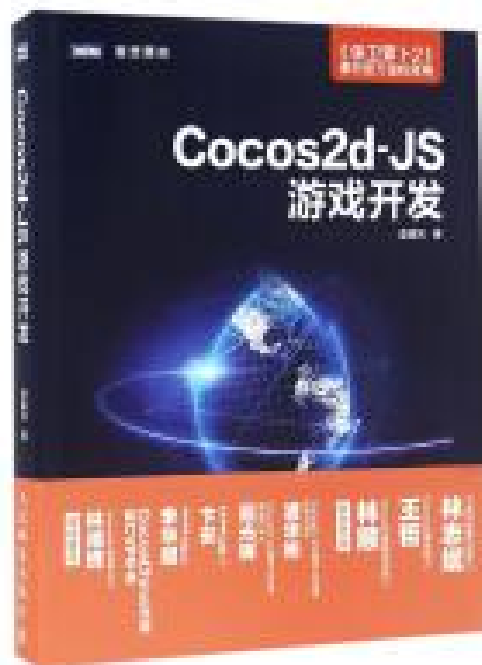


# 参考教材及学习工具

- 参考链接：

- <https://www.w3.org/TR/2dcontext/>
- <http://www.cocos.com/>
- <https://docs.cocos.com/creator/manual/zh/>

- 《HTML5 Canvas开发详解》
- 《Cocos2d-JS游戏开发》
- 《Cocos Creator游戏开发实战》
- Chrome、**VSCode**、WebStorm



# 课程价值

---

- 通过实践开发理解软件工程化思想
- 掌握游戏开发中的基本概念
- 掌握H5小游戏的基本开发能力
- 掌握微信小游的基本开发能力



## 课程考核及课程资料

---

- 平时表现（ 考勤、 学习状态、 个人Git学习仓库 ）
- 平时作业（ 在线任务、 课程实践 ）
- 期末考试

# H5 动画与游戏开发

---H5 图形图像简介

# 内容纲要

---

- **Canvas 概述**
- **SVG 概述**
- **WebGL 概述**



# Canvas 简介

- Canvas是H5出现的新标签，用于在Web端完成图形的绘制和动画的实现，它是图形动画的容器，与其他DOM元素节点类似（拥有对应的属性、方法）
- Canvas 提供了相应的API，通过JS脚本来完成相应的图形、图像及动画操作（用程序来绘画、图像处理和生成动画）
- Canvas是基于状态绘图，采用即时模式（immediate mode）进行位图的绘制和操作，能以.png 或.jpg 格式保存图像



# 内容纲要

---

- Canvas 概述
- **SVG 概述**
- WebGL 概述



# SVG ( Scalable Vector Graphics ) 概述

- SVG 是基于可扩展标记语言XML，用于描述二维矢量图形的一种图形格式（2003 年 1 月 14 日成为 W3C 推荐标准）
- SVG 基于 XML，这意味着 SVG DOM 中的每个元素都是可用的，可以为每个元素附加 JavaScript 事件处理器
- 在 SVG 中，每个被绘制的图形均被视为一个对象（SVG保留模式区别于Canvas的即时模式），如果 SVG 对象的属性发生变化，那么浏览器能够自动重现图形

# SVG 的优势

---

- SVG 图像可通过文本编辑器来创建和修改
- SVG 图像可被搜索、索引、脚本化或压缩
- SVG 是可伸缩的矢量图
- SVG 图像可在任何分辨率下被高质量打印
- SVG 图像文件尺寸更小（节省空间），可压缩性更强
- SVG 是开放的标准



# SVG 基本案例

- <https://developer.mozilla.org/zh-CN/docs/Web/SVG>

```
<body>
  <svg width="120" height="120" viewBox="0 0 120 120" version="1.1">
    <rect x="10" y="10" width="100" height="100" fill="blue"/>
    <circle cx="20" cy="20" r="20" fill="red" />
    <circle cx="100" cy="100" r="20" fill="red" />
    <circle cx="100" cy="20" r="20" fill="red" />
    <circle cx="20" cy="100" r="20" fill="red" />
    <ellipse cx="60" cy="60" rx="50" ry="25" />
  </svg>
</body>
```

