

# 68通信



ロップアップシン Vol.19



# はじめに

この度は、68通信をお手にとって  
くださりありがとうございます。

この68通信はサークル会員達の活動  
の成果を収録したものになります。

作っているゲームの記事であったり、  
技術に関する記事であったり、あるいは  
日々の生活についてのネタ等、様々  
な記事が集まっています。

専門知識のある方もそうでない方にも  
楽しめる内容となっていると思います。

少し長めの内容となっておりますが、  
最後まで楽しんでいただければ幸いです。

X680x0同好会部長  
金井 誠

当サークルの公式サイトではゲームや  
過去の会誌の公開も行っています。  
よろしくお願ひします。

<http://www.x68uec.org/>

# CONTENTS

## MAKING

- BREATH OF THUNDERBOLT II 古田・河内 P3～  
～BEHIND THE SCENES～
- 火消 小岩井 P17～
- Rim Romance 五目 P19～
- StarTarret 開発日記 Ao\_Kamakili P21～
- Future Shinobi Apollo P23～
- HEARTBREAKER で私がやり Tenpre P27～  
たかったこと

# REVIEW

- 日常の魅力 @hamo\_nik P31～
- モバマスニッキ 天麩レ P35～
- チャリで来た。の限界を4629 P39～  
知る

# KNOWLEDGE

- Amanatsu を作ってみた Hiroki P47～
- 提督のための地理 西腋 P55～
- iPad で絵を描こう じいる P63～
- Fami Tracker 事始め 五目 P67～
- 7文字描絵講座 ナスビ P75～
- かんたん! RPN 電卓を使おう 7tuv P81～
- Oculus Rift 入門講座 魯鈍ジョン P93～
- きれいな CD 焼きをしよ(同上) P105～  
う

# OTHERS

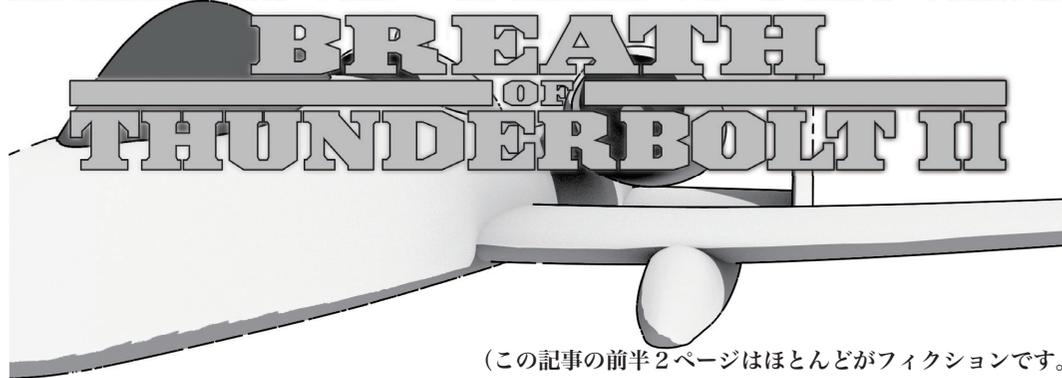
- 目指せブレイブルー脱初 はがね P111~  
心者!?
- “ ちょっとできるように  
なってきた人向け” 講座
- 秘境 NAGATORO のかき氷 金井 P116~  
を出せ
- Single Jouney 音ゲー一人 屏真太郎 P119~  
旅
- 華麗なる泥棒術 fumakura P124~
- THE・闇カレー ~おい パフーハ P128~  
しいカレーを作りたい~
- I AM TREND rinatsu P134~
- SOUND VOLTEX BOOTH 初 くものひと P138~  
心者講座 ~みーつま攻  
略指南書☆ミ~
- アルファを目指すために ツイッタラーT P141~  
やったこと
- 遊☆戯☆王 ろっぱ式 満足 P145~  
No.杯 ~No.1 ハンター  
はこの俺だ!!~
- アカウントが凍結され 暮 P151~  
た時の話
- つり野郎 いーはちえる P153~
- 実践! ボゴソート ゆーてー P159~
- 数独、やろう アロエ P165~



Making



# BEHIND THE SCENES



(この記事の前半2ページはほとんどがフィクションです。  
実際の企業、団体、国家等とは一切関係ありません)

## ストーリー

ある近未来、世界はオーストラリア大陸に墜ちた隕石被害からの復興に手を焼いていた。物語はその隕石墜落災害から一周年の日に始まる。日本・北海道に突如現れた戦車。しかもこの戦車は……おお、見よ！海中から現れたではないか！日本のレーダー網を、そして対潜哨戒網を掻い潜ったのだ！あまりにも常識を逸脱した事態を前に、頼りの自衛隊は駆けつけてくれそうにない。正体不明の暴力から人々を守ることができるのは、主人公の OA-10DJ だけである！彼等はどこから来たのか、何のために来たのか。謎を明かし、世界を守るため、孤独なサンダーボルトが征く――。

## OA-10DJ とは

米国での引退が進む「A-10 サンダーボルト II」を日本で買い取り、Mitsubishi-Fairchild Inc. が様々な近代化改修を施した前線航空管制機である。目的はあくまで自衛隊での航空管制補助——即ち E-767 航空管制機の補助——であったが、C4I システムの拡充を受けてここでも引退。それを日本国内の民間軍事会社で買い上げ、再武装を施した。

21 世紀の今でも、伝説の戦車撃破王ハンス・ウルリッヒ・ルーデルの助言を受けて開発されたサンダーボルト II が魂の輝きを失うことはない。それはルーデルがこの世に残した、陸の安寧を脅威から護る鋼の守護神なのだ。

プレイヤーが搭乗する OA-10DJ が搭載するエンジンは、通常のエンジンとは似て非なるものである。このエンジンは、日本での近代化改修時に試作されたものだ。名を、XTF34-11000 という。「11000」とは、N<sub>2</sub> 一万一千回転までキッチリ回せ（注1）という技術者たちのメッセージだ。オリジナルよりも出力に余裕のある設計の当エンジンは、その低燃費性能から機体の作戦可能時間延長を期待された（注2）。しかしながら、この新エンジンが量産されることは無かった。燃費の良さでは群を抜いていたが、噴射速度に劣る高バイパス比ターボファンと前例のない超多段中圧縮機の信頼性が嫌われたのだ。十分な信頼性を持った従来型エンジンとの比較検討用に、当て馬同然に開発されたとも言われている。そんな不遇のエンジンを2機搭載した、OA-10DJ の実験機版とも言えるのが本機である。

注1) TF34 の後継機エンジン TF39 の民間版の一つである CF6-80 の N2 回転数は 9827、これと比べると実際高回転だ。

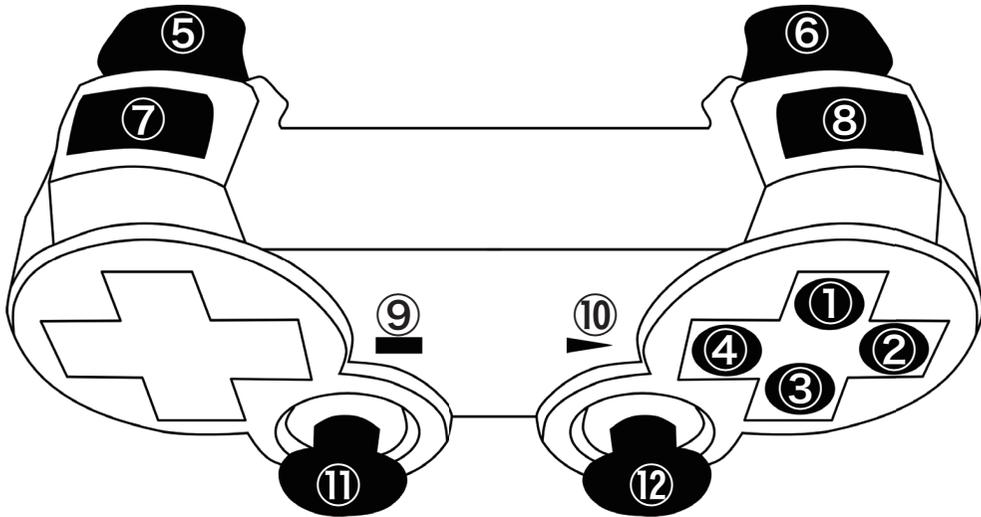
注2) E-767 のような AWACS の弱点の一つは、1機が管制可能な時間があまり長くないことにある。

## 製作動機（インタビュー・ウィズ・センシャより一部抜粋）

——希代のアンタイ・タンク・ヒーローとして認知された、我らがサンダーボルトⅡについて。「最初に生まれたのはmqo ファイルだった。A-10 のモデル。センシャを殺す。知っての通り、センシャ達は陸上を支配する半神のような存在。それを殺すなんて、なんたる反骨の精神か、とね。そこから全て始まった」

「まずサンダーボルトⅡと、あの恐ろしい復讐者のガトリング砲が生まれた。コウチ=サンがチョーフから送って来た exe ファイルを見て、私は稲妻に打たれたようになった。なにかが私たち 2 人の脳に降りて来たような感覚だった」

「すぐに、彼の永遠のライバルであるレッドセンシャ（編注：T-72 か）やツングースカが生まれた。昔から彼らヴィランを知っていたような感覚だったよ。とても、とても不思議だった」



## 操縦方法

本ゲームを起動したその瞬間から、プレイヤーは皆が A-10 神に仕える司祭である。そして、一部の者は未だ A-10 学校どころか、飛行機学校に足を踏み入れたことのないヒヨッコであろう。よって、司祭の常識として、以下にサンダーボルトⅡを操る術を記そう。

言うまでもないことだが、**左スティックが操縦桿**である。基本の操作はまずここからだ。

- ①**ロックオン切り替え**: 標的を切り替えることが出来る。的確に狙いを定め、迅速にこれを撃破すべし。
- ③**機銃発射**: アヴェンジャーを撃ちこむ。これが君達が扱う唯一にして最高の武器である。
- ④**広域地図表示**: 地図を見て、戦況を見極めよ。広い視野を持つことが勝利の鍵だ。
- ⑤⑥**減速/加速**: 敵との相対速度を冷静に合わせよ。速すぎれば敵を撃てず、遅すぎれば自分が撃たれる。
- ⑦⑧**左/右ヨーイング**: 機体を水平方向に旋回させる。目標との水平位置を正確に合わせる際に活用せよ。
- ⑩**ポーズ**: 所謂一時停止である。
- ⑪**自動速度制御**: パイロットの指定した目標速度に合わせ、自動で加減速する。

減速/加速ボタンを押すことで、目標速度は変更することも出来る。

- ⑫**視点切替**: HUD 視点・後方視点を切り替えることが出来る。汝の為したいように為すが良い。ボタン配置は全てが変更可能だ。操作しやすいように各種ボタン配置を切り替えるのも良いだろう。

