

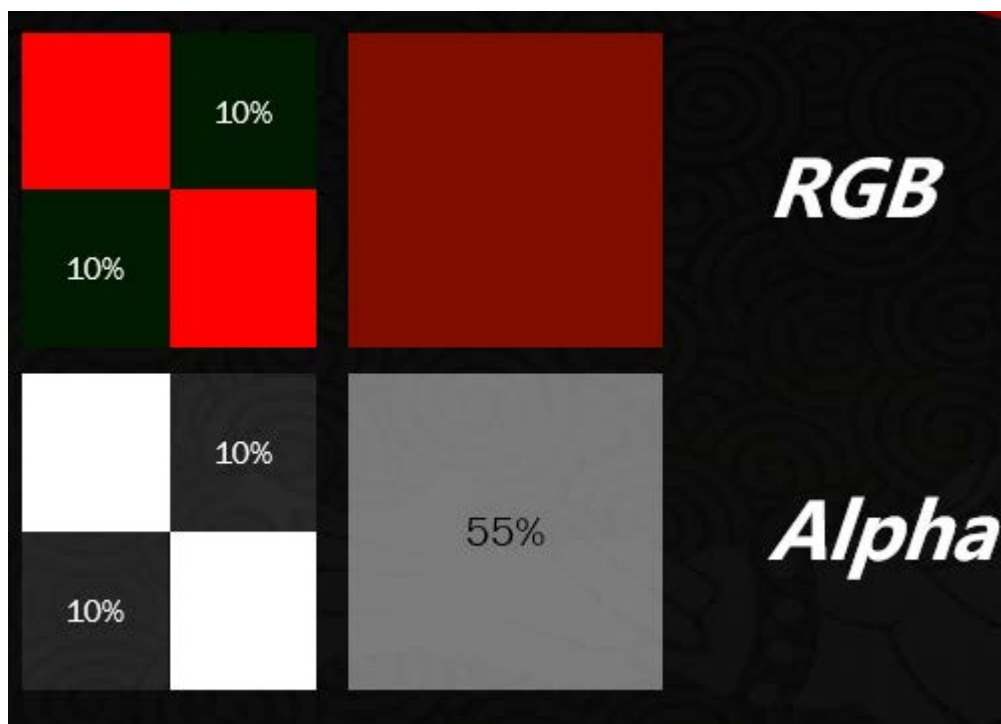
《游戏引擎基础系列课程》之《颜色的计算》课后作业

《颜色的计算》是前端开发基础中的基础，讲清楚颜色的感知、代数计算、线性等问题，可以保证今后的项目开发过程中，大家在颜色上的理解是一致的，避免走弯路。

提交截止时间 2017-7-21 23:59

1 考虑线性空间，完成以下颜色计算

1. 计算



在线性空间下：(255,0,0,255) 与 (0,255,0,26) 的平均值

- 生成一张从 (255,0,0) --> (0,0,255) 线性空间下均匀渐变的 1024x512 尺寸的图片 方法不限，但需要提供实现过程，如果是代码实现的需要提供代码
- 生成一张从 (255,0,0,255) --> (0,0,255,127) 线性空间下均匀渐变的 1024x512 尺寸的图片 方法不限，但需要提供实现过程，如果是代码实现的需要提供代码

2 考虑线性空间，叠加一张特效火焰贴图



你可以把这张图片下载下来，是 PNG 的，没有 Alpha 通道，也不需要考虑 Alpha 通道。分别叠加（ADD）

1. 1 次
2. 5 次
3. 10 次
4. 20 次
5. 50 次

需要提供提供线性空间和非线性空间的输出结果，也就是 **5X2=10** 张图片。图片尺寸请保持不变。请勿使用 **JPG** 等有损格式提交。

3 设计正确的混合操作，不用考虑线性空间

3.1 学习显卡的 Blend 模式

1. 基本的表述都是：

1. $\text{Out.rgb} = \text{Src.rgb} * \langle \text{SrcBlend} \rangle \langle \text{OP} \rangle \text{Dst.rgb} * \langle \text{DstBlend} \rangle$
2. $\text{Out.a} = \text{Src.a} * \langle \text{SrcAlphaBlend} \rangle \langle \text{OP} \rangle \text{Dst.a} * \langle \text{DstAlphaBlend} \rangle$
3. $\text{Blend} := \{\text{ONE} | \text{ZERO} | \text{SRC_ALPHA} | \text{INV_SRCALPHA} | \dots\}$
4. $\text{OP} := \{\text{ADD} | \text{SUBSTRACT} | \text{MIN} | \text{MAX} | \dots\}$