# Canvas 与 SVG 对比

---

- Canvas

- 依赖分辨率（基于像素的**位图**）、事件模型/用户交互是粒度（x，y）
- 单个HTML元素，仅通过脚本进行修改，**即时模式（Immediate Mode）**
- 最适合图像密集型的游戏，其中的许多对象会被频繁重绘

- SVG

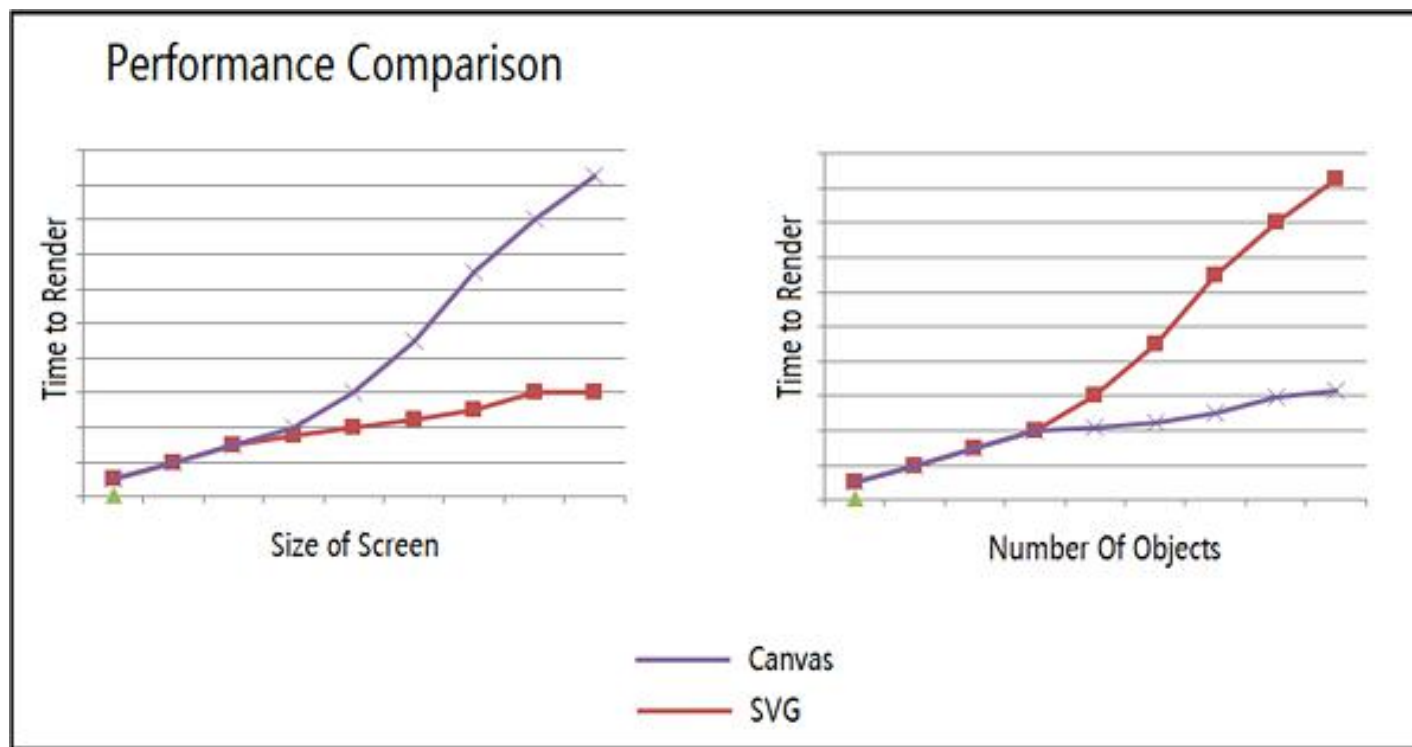
- 不依赖分辨率（基于形状的**矢量图**）、事件模型/用户交互被抽象（rect，path）
- 多个图形元素，成为DOM的一部分，可通过脚本和CSS修改，**保留模式（Retained Mode）**
- 复杂度高会减慢渲染速度（任何过度使用 DOM 的应用都不快）
- 不适合游戏应用