**「マニュアルの秘訣は3つのK! 気合い! 気合い!! 気合い!!! それでAllRight!!!」**

……醒めちまったこの会誌に……

……熱いのは……

俺達の MAKING……

## 3D の技術周りについて

ここでは、このゲームに使われている 3D 計算の技術について書くことにします。

多少、幾何学や線形代数の考え方が必要になるかもしれないので、そこはご容赦下さい(岡部いさく風)。

### ロール・ピッチ・ヨーを得る

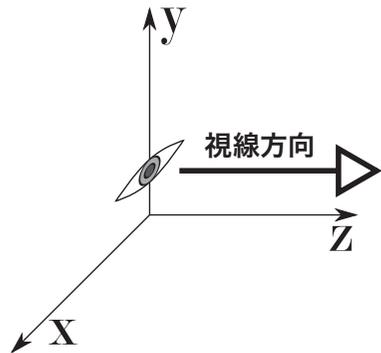
飛行機の操縦で重要な角度(車でも重要だったりするけど)、**ロール角・ピッチ角・ヨー角**は基本的に画面に表示しなければならないでしょう。ですが、これはそれほど容易なことではないのです。

3D での回転表現については、私の 2 年前の記事に詳しく書いたので、そちらを参照して欲しいです。PDF 版が X680x0 同好会のページから閲覧できるので、URL を載せておきます。<sup>[1]</sup>

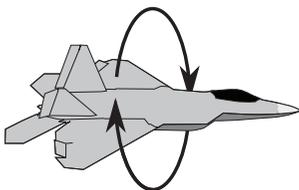
このゲームでは、機体がどのような回転をしているかという情報は、全て四元数で管理しています(四元数については<sup>[1]</sup>の 3 章をご覧ください)。

このままでは画面に反映されないため、オイラー角(カルダン角)に変換してモデルの回転状態を与えます(カルダン角については<sup>[1]</sup>の 2 章を(ry、カルダン駆動とは関係ないので注意)。この四元数→オイラー角の変換と、四元数→ロール・ピッチ・ヨー角の変換は、実は本質的には一緒です。

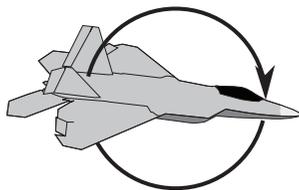
座標系はどちらでも同じ結果となるのですが、仮にプログラミングでよく用いられる左手系を使いましょう。あなたは、Z 軸方向を向いているとします。その時に、右方向が X 軸の正で、上方向が Y 軸の正です。この状態で、Z 軸回転→X 軸回転→Y 軸回転を行うオイラー角の変換がそのままロール・ピッチ・ヨーとなります。Z 軸の回転角が自機の傾き(ロール)を表し、X 軸の回転角が自機の仰角・俯角(ピッチ)、Y 軸の回転角が自機の向いている方角(ヨー)を表していることになるわけです。カルダン角が X 軸回転→Y 軸回転→Z 軸回転を行うオイラー角の変換ですから、ほとんど一緒ということが分かります(左手系なので、Z 軸の回転角が正の時は左にバンクしている状態で、X 軸の回転角が正の時



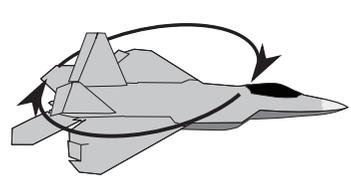
左手系



ロール方向の回転



ピッチ方向の回転



ヨー方向の回転

はダイブする状態、Y軸の回転角が正の時は面舵（注3）を取っている状態になります）

考え方は、変換元と変換先の両方を4×4の行列にしまい、成分の比較を行う方程式を解くというものです。

それでは実際に計算してみましょう。

四元数は任意軸回転行列を用いることで行列にすることが出来ます。<sup>[1]</sup>の3章と4章を見て下さい。まず、単位ベクトル $\vec{u}$ を中心に $\theta$ 回転させる四元数 $q$ を考えます。

$$q = [q_w; (q_x, q_y, q_z)] \quad (1)$$

$$q_x = u_x \sin \frac{\theta}{2} \quad (2)$$

$$q_y = u_y \sin \frac{\theta}{2} \quad (3)$$

$$q_z = u_z \sin \frac{\theta}{2} \quad (4)$$

$$q_w = \cos \frac{\theta}{2} \quad (5)$$

この四元数を式(1)のように表すとき、成分は式(2)～(5)のようになります。

これを<sup>[1]</sup>の式(26)に代入すると、

$$R_Q = \begin{bmatrix} u_x^2(1 - \cos \theta) + \cos \theta & u_x u_y(1 - \cos \theta) + u_z \sin \theta & u_x u_z(1 - \cos \theta) - u_y \sin \theta \\ u_x u_y(1 - \cos \theta) - u_z \sin \theta & u_y^2(1 - \cos \theta) + \cos \theta & u_y u_z(1 - \cos \theta) + u_x \sin \theta \\ u_x u_z(1 - \cos \theta) + u_y \sin \theta & u_y u_z(1 - \cos \theta) - u_x \sin \theta & u_z^2(1 - \cos \theta) + \cos \theta \end{bmatrix} \quad (6)$$

となります。この回転行列はベクトルに対して右から掛け算をするタイプです。もし左から掛け算をしたい場合は、転置をして下さい。

次に、カルダン角の変換行列(X軸回転→Y軸回転→Z軸回転)を計算します。行列を右から掛け算するので、<sup>[1]</sup>の式(7)と同じです。これを<sup>[1]</sup>の式(4)～(6)を代入すると、

$$R_E = \begin{bmatrix} \cos \theta_y \cos \theta_z & \cos \theta_y \sin \theta_z & -\sin \theta_y \\ \sin \theta_x \sin \theta_y \cos \theta_z - \cos \theta_x \sin \theta_z & \sin \theta_x \sin \theta_y \sin \theta_z + \cos \theta_x \cos \theta_z & \sin \theta_x \cos \theta_y \\ \cos \theta_x \sin \theta_y \cos \theta_z + \sin \theta_x \sin \theta_z & \cos \theta_x \sin \theta_y \sin \theta_z - \sin \theta_x \cos \theta_z & \cos \theta_x \cos \theta_y \end{bmatrix} \quad (7)$$

となります。

この式(6)と(7)の右辺が一致するので、各成分から次のようになります。

$$\tan \theta_x = \frac{\sin \theta_x \cos \theta_y}{\cos \theta_x \cos \theta_y} = \frac{R_{E23}}{R_{E33}} = \frac{R_{Q23}}{R_{Q33}} = \frac{u_y u_z(1 - \cos \theta) + u_x \sin \theta}{u_z^2(1 - \cos \theta) + \cos \theta} \quad (8)$$

$$-\sin \theta_y = \frac{R_{E13}}{R_{E33}} = \frac{R_{Q13}}{R_{Q33}} = \frac{u_x u_z(1 - \cos \theta) - u_y \sin \theta}{u_z^2(1 - \cos \theta) + \cos \theta} \quad (9)$$

$$\tan \theta_z = \frac{\cos \theta_y \sin \theta_z}{\cos \theta_y \cos \theta_z} = \frac{R_{E12}}{R_{E11}} = \frac{R_{Q12}}{R_{Q11}} = \frac{u_x u_y(1 - \cos \theta) + u_z \sin \theta}{u_x^2(1 - \cos \theta) + \cos \theta} \quad (10)$$

この時、式(2)～(5)から次のことが得られます。

$$u_x u_y(1 - \cos \theta) = \frac{q_x q_y}{\sin^2 \frac{\theta}{2}} (1 - \cos \theta) = 2q_x q_y \quad (11)$$

$$\begin{aligned} \therefore \cos \theta &= \cos^2 \frac{\theta}{2} - \sin^2 \frac{\theta}{2} = 1 - 2 \sin^2 \frac{\theta}{2} \\ \Leftrightarrow \sin^2 \frac{\theta}{2} &= \frac{1}{2}(1 - \cos \theta) \end{aligned}$$

注3) 船舶において三時方向・スターボード(右舷方向)へ舵をとること。

同様にして、

$$u_y u_z (1 - \cos \theta) = 2q_y q_z \quad (12)$$

$$u_x u_z (1 - \cos \theta) = 2q_x q_z \quad (13)$$

$$u_x^2 (1 - \cos \theta) = 2q_x^2 \quad (14)$$

$$u_z^2 (1 - \cos \theta) = 2q_z^2 \quad (15)$$

次に、

$$u_x \sin \theta = \frac{q_x}{\sin \frac{\theta}{2}} = 2q_x q_w \quad (16)$$

$$\therefore \sin \theta = 2 \sin \frac{\theta}{2} \cos \frac{\theta}{2} = 2 \sin \frac{\theta}{2} q_w$$

同様にして、

$$u_y \sin \theta = 2q_y q_w \quad (17)$$

$$u_z \sin \theta = 2q_z q_w \quad (18)$$

最後に、

$$\cos \theta = \cos^2 \frac{\theta}{2} - \sin^2 \frac{\theta}{2} = q_w^2 - (q_x^2 + q_y^2 + q_z^2) \quad (19)$$

$$\therefore q_x^2 + q_y^2 + q_z^2 = \sin^2 \frac{\theta}{2} (u_x^2 + u_y^2 + u_z^2) = \sin^2 \frac{\theta}{2}$$

以上より、X軸回転→Y軸回転→Z軸回転で表されるオイラー角へ四元数から変換を行う場合は、式(8)～(10)に式(11)～(19)を代入して、

$$\theta_x = \tan^{-1} \left( -\frac{2(q_y q_z + q_x q_w)}{-q_x^2 - q_y^2 + q_z^2 + q_w^2} \right) \quad (20)$$

$$\theta_y = \sin^{-1} (2(q_y q_w - q_x q_z)) \quad (21)$$

$$\theta_z = \tan^{-1} \left( -\frac{2(q_x q_y + q_z q_w)}{q_x^2 - q_y^2 - q_z^2 + q_w^2} \right) \quad (22)$$

となります。同じように、ロール・ピッチ・ヨー回転を得る変換を考えます。Z軸回転→X軸回転→Y軸回転なので、

式(6)と(23)の右辺が一致することから、

$$R_{E2} = \begin{bmatrix} \sin \theta_x \sin \theta_y \sin \theta_z + \cos \theta_y \cos \theta_z & \cos \theta_x \sin \theta_z & \sin \theta_x \cos \theta_y \sin \theta_z - \sin \theta_y \cos \theta_z \\ \sin \theta_x \sin \theta_y \cos \theta_z - \cos \theta_y \sin \theta_z & \cos \theta_x \cos \theta_z & \sin \theta_x \cos \theta_y \cos \theta_z + \sin \theta_y \sin \theta_z \\ \cos \theta_x \sin \theta_y & -\sin \theta_x & \cos \theta_x \cos \theta_y \end{bmatrix} \quad (23)$$

となります。

$$\theta_x = \sin^{-1} (2(q_x q_w - q_y q_z)) \quad (24)$$

$$\theta_y = \tan^{-1} \left( -\frac{2(q_x q_z + q_y q_w)}{-q_x^2 - q_y^2 + q_z^2 + q_w^2} \right) \quad (25)$$

$$\theta_z = \tan^{-1} \left( -\frac{2(q_x q_y + q_z q_w)}{-q_x^2 + q_y^2 - q_z^2 + q_w^2} \right) \quad (26)$$