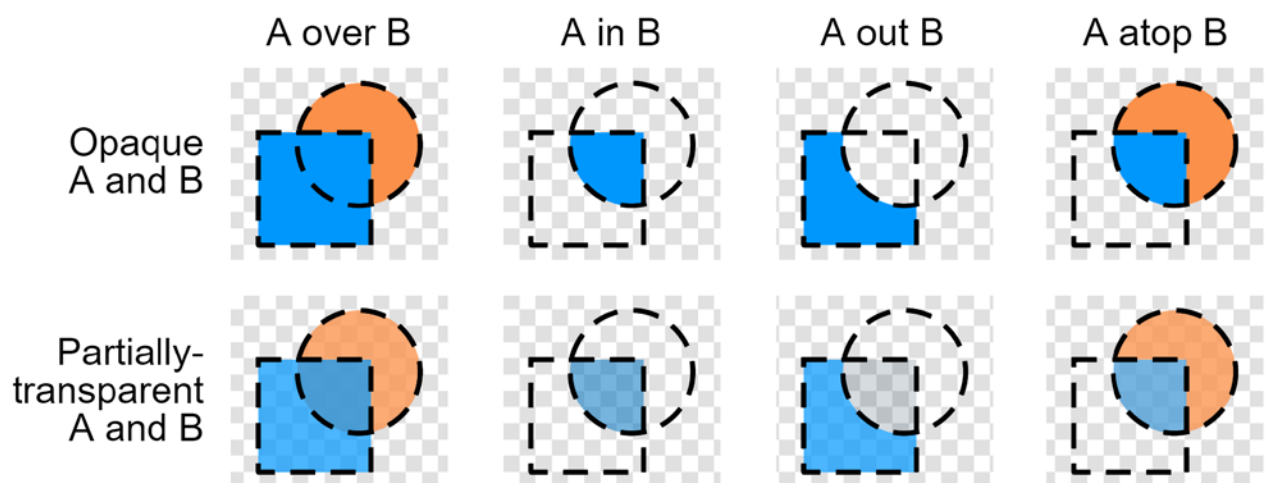
1. [OpenGL 的 Blend](<https://www.khronos.org/opengl/wiki/Blending>)

1. [D3d 的 Blend]([https://msdn.microsoft.com/en-us/library/windows/desktop/bb172252\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/bb172252(v=vs.85).aspx))

3.2 在上述基础上，设计一种特效直接叠加（ADD）的混合模式

1. 样例: $\text{Out.rgb} = \text{Src.rgb} * \langle \text{SrcAlphaBlend} \rangle + \text{Dst.rgb} * \langle \text{ONE} \rangle$
2. **你的设计应该和样例不同**

3.3 设计一组更加复杂的特效叠加算法



根据图片的描述，设计以下混合模式：

1. A over B
 - i. 样例:
 - ii. $\text{Dst.rgb} = \text{A.rgb} * \text{A.a} + \text{Dst.rgb} * (1 - \text{A.a}); \text{Dst.a} = \text{A.a}$
 - iii. $\text{Dst.rgb} = \text{B.rgb} * \text{B.a} + \text{Dst.rgb} * (1 - \text{B.a}); \text{Dst.a} = \text{B.a}$
 - iv. A over B 不用再实现了，做 2、3、4 即可
2. A in B
3. A out B
4. A atop B

3.4 设计一个离屏（Off Screen）半透叠加算法

$$\begin{bmatrix} Draw(A) \rightarrow C \\ Draw(B) \rightarrow C \\ Draw(C) \rightarrow D \end{bmatrix} \equiv \begin{bmatrix} Draw(A) \rightarrow D \\ Draw(B) \rightarrow D \end{bmatrix}$$

A, B 可以参考上一题的示例图片

```
Draw(A) -> C
Draw(B) -> C
Draw(C) -> D
```

与

```
Draw(A) --> D
Draw(B) --> D
```

等价

4 思想实验

既然显示器、人眼（大脑）都不是线性的，那我如何知道那种灰度正好是 50% 的物理灰呢？设计一个实验来验证它。