# Canvas 与 SVG 对比

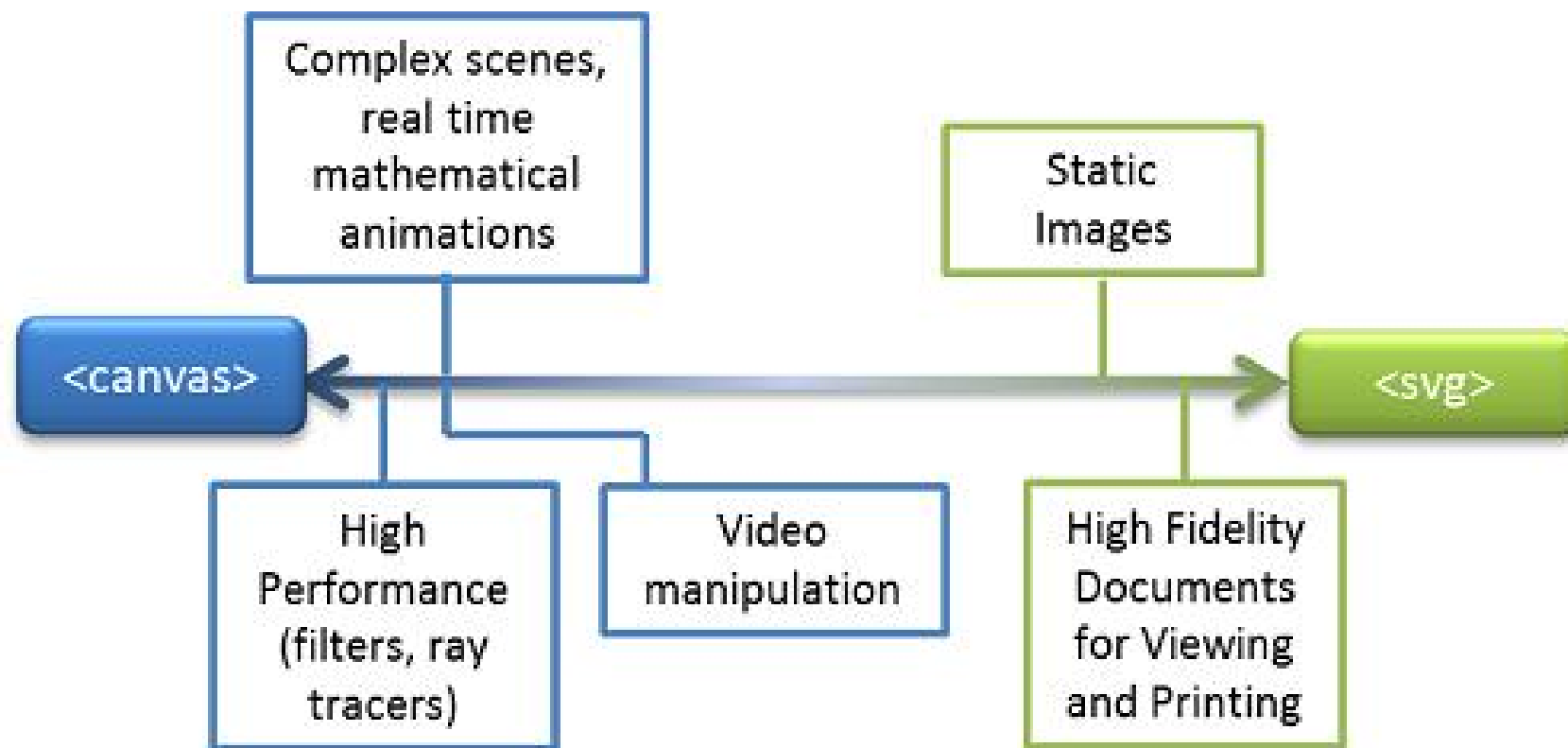
- 尺寸与数量对性能的影响

- 通常，随着屏幕尺寸的增加，Canvas效率开始降低，因为需要绘制更多的像素。随着屏幕上对象数量的增加，SVG效率开始降低，因为我们不断将它们添加到DOM



# Canvas 与 SVG 的选择

- 针对不同应用场景进行合理（Canvas、SVG、Canvas+SVG）



# 内容纲要

---

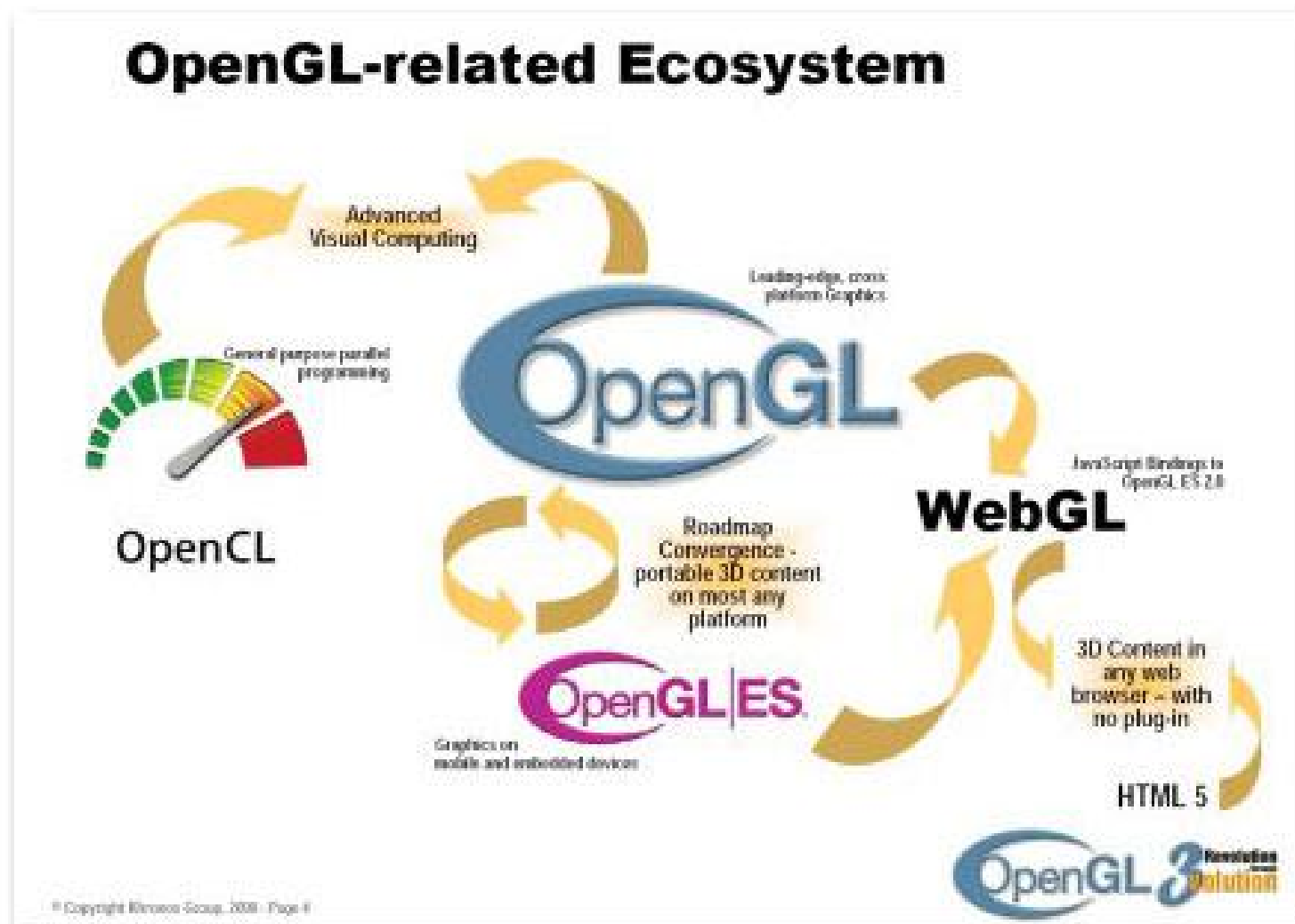
- Canvas 概述
- SVG 概述
- **WebGL 概述**



# WebGL 简介

- WebGL是一种3D绘图标准，这种绘图技术标准允许把JavaScript和OpenGL ES结合在一起，通过增加OpenGL