お疲れ様でした。簡単に結論だけまとめてしましましょう。四元数→オイラー角の変換は、式(20)～(22)を使いましょう。オイラー角を使った変換方法を用いるライブラリはとても多い(注4)ので、この計算は意外と多用すると思います。

注4) 例えば、DXライブラリのMV1SetRotationXYZという関数にはこのオイラー角を与える必要があります。

さらに、機体のピッチ・ロール・ヨーを得たいときは、式 (24) ~ (26) を使いましょう。 $\theta_x$  がピッチ角、 $\theta_y$  がヨー角、 $\theta_z$  がロール角になります。

## 3D の座標から自力で 2D の座標を得る

HUD などを実装する場合、3D モデルを描画している上で、そのモデルと対応するような画像を画面上に 2D で表示したいという状況が生まれます。たとえば、敵の周りを四角く囲いたい、敵が画面上に表示されているかを知りたい、というときです。そういう場合には、ライブラリなどで 3D モデルが画面に表示される工程を、自分で行わなければなりません。

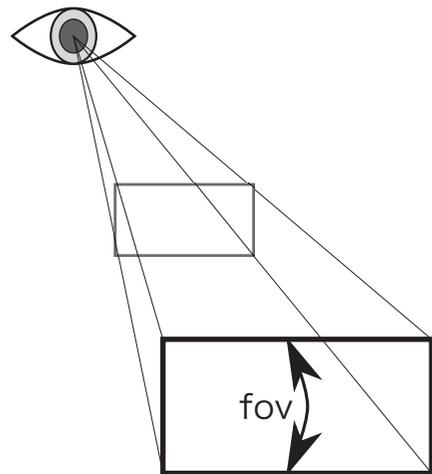
前提として、モデルがどの位置に・どのような回転状態で・どのような大きさで表示されるかを表す変換行列によって、3D モデルの各頂点が変換されますが、これは省略します。この変換で、3D 空間上での座標が得られます。その座標が画面に映るときに、どこに表示するかをこれから計算してゆきましょう。

まずは、その座標がカメラから見てどの位置にあるかを計算します。つまり、カメラからの相対座標です。カメラの基準座標の取り方・3D 空間の座標の取り方などは、後述の Projection 行列・ViewPort 行列をどのように作るかで幾らでも変更できます。しかし、ここでは説明のために定義することにします。本質さえ理解してしまえば、右手系・左手系の違いなどは些細なものだと感じるようになるでしょう。Projection 行列をライブラリから得るような場合でも、その行列がどのような変換を行っているかが分かれば、それに合うように適宜正負を反転させていくだけで対応できるでしょう。

左手系座標において、カメラが Z 軸の正の方向を向いており、上が Y 軸の正の方向で、右が X 軸の正の方向を向いていることにします。これを Projection 行列は、X-Y 平面上にカメラの写しているものを射影するものと定義します。

このとき必要になるのは、視野角 (Field of View) です。この視野角というのは、カメラが写すことができる角度を示します。たとえば、鳥の目などは広角で広い視野角を持っています。この視野角  $fov$  は普通垂直方向についての場合が多いです。一般的な視野角としては、45 度が用いられています。

そのまま縦横同じ視野角を用いてしまうと、正方形な範囲しか得られず、ワイドな画面に射影するときに横に伸びてしまいます。そこで、画面の縦と横の比率であるアスペクト比  $aspect$  が必要になります。これは普通、横幅を高さで割ったものになります。16:9 のワイドディスプレイでは、この値は  $16/9=1.777\dots$  になります。



視野角

さらに、どの範囲の距離 (正確には距離ではなく Z 軸方向での遠さ) を表示するかを決める必要があります。最も近い点  $Z_n$  と最も遠い点  $Z_f$  です。この二つの値は、3D の座標系がどのようなスケーリングで用いられているかによるため、非常にまちまちです。

以上の値から Projection 行列を作ってみましょう。視野角の直線上に存在する場合は、-1.0 か 1.0 となるようにすれば良いこととなります。二等辺三角形になるわけですから、Z で割り算してしまえばよいわけです。Z 軸に関しても範囲内かどうかという変換 (さらに言うと、そのモデルが他のモデルと比べてカメラと近いか遠いかを判断することにも用いられます。この値を並べたものを Z バッファなどと呼び、

モデルの描画順に関与します)が必要となるため、ベクトルの W 成分にベクトルの Z 成分を格納し、ベクトル全体を W で割るという方法が用いられます。

このような変換は次のようにあらわすことができます。

$$x' = \frac{x}{\text{aspect} \tan\left(\frac{fov}{2}\right)} \tag{27}$$

$$y' = \frac{y}{\tan\left(\frac{fov}{2}\right)} \tag{28}$$

$$z' = \frac{(z - Z_n)Z_f}{Z_f - Z_n} \tag{29}$$

$$w' = z \tag{30}$$

このとき  $(x'/w', y'/w') = (-1, -1)$  となる場所がカメラにとっての左上になり、 $(x'/w', y'/w') = (1, 1)$  が右下になります。さらに、 $z'/w' = 0$  となる場所が画面に映る最も近い場所であり、 $z'/w' = 1$  は最も遠い場所になります。

式 (27) ~ (30) をベクトルに対して右から掛け算をするような行列で表現する場合は、次のようになります。

$$\begin{bmatrix} x' & y' & z' & w' \end{bmatrix} = \begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{\text{aspect} \tan\left(\frac{fov}{2}\right)} & 0 & 0 & 0 \\ 0 & \frac{1}{\tan\left(\frac{fov}{2}\right)} & 0 & 0 \\ 0 & 0 & \frac{Z_f}{Z_f - Z_n} & 1 \\ 0 & 0 & -\frac{Z_n Z_f}{Z_f - Z_n} & 0 \end{bmatrix} \tag{31}$$

これが Projection 行列そのものになります。

式 (31) で変換されたベクトルから画面の座標に変換してみましょう。上記のように、 $(x'/w', y'/w') = (-1, -1)$  が画面の左上になり、 $(x'/w', y'/w') = (1, 1)$  が右下となるため、画面の幅を width・高さを height とすると、次で計算できます。

$$x_d = \frac{\left(1 + \frac{x'}{w'}\right) \text{width}}{2} \tag{32}$$

$$y_d = \frac{\left(1 - \frac{y'}{w'}\right) \text{height}}{2} \tag{33}$$

式 (32) (33) を 4x4 行列に拡張したものが ViewPort 行列になります。このとき、Z 成分と W 成分はそのまま渡します。

$$\begin{bmatrix} x_d & y_d & z_d & 1 \end{bmatrix} = \begin{bmatrix} \frac{x'}{w'} & \frac{y'}{w'} & \frac{z'}{w'} & \frac{w'}{w'} \end{bmatrix} \begin{bmatrix} \frac{\text{width}}{2} & 0 & 0 & 0 \\ 0 & -\frac{\text{height}}{2} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \frac{\text{width}}{2} & \frac{\text{height}}{2} & 0 & 1 \end{bmatrix} \tag{34}$$

なかなかシンプルだったでしょう。

(文責：T08 河内)

————— ! とても宇宙的な ! —————

————— ! 初心者向な記事催す ! —————

## 2D/3Dグラフィックについての Tips

予め申し上げておきますが、私の書く記事の全ては少し Google を使えば簡単に調べられる内容です。さて、ここで主役として扱うアプリケーションは2つ。

- **Javie**
- **Blender+Freestyle**

まず、Javie は本記事の筆者がマカーの貧乏人であることから使っている、**映像編集ソフトウェア**です。断言します。ほとんどの人はこれをわざわざ選ぶ必要はありません。Windows にはフリーウェア NiVE という選択肢がありますし、また十分な資金力がある場合は映像編集ソフトウェアの最高峰の一つである Adobe After Effects (以下 AE) を選ぶことが出来ます。本記事で記述した全てのことは NiVE や AE でならより簡単に、より短時間で行うことが出来るでしょう。そもそもエフェクトを作るならそれに特化したアプリケーションが存在します。しかし今回は Javie を前提として記述します。筆者がマカーなので。

Blender は **3D コンピュータグラフィックスソフトウェア**です。フリーウェアながら非常に強力な機能を取り揃える一方で、操作性には強い癖があります。実際、筆者ごときでは到底使いこなせるものではなく、本記事では**既存のモデルを見た目よく出力するためだけに使われる存在**です。今回はそのためにアドオンの Freestyle を使用しています。

### ほのおのにおいしについて Javie

ちょっとだけむせるんじゃ。Javie は、今回エフェクト製作のために活躍してもらいました。完成図は図1のような爆発画像となります。実際のゲームでは、敵機が爆発四散するときにこのような画像が多く散らばる形で使用されています。細かい説明はともかく、まずは図2のようにコンポジションを新規作成しましょう。

続いてコンポジションの設定に関するウィンドウが表示されます。

「幅:」を 480 にし、名前は適当にわかりやすくつけましょう。そして「デュレーション」には初期状態で「0:01:00:02」という値が入っているはずですから、これを「1」に書き換えます。残りの項目は今回関係ないので放置。そして「Finish」を押します。

丁寧な解説書ならここで各種画面の説明とかするんですが、スペースが勿体無いので適当に説明しましょう。画面右上が



図1: ゲーム中と同じタイプの画像

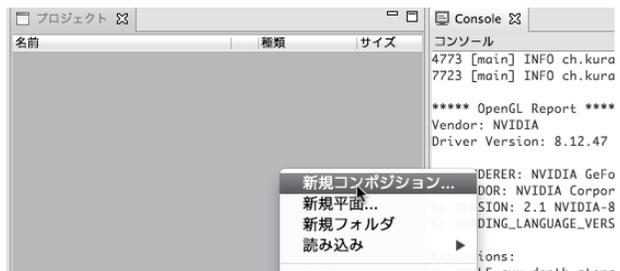


図2: 画面左上部の空白を右クリックして、コンポジションを作成

ビデオ画面、左上が素材一覧、下半分がタイムライン。基本的には左上に D&D してインポートした素材をタイムラインに投げ込んでアレコレすることで、編集は進みます。

コンポジション作成時と同様にコンテキストメニューを開き、上から2番めの「新規平面」を選択しましょう。サイズは今回はコンポジションと同じ 480x480、カラーはオレンジ色としましょう。この色が後々爆発の色を決めますから、悩んでもいいでしょう。今回は写実的爆発が必要なのでオレンジ色としますが、例えばアヤシイ緑色の粒子の爆発が欲しい時とか、ニーズに合わせて変更はしていくべきですね。

先ほど作った平面を2度、タイムラインに投げ込みます。続いて、2つあるうち上の平面の（ここではオレンジ色の）四角の左側にある三角（▶方向）を押します。内容が展開されて（▼方向）、「エフェクト」「トランスフォーム」が並ぶはずで、「エフェクト」を右クリック、図3のように「エフェクト」「ノイズ&グレイン」「フラクタルノイズ」と選びましょう。黒白の煙っぽいものが見えるようになったはずで。

