

---

# 画像の軽量化と利点

# 「画像の軽量化」について

# 「画像の軽量化」とは？

---

画像のデータ容量を小さくすることです。

画像の形式を変更する・使う色数を減らす・

画像自体のサイズを縮小する、

などの手法によってデータ容量を

小さくすることが出来ます。

# 軽量化の利点は？

## 容量制限に 対する回避

- 近年のスマートフォン向けアプリは、リリースするためにストアに提出する際、アプリ容量の制限が設けられています。気を配り開発しなければ、容量の制限をクリアできない場合が多いです。画像を軽量化すれば、回避可能な場合があります。

## ダウンロード 時間の高速化

- ログイン時に画像データをダウンロードさせるアプリは市場に多く存在します。その際にユーザーの待ち時間が発生しますが、容量に気をつけていなければ膨大な待ち時間が発生して、ユーザーがアプリから離れて行ってしまいます。

# 画像軽量化の重要性

## ゲームにおける軽量化の重要性

- 努力してソースコードを短くしてもアプリケーションのサイズが劇的に小さくなることはほとんどありません。
- なぜなら、アプリケーションの中で大きくサイズを使っているのは「画像データ」だからです。
- その上、スマートフォンの画面の解像度は、年々大きく（広く）なっていますが、それにしただがってアプリの容量はどんどん増えていきます。
- より一層画像の容量には気を配るべきです。

# 実際のアプリ想定例（改善前）

## アプリデータ

システムデータなど(5MB想定)

画像データ(30MB想定)



35MBの壁紙アプリを作成することを想定します。

- 1枚の壁紙につき1MB占領します。
- アプリのシステムが5MB占領する場合、**30枚**の壁紙をアプリに入れる事が可能です。



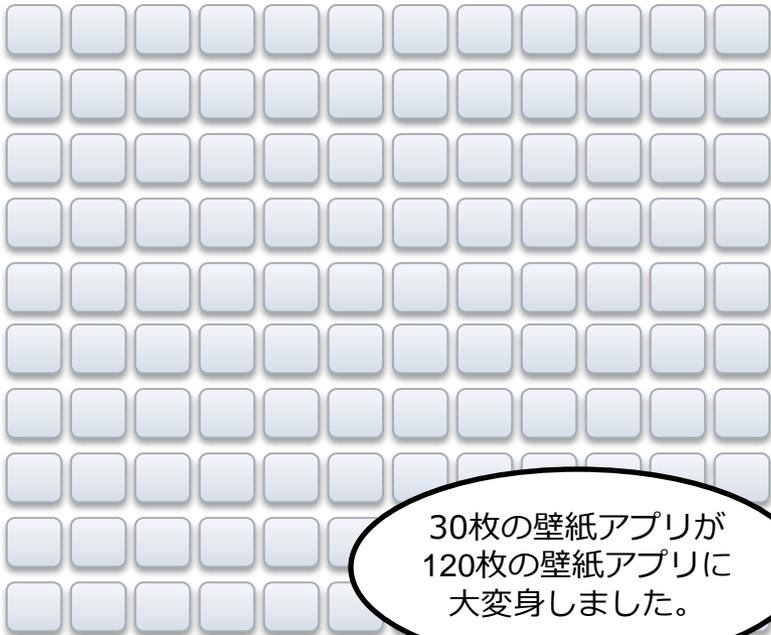
壁紙画像の軽量化を行うと…？

# 実際のアプリ想定例（改善後）

## アプリデータ

システムデータなど(5MB想定)

画像データ(30MB想定)



画像1枚につき容量を3/4  
軽量化することができました！

- 画像の軽量化によって、1枚の壁紙につき0.25MB占領で収まることになりました。
- 改善前と容量は同じアプリなのに  
**120枚**の画像が含まれた壁紙アプリになりました。

市場のゲームアプリの表示物の殆どは画像です。  
分かりやすい例として壁紙アプリを  
例に上げましたが、壁紙アプリでなくても  
同様の効果が望めます。

# 「軽量化の手法」について

# 画像軽量化の手法

---

画像軽量化の手法は様々なものが存在します。  
そしてそれを自動的に行うソフトウェアも存在します。  
ですので、具体的な軽量化を行う手順は  
自身で調べてみましょう。

基本的な手法の中の一つを、次ページで紹介します。

# 誤差拡散法

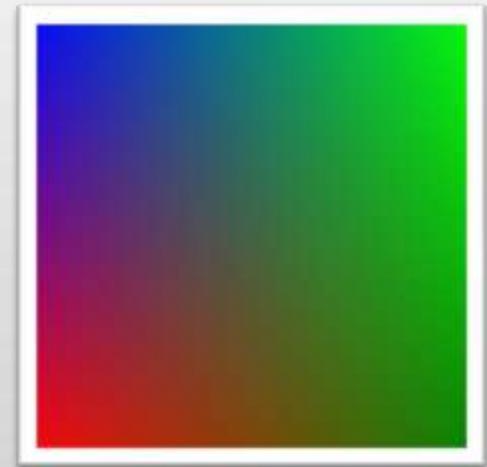
- 画像の表現テクニックの一つに「誤差拡散」というものがあります。
- 「ディザリング」という言い方もします。
- 赤のピクセルと、緑のピクセルを、交互に市松模様のように並べると2色の中間の色が出来上がるように、バランスよく交互に並べると擬似的に多くの色を表現出来る手法です。