ES的一个JavaScript绑定，WebGL即可以为HTML5 Canvas提供硬件3D加速渲染，这样Web开发人员就可以借助系统显卡来在浏览器里更流畅地展示3D场景和模型了，还能创建复杂的导航和数据视觉化。WebGL技术标准免去了开发网页专用渲染插件的麻烦，可被用于创建具有复杂3D结构的网站页面，甚至可以用来设计3D网页游戏等等
- WebGL完美地解决了现有的Web交互式三维动画的两个问题：第一，它通过HTML脚本本身实现Web交互式三维动画的制作，无需任何浏览器插件支持；第二，它利用底层的图形硬件加速功能进行的图形渲染，是通过统一的、标准的、跨平台的OpenGL接口实现的

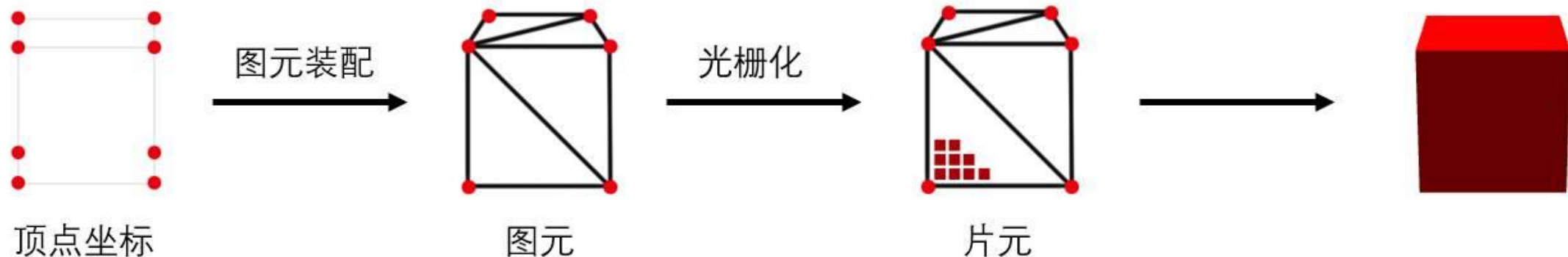
# WebGL相关生态系统



# WebGL执行原理

## • 图形流水线

- 顶点着色器（顶点数量越多，销毁的硬件资源越多）
- 图元装配
- 光栅化
- 片元着色器
- 逐片元操作



# 图形游戏相关引擎概述

---

- 图形绘制相关引擎

- pixi.js、Three.js、D3.js、Artisan JS、Snap.svg等

- 游戏相关引擎

- Cocos、Egret、Laya、CreateJS等

# 小结

---

- Canvas 概述
- SVG 概述
- WebGL 概述



The background of the slide is decorated with numerous overlapping circles in various shades of green and yellow, scattered across the top and right sides.

# Thank You !