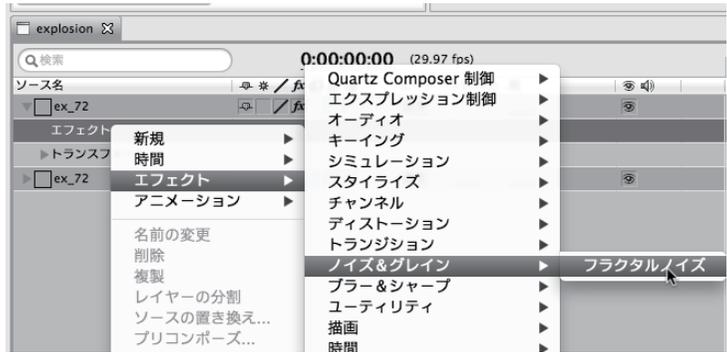


図3：フラクタルノイズを上側のレイヤーへ

続いて先ほどフラクタルノイズを適用した平面の「モード」を「通常」から「リニアライト」

へ（図4）。オレンジ色の光る、ちょっと爆発っぽくなった煙が見られるでしょうか。さあ、一気に調整して仕上げてしまいましょう。

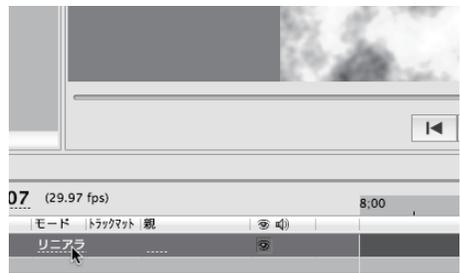


図4：「リニアライト」だけでちょっとそれっぽく？

タイムラインには「▶フラクタルノイズ」という表記が見えるはずで。先ほどと同様に▶をクリック。すると、フラクタルの種類・反転・コントラスト・明るさ……とメニューが並びます。このへんも調整すれば色々と変化を楽しめますが、今回は「コントラスト」「明るさ」「トランスフォーム/オフセット」のみを弄ることとしましょう。大雑把に方針を述べますと、1. 150～200程度にコントラストを上げる。2. 10前後に明るさを上昇させる。3. 「それっぽい形の爆炎」を探し当てられるまでオフセットを動かす。以上です。いくつか手順を増やすと3を省くことが出来る場合がありますが、ここでは扱いません。

タイムライン上の Shift を押しながらの選択で2つとも選んでやり、コンテキストメニューから「プリコンポーズ」を押します。適当に新規コンポジションの名前を与え、「OK」。これを行った時点で、色調以外の調整はほとんど不可になるとお考えください。「これで良いな」と納得できる段階で実行しましょう。

では、「File」「書き出し」「イメージシーケンス...」と選び、適当にファイル名を付けて書き出しましょう（図5）。この時、ファイル名は「(任意).png」という形で、png までをセットで記述しましょう。さもないと、ビットマップ



図5：Javie 上の作業はこれでおしまい

ブで吐き出されてしまいます。ビットマップは透明を扱えませんので。

では、作業の場を Gimp に移します。もちろん、利用可能ならば Photoshop がベターです。ペイント？キミは帰りなさい。ここでは Gimp の使い方は既知であるとします。一応図は掲載しますが、Gimp で画像を開き、図6のように自由選択ツールで爆発の概形を切り取りましょう。この時、暗い部分をなぞるようにすると良いでしょう。また、この時必ず「境界をぼかす」をオンにしましょう。「半径」の値は40以上で適当に取りましょう。作った境界で画像をコピーし、「ファイル (F)」 「画像の生成 (C)」 「クリップボードから (T)」 と選択。切り取られた爆発部分だけを取り出せるでしょう。これを保存すると、図1のようになるはず。これで完了です。お疲れ様でした。

でも、普通の Windows ユーザーならこんなことしてないで、エフェクト作成用ソフト使いましょうね？

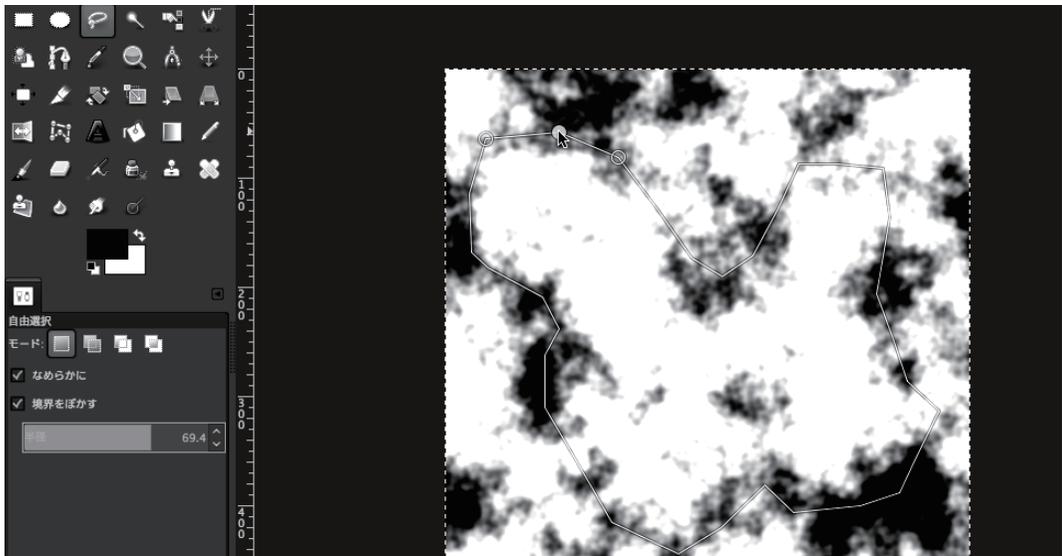


図6：このように きりとるのだ

## アニメじゃない アニメじゃない 不思議な Blender

3D のことさ。さて、当記事ではアプリケーションの日本語化とトゥーン風画像作りのみに触れます。操作に関してはあまり詳しく説明しません。公式運営の Blender Wiki ならばほぼ確実に詳細で正確な Blender についての情報が入手できます。操作やテクニックなどのわかりやすさを重点する場合は「CG 制作」様の Blender の項 (<http://cg.xyamu.net/Blender/>) の閲覧をオススメします。それでもわからない場合はググる。

### 日本語化

では、Blender を起動。ウィンドウ左下の立方体のボタンを押し、「User Preferences」を選択します (図7)。「Interface」「Editing」と並ぶタブから「System」を選択し、画面中央から少し右下に位置するであろう「International Fonts」のチェックを入れます。続いて「Language:」からなんとか「Japanese (日本語)」を探し出し選択、また「Translate:」下の

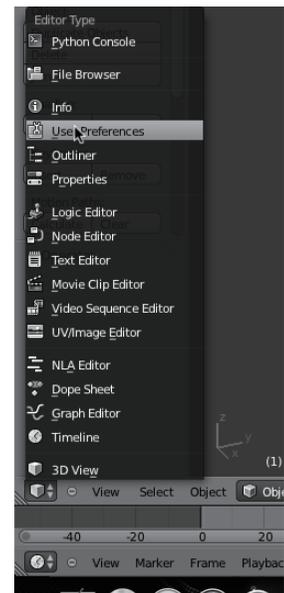


図7：「環境設定」へ

「Interface」「Tooltips」「New Data」も全て押下しましょう(図8)。そして、ウインドウ左下の「ユーザー設定の保存」をクリック(図9)。これで日本語化は完了です。

以上の手順により行った日本語化は多くの場合はユーザーに分かりやすさをもたらします。一方で、Blenderの発展的使い方を求めた場合、そのテクニックは多くが英語で記述された文章や動画の中にあります。逆に言うと、日本語の文献はあまりありません。ゆえに、下手に日本語化を行うとかえって何が起きているのかわからなくなる場合もあります。この際は、先ほど図8で設定した「International Fonts」(日本語では「ローカライズ」)のチェックを外し、英語に戻しましょう。

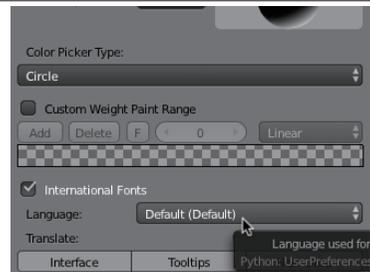


図8：言語設定を変える



図9：ユーザー設定の保存をお忘れなく

### すごく適当な仕組みのお話

今回取り上げる Freestyle は、モデルの輪郭線を描いてくれるアドオンです。実は、Blenderにはアニメ塗りっぽいことをする機能は昔から存在する一方、アニメっぽい線を描画することは簡単ではありませんでした。そして、この線の有無が見栄えに大きな差を及ぼします。例を挙げましょう(図10)。今回は、ある自動車のタイヤ部分をトゥーンレンダリングしてみました。左半分は Freestyle 未使用、右半分が Freestyle を使用したものです。左側ではホイールの輪郭がまったく見えづらく、その一方右側ではその線がために輪郭を判別することができます。これが Freestyle の力なのです！



図10：輪郭線の有無

### カメラ調整

モデルは各々でご用意の上、以下の作業をお試してください。モデルの頂点だの辺だのを変更することはせず、ただ絵を作る、撮影することのみを説明します。

今回使用するモデルは戦車「T-72」(図11)。現代においてはロシアでもっとも有名な戦車であり、やられ役として最も活躍している戦車です。それはフィクション・ノンフィクションを問うことはありません。このゲームでも主敵として大活躍です。さて、視点を回してみましょう。Blenderでは、マウス中ボタンを押しながらマウスを動かすと、視点を回転させることが出来ます。また、Shiftキーを押しながらマウスを動かすと視点を移動させることが出来、マウスホイールで視点を拡大/縮小することが出来ます。これら3つの操作を使って、何はともあれモデルの写りが良い見え方を探してみましょう。完了したら、Ctrl+Alt+テンキーの0を押すか、図12(次ページ)のように「ビュー」「視点を揃える」「現在の視点にカメラを合わせる」と操作していきます。すると、画面中央部以外が暗くなり、また画面中央の明るく残った部分と暗い部分の境界には破線が引かれることで



図11：T-72

図12(次ページ)のように「ビュー」「視点を揃える」「現在の視点にカメラを合わせる」と操作していきます。すると、画面中央部以外が暗くなり、また画面中央の明るく残った部分と暗い部分の境界には破線が引かれることで

しょう。この破線の内側が、レンダリングを行った時に画面に映るものとなります。つまり、視野であります。Shift+F キーで、「フライモード」に入ります。WASD で前後に動く、マウスで視点方向を変えるなど……所謂 FPS ゲームに近い操作感でカメラを操作することが出来ます。左クリックを押すとフライモードは終了します。とりあえず、試しに一枚

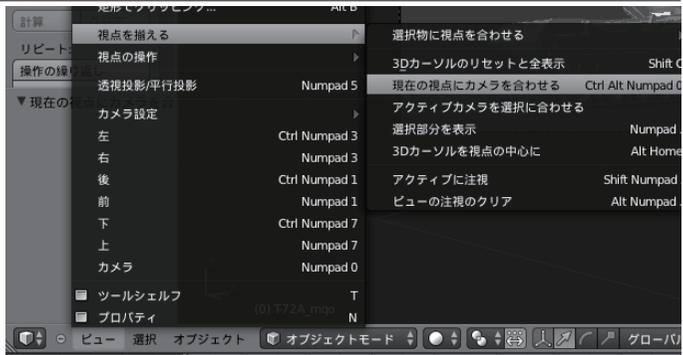


図 12：最もわかりやすいカメラの設定方法

レンダリングしてみましょう。F12 キーを押してください。

時間はかからないでしょう。では、出来上がった図を早速……（図 13）暗い！ なんか暗い！ 下半分何も見えないじゃん！



図 13：ほぼ初期状態レンダリング

レイ・トレースの結果が間違っているとかそんなことはありません。この絵がここまで暗い主な理由は、この戦車を照らす光が戦車左上空に浮かぶポイントライトだけであり、そしてこの照明がかなり強い影を創りだすように設定させていることにあります。では、照明を弄って少し明るい絵を目指してみましょう。