右記の画像（32bit RGBA8888）

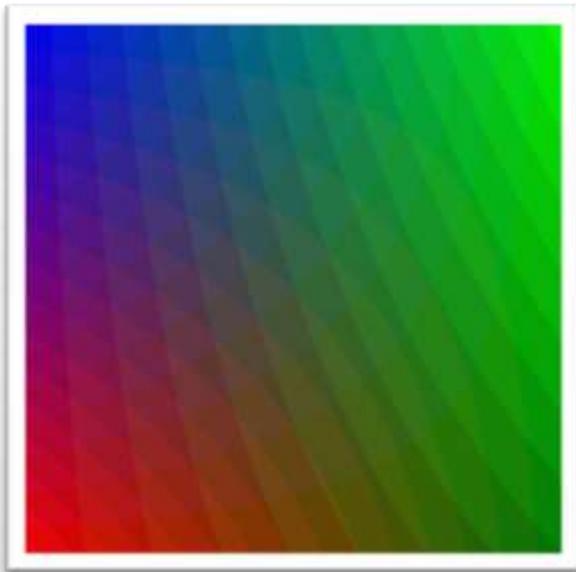
を例に次ページで、誤差拡散法の効果を見てください。

## 補足説明

- 32bit RGBA8888 = 約1600万色と透明度の指定が可能な形式です
- 16bit RGBA4444 = 約4100色と透明度の指定が可能な形式です。

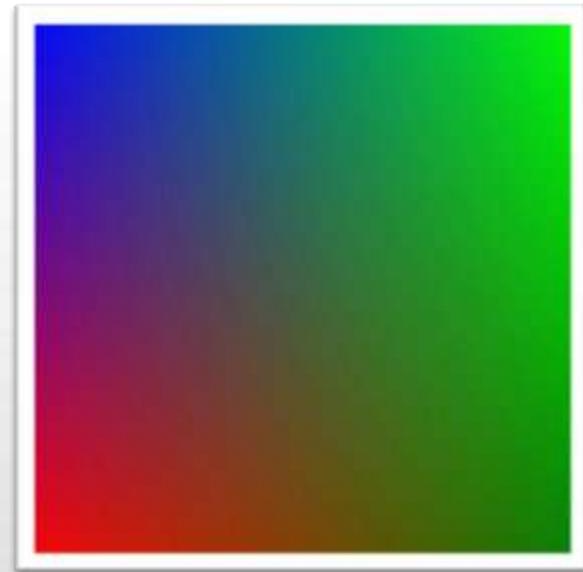


# 形式変更・減色の効果



誤差拡散法を使用せずに16bit RGBA4444 に変換

- 減色の結果、色の段階が少なくなり、縞模様(マッハバンド)が目立ってしまう



誤差拡散法を使用して16bit RGBA4444 に変換

- 色の段階を誤差拡散法で補っているため、マッハバンドが無くなる。

# その他豆知識

モノクロのPNG画像をフルカラーで保存しているのであれば、グレースケールで保存するだけで、画像は約半分以下の容量になります。

グラデーションに対して弱い手法・エッジやコントラストに弱い手法など、手法によって向き不向きが存在します。  
ですから、まずは自身で様々な方法を調べましょう。  
容量を削減したい画像の特徴に合わせて適切な手法を取る事が理想です。

また、タブレット向けに高解像度向けにした画像を使用するより、スマートフォン等向けのサイズの画像を使用して、タブレットでは拡大表示を行うようにすることもアプリ削減には効果的です。  
ただしぼやけた表示になる場合があります。

# まとめ

# まとめ

---

画像を圧縮や減色等の手法で軽量化するべきです。

アプリを作る際、画像の軽量化を行わないと  
容量制限やダウンロード速度で困る場合があります。

ソフトウェアを使用して軽量化を行うべきです。  
(サクセス社内ではOPTPiXを使用しています)

※OPTPiXは、株式会社ウェブテクノロジーが開発した画像最適化ツールです。

# 参考資料01(形式での違い)

内容 \ 形式	BMP	GIF	JPEG	PNG
最大色数	1677万色	256色	1677万色	280兆色
色空間	RGB	RGB	RGB,CMYK YCbCr,グレースケール	RGB グレースケール
透過	×	○	×	○
アニメーション	×	○	×	△
画質劣化	なし	色数が同じなら基本なし	あり	なし

## ワンポイント メモ

- ゲームではPNG形式が多く扱われます
- 色数が豊富で、アルファ値(透過度)が指定でき、サイズが大きすぎない為です。

# 参考資料02(形式での違い)

内容 \ 形式	BMP	GIF	JPEG	PNG
データサイズ	非常に大きい	非常に小さい	非常に小さい	そこそこ小さい
用途	編集中など高画質を求める場面	アイコン、ロゴなど単調な画像	写真やCGなど多彩な色を表現したい画像	透過を生かした画像合成劣化を許さない場面
不向き	データの転送(容量が大きい)	写真など(色数が少ない)	ロゴ、イラスト(ノイズ発生)	データ転送(画像によっては容量が大きくなりすぎ)

# 参考URL

---

- 株式会社ウェブテクノロジー
  - <http://www.webtech.co.jp/imesta/mobile/feature1.html>
  - <http://www.webtech.co.jp/blog/game-develop/2562/>
- KAYAC Designer BLOG
  - <http://design.kayac.com/topics/2014/02/unity-RGBA4444.php>
- パソコン実践講座 ~道すがら講堂~
  - [http://michisugara.jp/archives/2012/img\\_format.html](http://michisugara.jp/archives/2012/img_format.html)