## 光と闇の調整

Das Marchen des……もとい、まずは現在使われている照明を見てみましょう。

視点を引き、図 14 のような照明アイコンを探し出し、右クリックします。続いて、



図 14：照明アイコン

画面右下からランプに関する設定を表示しましょう（図 15）。「▼ランプ」となって いるところが見えるでしょうか。すると、ここに「ポイント」が設定されていることがわかんと思います。これは豆電球のような照明で、遠くに行けば行くほどだんだん光が減衰する性質があります（ただし、今回の場合は減衰の影響を考えなければならないほど遠くに物体は存在しません）。今回のようにアイコンとして使う、写実性よりもくつきりさが絵に求められる場合ではポイントライトよりサンライトが適するでしょう。こちらには減衰がありませんが、一方で指向性のある光です。ポイントライトは無指向性でした。それと、影も調整しまし

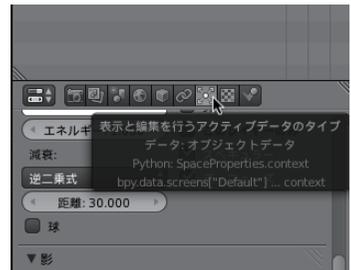


図 15：Lamp 設定

よう。ランプのメニューの一番下から一個上、「影」という設定項目があります。初期状態では真っ黒の部分をクリックすると、図 16 のように色の設定項目が現れます。RGB に直接値を（0～1 の間で）入力してもいいでしょうし、目で見ながら色を変更しても良いでしょう。今回の例では、R=G=B=0.152 としました。続いて、ランプの位置と角度も変更しましょう。ランプからは現在、図 15 のように赤・緑・青色の矢印が出ているはず。この矢印をそれぞれクリックして動かすことによ

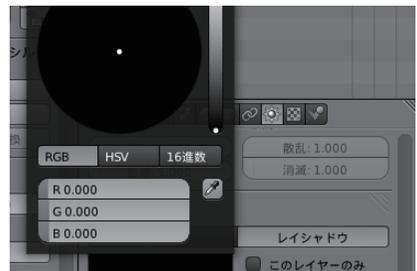


図 16：影色設定

り、XYZそれぞれの方向への照明の位置変更が可能です。また、このままキーボードのRを押す、もしくは図17のマウスポインタの位置にある弧のマークを押すことで角度の変更が出来ます。今回はとにかく図としてのわかりやすさ重視ですから、視線とほぼ同じ方向に光が差すように照明を設定しました。

この状態でレンダリングすると、見た目が遙かに良いことが確認できるでしょう。しかし、影が光に対して少しずつ深くなっていきます。当然の事で、これはモデルの材質が現在ランバートというものに設定されているからです。ともかく、このランバートをトゥーンにしてやらぬことには、今回我々が望む2階調の結果は得られません。それでは地味な作業を始めましょう。

### 材質変更ジゴク

図18を御覧ください。この図と同様のものは、一般的なBlenderの画面では右側に存在するはずですが、「T-72\_mqo」の部分には各々で準備したモデルのファイル名が入っているはずですが。その左、⊕のようなマークを押すと、モデル内のオブジェクトがずらりと展開されます。それらを選び、画面右下側プロパティのタブ群から「マテリアル」（図18では右から4番目の金属球のようなアイコン）を選択します。すると、図では「本体色」とある部分に、そのオブジェクトに属する材質が一覧で表示されます。トゥーン化を行うためにすることは、それらを一つ一つ選び、「▼ディフューズ」とあるところの「ランバート」だの「ミンナート」だのを全てトゥーンにし、サイズを1.5～2.0程度に設定し、スムーズを0.1以下に下げること。そして、「▼スペキュラー」とある部分の「強度」を0まで落とすこと、この2つです。これを、トゥーンに設定されていないディフューズが駆逐されるまで繰り返します。このまま仕上げます。Freestyleによる線付けです。

### Freestyle

Freestyleはありがたいことになかなか秀逸な初期設定を持ちます。もちろん、Blenderの機能である以上、細かい設定もいくらでもできますが。先ほど「マテリアル」を選んだのと同じように、一番左側「レンダー」を選び、その中からFreestyleを探し出し、これにチェックをいれます。続いてその隣のタブ、「レンダーレイヤー」に移りましょう。こちらにも「Freestyle：以下いくつもの設定項目がありますが、最もベーシックな使い方では、その「Freestyle」の中身だけで構いません。この角度より鋭い角には線が描かれるという閾値になる「クリーズ」、線がうまくつながらない時にオンにするとこれが解消するかもしれない「面のスムーズさ」チェックボックスです。とにかく、レンダリングにかけてみましょう。おや、既にほとんど文句のつけようがありませんね

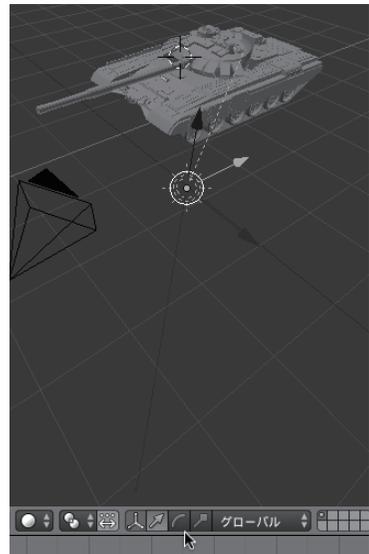


図17：照明方向設定



図18：材質設定

(図 19)。ということは、あまりにもあっさりですが、このサンプルはここで完成ということでいいでしょう。ここからは、いくつかの Tips をご紹介して、記事をおしまいにしましょう。

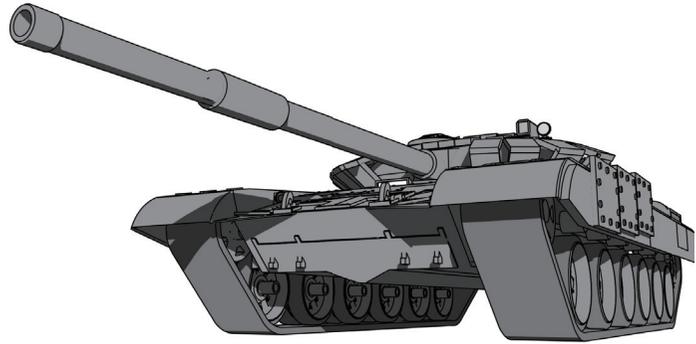


図 19：ほぼ初期状態 Freestyle レンダリング

## 細かなこと

- ・ Freestyle の線を描く処理は、Blender による「塗り」が完了してから行われます。よって、レンダリング結果にまだ線が描かれていなかったとしても、それは多くの場合不具合でもなんでもなく処理が完了していないだけです。
- ・ 「プロパティ」の「レンダー」には解像度の設定があります。つまり出力サイズの設定です。「縦」「横」「大きさのパーセンテージ」と並びます。「値とか色とか、これでいいのかな」というテストの間はそのパーセンテージを 50% とか、25% とかに絞っておくことをオススメします。色々弄ったモデルのレンダリング結果が、超長時間をかけて悲惨なことになったら辛いですし。
- ・ 「ここ、ちょっとだけカクカクしてるんだけどなんとかならんかな」と思ったら、その部位を右クリックで選択し「シェーディング」の「スムーズ」を押してみましょう(図 20)。改善するかもしれません。ただし、これを有機物に使うと物体が溶ける可能性がありますので注意しましょう。
- ・ スムーズだと溶けるんだけど滑らかにしたい！ という時はモディファイアの辺分離 (EdgeSplit) と細分割曲面 (Subsurf) が役に立つかもしれません。プロパティにモディファイアというタブがあり、ここから設定できます。詳細はおググりください。
- ・ Freestyle の出力結果の画像に、角なんかありやしないところに断片的に線が描かれた場合は、クリース角度設定の下にある「カリング」チェックボックスをオンにしてみましょう。解決する場合があります。
- ・ 最初から説明しておけ、レベルの話ですが、アドオンを追加することによって Blender は実に様々なモデルを読み込むことが出来ます。今のところ、直接の読み込みが難しいのは Google 3D のモデルぐらいでしょう。例えば、Metasequoia 用のインポート・アドオンはインストールしておいて損することは無いでしょう。またこのインポーターの作者は MMD 用のインポーターまで作っていらっやいます。MMD を触る場合は、これも有用です。

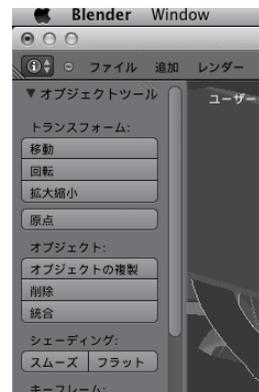


図 20：シェーディング設定

## 最後として

とっつきにくさに定評のある 3D という世界でもさらにとっつきにくい分類に入る Blender ですが、そのうちの限定的な機能を使う分には私のような無学の人間でもでもなんとかなりました。ゲーム開発も、これからはだんだんと 3D の時代かもしれません。それを先取りできる技術を部分的にでも会得できるなら、それはあった方がかっこいいじゃん。

(文責：C09 古田)



3J 小岩井

## 1. 当記事について

この記事では、私が制作したゲーム「火消」のたまかな説明と制作中に考えていたことなどを掲載したいと思っています。ゲームの詳しい説明は別に用意してある（予定）なので、この記事を読んで下さった方は是非ゲームの方もプレイしてみてください。

## 2. ゲーム概要

これはリアルタイムのシミュレーションゲームです。江戸ライクな町の中で発生した火事の被害を最小限に留めるために、昔の消防隊である「火消隊」の指揮官となって隊員に消火活動を行わせて下さい。

火消隊が行う消火は炎の近くの可燃物を破壊することで延焼を防ぐ破壊消火です。逃げ惑う市民や跋扈（ばっこ）する犯罪者達の妨害に対処しながら、より多くの家を火災から守って下さい。

## 3. ゲーム説明

ゲームは町の区切られた一画で進行し、様々な建物や人及び火元が配置された状態で開始します。時間経過に合わせて火元から周囲の道や建物へと延焼し火事が広がりますが、隊員達に指示を出し火元近くの建物を破壊しておくことで延焼を防ぎ、その先の建物を火災から守ることができます。隊員達に適切な指示を出すことで、町に点在する防衛目標を鎮火するまで守り抜くことがこのゲームの目標です。

ゲーム内には主に3種類の人間がいます。1つが先程から触れている、プレイヤーの指示通りに行動する「火消隊員」。もう1つが炎から逃げることしかできない「市民」。そして最後の1つが混乱に乗じて悪さを働く「犯罪者」です。犯罪者は隊員を隣接させることで逮捕することができます。

ステージ内の全ての炎が鎮火した時点でゲームクリアとなり、その時まで保全された建物、生存した隊員と市民、犯罪者の逮捕数とその他ボーナスに応じて得点が与えられます。一度クリアできたステージでも、更なる高得点を目指して再プレイしてみてください。



## 4. 制作経緯

私がゲームを制作する際に意識していることがあります。それは「一般的なゲームは作らない」または「既に世に存在するゲームは作らない」ということです。ゲームには様々な種類がありますが、それぞれに代表的なシステムというものがあります。例えば、シューティングゲームと言われて多くの人が想像するのは、強制スクロールの画面内で敵を倒しながらアイテムを回収するようなシステムでしょう。私はこのような「代表的なシステム」を有するゲームを作らないようにしています。

このような考えは私の天邪鬼な性格に起因していますが、私に自信が無いことも要因となっています。世界中でプロアマ問わずにゲームを作っている人は大勢おり、その中の多くの人々が一般的なシステムを有したゲームを作っているでしょう。私はそのような人々より面白いゲームを作れる自信がありません。特に難易度調整やステージ作り、インターフェイスなどのゲームデザインは私が最も苦手とする分野です。なので、私はゲームデザインではなくゲームのアイデアで人を楽しませようと考えています。だから私はひねくれたゲームを作るのです。

前置きが長くなりましたが、このゲームのコンセプトは「プレイヤーにやむを得ない犠牲を払わせること」でした。最終利益を大きくするために目の前の利益を切り捨てる、そのような非情な判断をプレイヤーに迫るゲームを作りたいと考えたのです。実際に完成したゲームでは、ゲームクリア時に残存した家と人の数に応じてスコアが加算されるようになっていました。スコアを増やすためにはより多くの家と人を救えば良いのですが、クリアのために幾つかの家を破壊し、場合によっては人を犠牲にする必要があります。プレイヤーの皆さんには、このような「ままならない状況」の中で苦悩して頂ければと思っています。

ここまで読んで下さった物好きな皆様は私のゲームもプレイして頂けることでしょう。ご高覧いただき有難う御座いました。

# Future Shinobi Making

2J Apollo

## ゲーム概要

この作品は2D 横スクロールアクションです。このゲームでは忍というキャラクターを操作して遊びます。忍は高い身体能力(5段ジャンプや壁ジャンプ)と、刀、銃、忍具、そして忍術を用いて敵を倒していきます。実装できれば、ある特定の攻撃でしか倒せない敵や、進めないギミックも考えています。

攻撃の種類	攻撃の特徴
通常攻撃(パンチとかキックとか)	近距離で威力は普通
刀	近距離で威力はやや高め
銃	遠距離で威力はまちまち
忍具	遠距離で威力はやや高め
忍術	中距離で威力は高め

## 開発話

ドーモハジメマシテ、ドクシャ=サン、Apollo です。

このゲームは去年の調布祭が終わってから製作しました。最初のころ、ゲーム制作のプログラミングはおろか大学の講義のレベルすら理解していなかったので、非常に苦労しました。ゲームプログラミングができる友人に教わりながら文法を理解して

いったのは今でも覚えています。

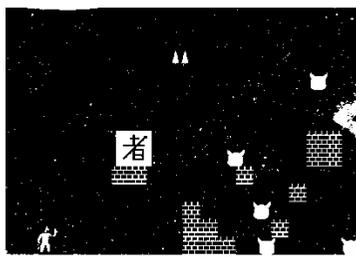
プログラミングでつらかったのはこの作品の最も重要なアクションである壁ジャンプです。これの実装に一週間以上はかかったと思います。

また、最初は重力があるシューティングを考えていましたが、ゲームの操作性が悪かったのと、アクションゲームが好きだったこと。それに加え某漫画の影響を受けて、今のような仕様に変更しました。ちなみに、最初に作った時の名残としてホバリング(5段ジャンプ)を仕様として残しています。しかし、期限にまで間に合うか微妙です。はい。調布祭までに公開できるように邁進していきます。そして、担当の絵師からまだ主人公の絵が完成していないので画像も貼れないという有様です。もう少し急かすべきだったか？

調布祭終わった後の開発者に一言、お疲れ。新しい社畜が始まるとおもうけど。

最後に私のわがままを聞いてくれた絵の担当の2K 時空の操者、いろいろな建設的な意見をくださった方々や BGM を提供してくれた方々。そして、プログラムが全く分からなかったころに基本をいろいろと教えてくれたり、ゲーム開発の相談に乗ってくれたりした2M あおかまに感謝します。

記事を読んでくださり、ありがとうございます。では、また逢う日まで。



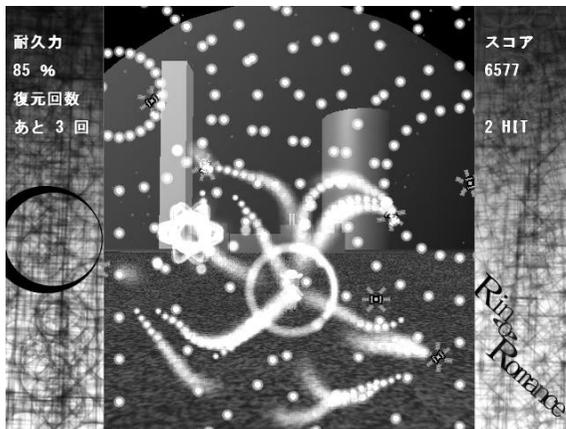
(※画像は開発中のものです)

# Rin Romance

## 弾幕「干渉」シューティングゲーム

11 五目

このゲームは、近未来的な雰囲気のある弾幕シューティングです。大きな特徴は、「敵の弾幕に干渉する」という特異な攻撃手段です。通常ショットが撃てない代わりに、敵の弾を跳ね返したり、弾を吸収したりできます。基本的にはこの「干渉」を軸にステージを攻略します。ゲーム自体の説明やルールはマニュアルを確認してもらおうとして、ここではどのような経緯で制作に至ったかを適当に語ります。適当に。



弾幕も激しいが、こちらの攻撃も強力。

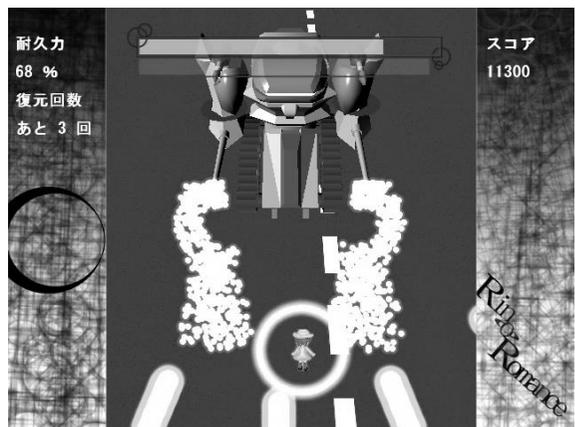
### タイトルについて

「リノ」は主人公の名前で、確か「ニュートリノ」から取ったと思います。「ロマンス」は、まあ「R」から始まる単語で適当に付けました。元々「弾幕に干渉しない」というコンセプトから開発が始まったので、その辺りのイメージをニュートリノに結びつけたのかな。流石にそのままだとゲーム性がないので、「干渉」というキーワードを基に、「弾幕干渉シューティング」という方向性が決まりました。

### 世界観について

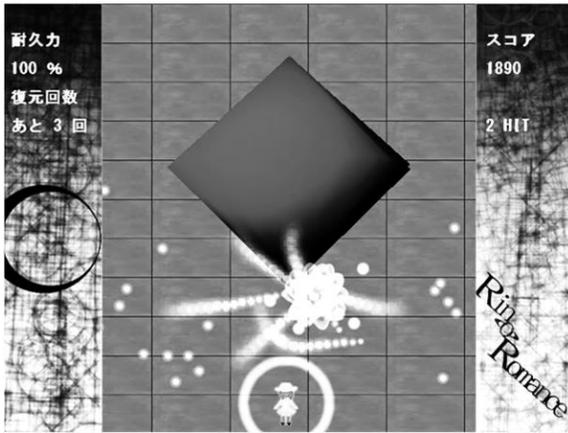
ざっとかいつまんで書くと、

- 1.地球の環境破壊や人口過多がヤバい。
- 2.月面移住計画が急ピッチで進む。
- 3.月日は流れ、月面コロニーが点在し人類の移住が進んでいく。
- 4.そんな中、第3コロニーの通信が突如途絶え、生物反応が消える。
- 5.原因調査とテストも兼ねて、試作型の人工生物兵器「リノ」が月へ出発。



巨大なボスも出現。

……なんとリノちゃん、ガイノイド（女性版アンドロイド）だったのですか。「エネルギーに干渉して自在に操る能力」って、恐ろしいと思うの。



このオブジェクトは一体……？  
備えよう。

## 進捗、どうですか

ステージの構成はほぼまとまっています。全五面の予定です。つい最近ラスボスが完成しました。やったね。恐らくこの調子でステージ全体を完成させた後、バランスの調整をしていくことになると思われます。

「背景を 3D にしてボスもでっかい 3D の奴を出したいねー」という当初の目的は達成されたので、あとはウイングランということになります。3D の処理に関しては、DX ライブラリをそのまま用いたのですが、思ったより簡単にモデルの表示やテクスチャの移動などを行えたのでよかったですね。カメラをグリグリするだけで何か凄いことをしている気分になれる。

このゲームは私一人でプログラム・絵・音全てを担当しているので、やろうと思ったことがすぐに反映できてそれはそれで良いのですが、それだけ自分がやらないとまらない仕事が増えるので大変です。よく考えたらまだ作らなきゃならない曲や画像素材が色々残っているので、これから急いで作ります。今回は音源に SD-90 を使っているので、より「らしい」曲になっているかも知れません。何がとは言いませんが。

これまで私が調布祭に向けて作ったゲームに「エンディング」が付くことはありませんでした（すなわち、未完成）。今回こそはきちんと「終わらせる」ことが（たぶん）できると思うと感慨もひとしおですね。

それでは、お楽しみください。

七か七部長と副部長でゲームのジャンルと雰囲気は被るとかなかなか

十月某日

五目

# StarTarret 開発日誌

ジャンル「全方位弹幕シューティング」



'12M Ao\_Kamakiri

## 0. はじめに

初めましてこんにちは。X680x0 同好会所属、2年のAo\_Kamakiriです。このページでは自分の制作する「StarTarret」というゲームについて紹介しようと思います。このゲームは調布祭にて配布されるX68DVDで遊べます。あ、Twitterやっています。よろしくお願いします。

## 1. ゲーム概要

### 1.1. コンセプト

このゲームはクォータビュー視点で展開されるシューティングであり、四方八方から迫りくる敵を倒すことが目的です。似たもので、某STG「地球防衛軍」をビュー視点に下ろしたものでしょうか。

主な操作は、2つのアナログスティック

で移動・攻撃を行います。複雑なボタン操作が少ない分、直感的にプレイできるでしょう。

また、本作はマルチプレイ対応で、最大4人で協力して遊ぶこともできます。力を合わせて敵を殲滅したり、または獲得スコアを互いに競ったりします。

## 1.2. ストーリー

妖精と人が暮らす平和な銀河系。そこに突如、謎の侵略軍が現れた。彼らは圧倒的な物量で銀河系の惑星を呑み込んでいき、残った惑星は地球のみとなってしまった。地球に住む人と妖精は、奪われた惑星を取り戻し、銀河系を救うため、最後の希望を一人の小さな妖精に託した…。

……な、なんですか？ てきとうに考えたわけではないんですよ（汗）。

要すると、妖精でビュー視点弾幕ゲーがしたかったんですよ。

ゲームの舞台は前述のとおり太陽系で、各惑星がステージとなります。第1ステージ「地球」がチュートリアル面です。今作では「金星」「火星」「水星」「小惑星帯」のステージを実装する予定です。

## 1.3. 武器について

プレイヤーはデフォルトでバルカンが撃てます。威力は低いが連射が効く、どのシューティングでも初めに使えるアレです。しかしゲーム中では武器アイテムを拾うとより強力な攻撃が使えます。種類は次

の3つです。

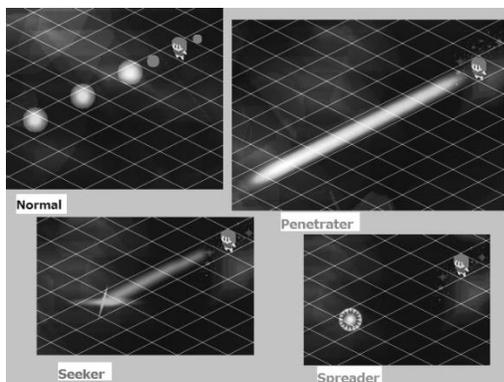


図1. 各種武器

\* 図は開発途中のイメージです。

○ペネトレーター（図右上:Penetrator）

貫通性能を持ったビームです。敵を貫通して直進するので、敵の集団に高い効果を発揮します。威力も高いのでボス敵にも有効です。

○スプレッダー（図右下:Spreader）

拡散性能を持ったエネルギー球です。敵に触れると爆発する、ロケットランチャーのようなものでしょう。連射は遅いですが、かなり広範囲にわたって敵にダメージを与えます。

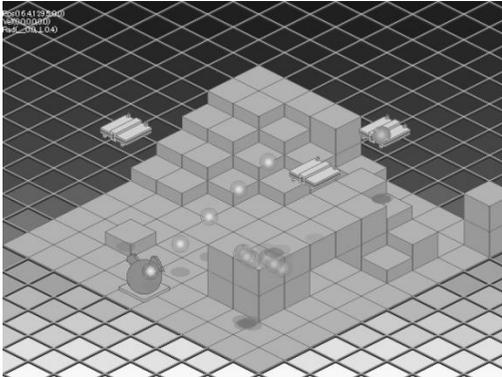
○シーカー（図左下:Seeker）

追尾性能を持ったレーザーです。近辺の敵を捕捉、追尾するので楽ちんです。発射直後は直進するのでそれなりに狙う必要があります。

文章で説明するだけでは不十分でしょう。それでは実際にX68ブースで体験してみてください！（なげやり）

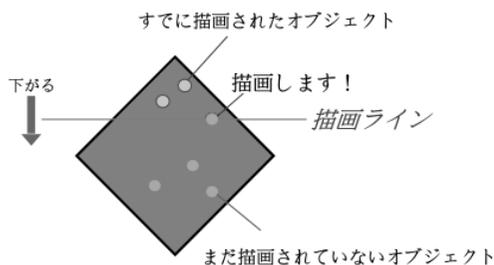
～ 以下、苦労した点などを日誌形式でお送りします ～

## 4月



始めはタイトルもまだ決まっておらずフィールドも小さいプロトタイプでした。「3次元の表現ができる」ということで、上から敵を降らしてみただけですが、プレイヤーが敵を狙いにくいなど問題点は様々でした。

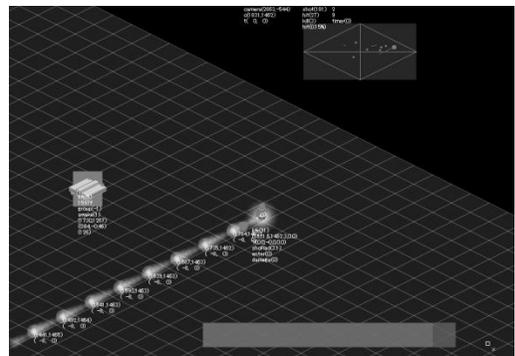
その一つに「重ね合わせ描画」です。通常奥にいるものは手前のものに隠れますが、これをプログラムでどう表現するか悩みました。解決策として、上から描画ラインを少しずつ下ろしました。



このラインはゲーム上の地面に沿ったもので、ラインに辿り着いたものから順に描画するという手法です。この描画ラインは1フレームの間に上から下まで流すの

で、その分処理は重くなりますが、ラインの移動幅を大きくすれば負担は軽減します。これで不自然な重なりは解決しました。

## 5月



フィールドが広くなりました。背景が真っ暗で寂しいですね。

ショットの残影エフェクトを作っていました。というかこの時期はエフェクト改造しかしていなかった気がします。

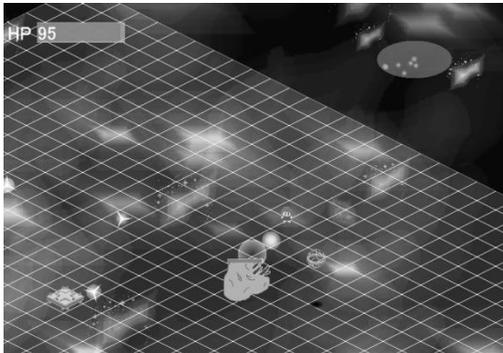
敵は上から振ってくることはなくなり、その場にシャキンと出現するようになりました。さらに敵の行動も、「覚醒」しているか「待機」しているかで分けるようにしました。このゲームの敵は始め「待機」していますが、近づいたり攻撃したりすると「覚醒」し、プレイヤーに襲い掛かるようになります。フフフどうでしょう！ こういうシステムを思いついて実装することが開発の楽しさってやつですね！ えっ既出？ ソナア……

## 6月

この頃も相変わらずエフェクトばかり弄ってましたが、新たな敵や武器の設計もちゃんとしてました。敵の挙動については、横STGのように、画面を出るとそのまま消滅するということはありません。その場に居続ける動きをしなくてはならないと思っていました。ひたすらプレイヤーに向けて加速したり、プレイヤーの周りの指定位置を漂ったり、或いは移動せずに射撃し続けたり、一定間隔で突進したり……色々工夫をしてました。

今思うとこの頃は、絶対に自然消滅しないというシステムにこだわり過ぎていた気がします。

## 2. 7月



フィールドに背景描写を実装しました。静止画では分かりにくいかもしれませんが、背景レイヤを3枚（奥の奥、奥、手前）に分け、スクロール量を分けることでより立体的な見栄えとなりました。

背景レイヤは、いくつかの小テクスチャを数百個ランダムに散らばせて描画する

ので、遊ぶたびに背景が微妙に変えることもできます。

## 8月

期末試験を終え、夏休みです。ツシャア遊ぶぜ！ あ、開発もしっかり進めますから大丈夫デスヨー!! プログラムだけでなく、絵や音の素材も作らなきゃですー！

## 9月

そろそろクッキーが9,000,000,000,000枚を超えるな……。ここまで進むと施設の増設でもCpS (Cookie per Second)の上昇も微妙になるし、あとはひたすら放置して待つしかないかなあ。おっ！ ゴールデンクッキー！ に が さ ん ！

ふう……。放置もいいけど、手を動かしてクッキーを増やすっていう作業も楽しいんだよねーこのゲーム。

……。そういえば何か忘れているような……。ん……。？

(終わり)

# HEARTBREAKER

で私がやりたかったこと

11I tenpre

## 1 あらすじ

そう遠くない未来、人類がまだ「人類」しかいなかった頃。その世界の星の煌めきは恐るべき犠牲と新たな「生物」を生み出した。地球への「隕石」の落下。とても隕石と呼べる大きさではなかったが、地球へと降り注いだ。まるでそのようなる



という夢を見たのさ。

## 2 ほんすじ

このゲーム、HeartBreaker は前回の調布祭から作ってきたのですが、今回再度発表するまでに基本システムや画像等が少しずつ変化してきました。正直迷走していました。主に私——というか私個人が。

この記事ではどういう目的、意図でシステムを考えたのかとか、どういった雰囲気を作った（作りたかった）のかを徒然なるままに書き連ねてみようと思います。

## 3 システム

### 3.1 ゲージ共有システム - Heart ゲージ

サブタイまんまだー。という事でこのゲームの主なところのゲージシステムです。このゲームのメインシステムは当初某ゲームのプレイ<sup>\*1</sup>を見て思いついたものでした。攻撃ボムと防御ボム、残機は一緒のゲージにして、プレイヤーが状況に合わせて使い分けていけばプレイしやすくサクサクと進めるのではないかと、というのが当初の思惑でした。

で、す、が、そんな上手くいく筈もなく……。

まず露わになってきたのが攻撃ボムと防御ボムについては、ちゃんと目で分かるレベルの差別化を図らなければならないということでした。システムの違いがあっても目で効果の違いが分からないと、遊んでいるプレイヤーはどのタイミングでどちらを使えばいいのか、というのが直感的に分かりにくいという問題が露わになりました。

例えば私の大好きな彩京のガンバード2というゲームでは、通常の個数制のボムの他にパワーを消費して発動する、近距離攻撃というものがありました。このゲームは最初は効果の大きい

<sup>\*1</sup> CAVE の iPad 版怒首領蜂大復活が「ボムは残機ゲー」に見えたのがきっかけ。あくまでも個人の主観ですのでええ。

ボムを使いがちですが、徐々にその個数による制限を感じて使わなくなっていきます。逆に敵の配置等が覚えられるようになってくると、敵が出てきた瞬間にパワー消費で何度も使える近距離攻撃で倒す、という戦法を思いつくようになります。最終的には敵が弾を打つ前に近距離攻撃で潰して、危なくなったらボムを使う、という非常にパターン性の強いオーソドックスな 2DSTG として定着してきます。このようにちゃんとシステム上で明確な役割を決めておき、後は見た目でも分かるような差別化が図られていれば、メインシステムとしては申し分のないものとなると思います。

これは片方をインフレさせればある程度は解決しそうですが、今回のシステム的にはある程度均等なバランス関係を保ちたいと思います。出す頃にはある程度落ち着いた形になってるんじゃないかな！ た、多分。

**教訓・目で見える程度には違いをつけよう。**

## 3.2 近未来(?)な雰囲気

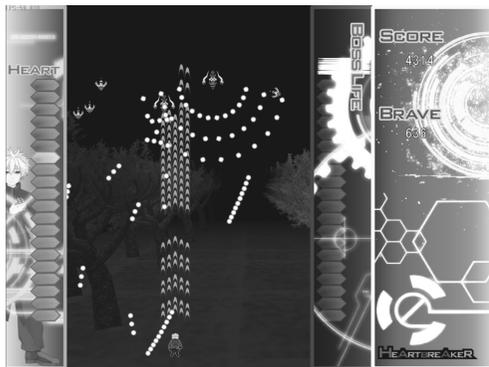
今回の一番のミス、と言っても過言ではない(いきなり?)と思われる部分です。正直、他の人に一番迷惑をかけた気がします。

さて質問ですが、皆さんが「近未来」というと何を思い浮かべるでしょうか？ 20世紀に想像された未来都市？ はたまた現代よりも少しだけ技術の進んだ今風の都市？ はたまた SFチックな世界でしょうか。果たして何が「近未来」なのでしょう。

……つまり私の失敗とは「近未来」の「近」という言葉について深く考えてなかったのです。「近未来」という言葉には多数の解釈があるでしょうし、そのどれもが正解、となるでしょう。古代とか中世とかならある程度は具体的なイメージは出来ると思います。それは今という基準があり、そこからの尺度を図ることができるからでしょう。一人一人のイメージにはズレがあるので、そこをしっかりと合わせる必要があるのです。

結局のところ「近未来」という聞いた感じカッコ良さげなこの言葉、そのイメージを固めるのが不足していたのだらうと思います。実際には E.S.P. RADE \*2 のような世界観をちょっと進めた感じをイメージしていましたが、私の計画性とコミュカの不足で伝わらなかったのは残念でした。もうちょっと考えておけばよかった、と後悔してます。

**教訓・人にもものを頼む時は、ちゃんとイメージを考えてしっかり伝える。**



## 4 調布祭について

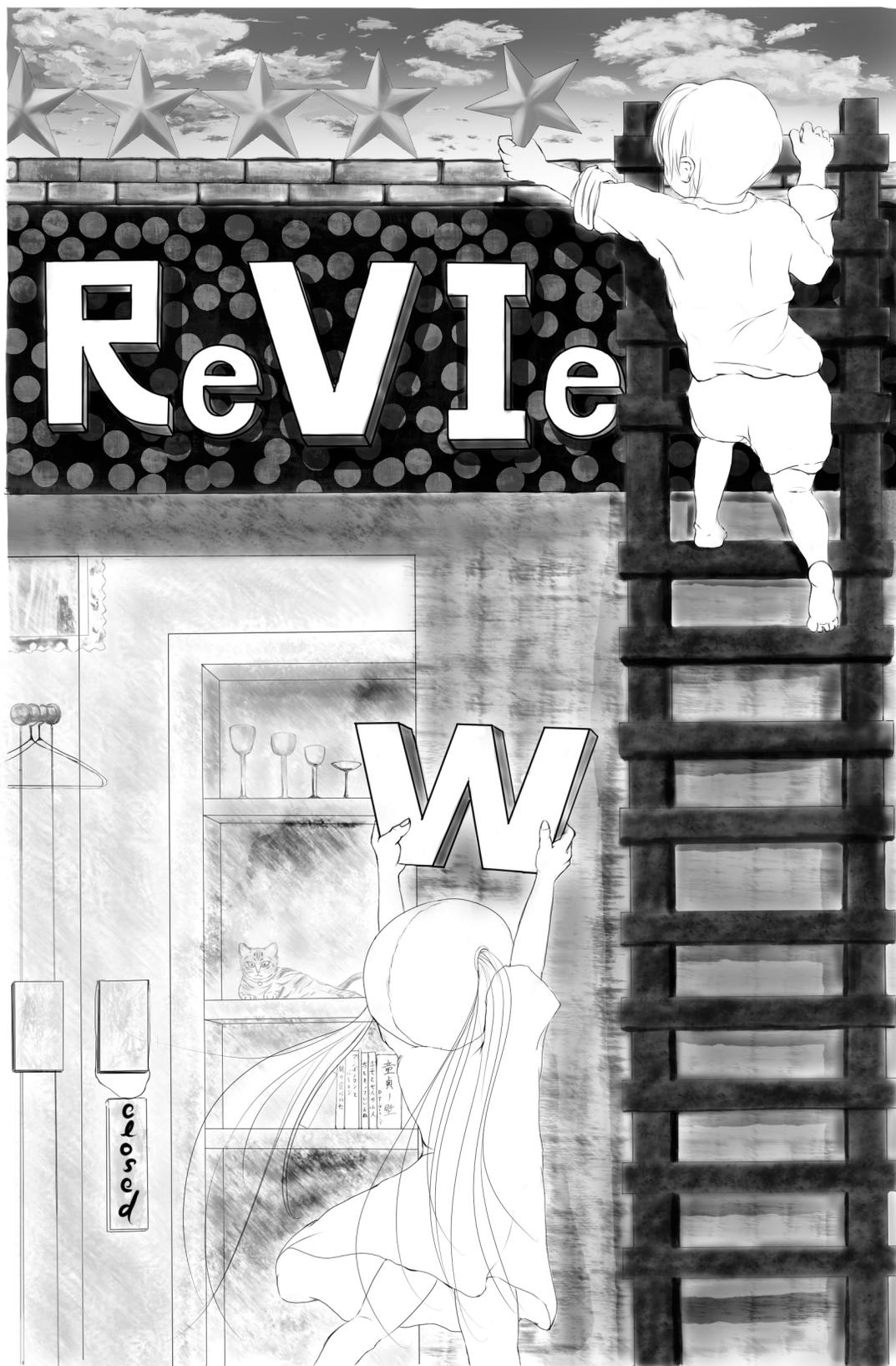
幾度か素材作り直したりとかしましたが、多分今度の調布祭での発表が最後の更新になると思いますので、現状それに向けて作業を進めているところです。とりあえずは系統的に多少は楽しめるものにしたいなー、という感じでしょうか。後は未実装の機能を早く実装していきたいです。

とりあえず今回の反省を活かして次作るゲームはもうちょっと頑張りたいです。一度作ってみて改善すべき点なども大分見えてきたので、あとはもう少し面白げなゲームシステムを考えたい

と思います。でも次の作品を作る時間あるのかな……。

\*2 CAVE が 1998 年に発表した 2DSTG の傑作。現代と殆ど変わらない程度の技術の世界に超能力者を取り入れた世界観が人気の作品。





## 日常の魅力

13l @hamo\_nik

現代の日本人で漫画、アニメまたはライトノベル<sup>\*1</sup>といった作品を見たことのない人はたしてどれくらいいるでしょうか？少なくとも今の10代～20代の日本人でこれに当てはまる人はごく少数に限られるでしょう。アニメで例を挙げるならば、国民的アニメである「サザエさん」や今でも根強い人気を誇る「プリキュア」<sup>\*2</sup>、などなど日本には数多くのアニメや漫画、ライトノベルの作品が存在します。今回はその中でも日常系と呼ばれるジャンルについての説明と日常系の作品のレビューを書きたいと思います。

まず、日常系とはどのようなジャンルなのか？広義的には日常系とは登場キャラクターの日常生活や、やり取りを描いた作品だと言われています。たとえば、世間で一大ブームとなった「けいおん！」<sup>\*3</sup>や国民的作品である「ちびまる子ちゃん」も日常系の作品として挙げることができます。また、主な登場人物が小学生～高校生といったことが多いことも特徴として挙げるすることができます。

さらに、日常系作品は2つのタイプに分けることができます。1つは主にギャグを重視している作品、もう1つはキャラクター同士の他愛のないやり取りや作中の世界の雰囲気重視している作品です。漫画を例に挙げると、前者は「生徒会役員共<sup>\*4</sup>」「男子高校生の日常<sup>\*5</sup>」、後者は「ひだまりスケッチ<sup>\*6</sup>」「らき☆すた<sup>\*7</sup>」などが挙げられます。

---

**注釈** ※作者敬称は省略しました。

<sup>\*1</sup> 一般的には美少女キャラクターや漫画のイラストや挿絵を多用し、キャラクター主体で作中での世界観が固定されている小説のことを指します。

<sup>\*2</sup> 2004年の「ふたりはプリキュア」から続いている女兒向けアニメシリーズのこと。そのキャラクターや内容から大人でも楽しめる作品となっています。

<sup>\*3</sup> かきふらいによる四コマ漫画作品、およびそれを原作としたアニメ作品のこと。2009年には第一期が、2010年には第二期が放送され数多くのファンを得ました。

日常系作品のよい点としては、まずキャラクターの人物像を把握していれば、ある1つのストーリーだけを読んでもその内容が理解しやすい場合が多いということが挙げられます。これは、日常生活という括りの中で、それぞれの話に深い関連性があまりないからだと考えられます。また、ストーリー重視の作品と比較して深い恋愛要素やシリアスな展開があまり含まれないため、日頃のストレスを解消するために見るという人もいます。

一方、日常系作品の問題点としてはストーリー性がない、または弱いことが挙げられます。これは、日常系作品の性質というものを考えると仕方のないことではありますが、ストーリーを重視する人にとってはつまらない作品とみなされてしまうでしょう。また、近年では日常系の作品の中でも安易な萌えを量産する作品が増加しているために、日常系作品＝萌え作品というレッテルを貼られてしまうことも問題として考えられます。

しかし、僕は日常系というジャンルが好きです。バトルものや SF も好きですが、日常系作品の空気や日常的なギャグ、さらに没個性的なキャラクターでもそのキャラが他のキャラとコミュニケーションを交わすことでその個性を感じることができる点に魅力を感じるのです。

---

\*4 週刊少年マガジンで連載中の氏家ト全による四コマ漫画作品。桜才学園生徒会の人間が繰り広げる学園コメディ。下ネタが多いのにエロくない、不思議です！

\*5 ガンガン ONLINE で連載されていた山内泰延による四コマ漫画作品。男子高校生の日常を描くドタバタコメディです。

\*6 まんがタイムきららキャラットで連載中の蒼樹うめによる四コマ漫画作品。ひだまり荘の住人が繰り広げる楽しく賑やかな日常が描かれています。

\*7 コンプティークで連載中の美水かがみによる四コマ漫画作品。陵桜学園高等部の女子高生の日常生活がゆるゆると描かれた作品です。埼玉県の鷲宮神社はらき☆すたの聖地\*8として有名です。

\*8 アニメや漫画に縁のある土地のこと。その愛好家が実際に聖地を訪れることを聖地巡礼といい、多くの自治体が町おこしの一環として彼ら呼びこむべく様々な取り組みを行っています。

次に、僕が見た日常系作品についてのレビューを書こうと思います。対象は漫画作品、作品は「ゆるゆり」「スケッチブック」「キメらるクラブ」「坂本ですが？」の計4作品です。

### ○ ゆるゆり

**あらすじ**：主人公の赤座あかりと彼女の幼なじみの歳納京子と船見結衣は元茶道部の部室を不当に占拠し、ごらく部と称した部活を行っていました。そこに茶道部に入部することを希望していた吉川ちなつも加わり、時には生徒会もの女の子達も加わりながら今日もゆるい日常生活が繰り広げられるのです。

**レビュー**：タイトルにも含まれる通り百合要素を全面に押し出した作品です。とは言えども内容的には軽い百合なので、普通の漫画としても読めます。そしてキャラの一人一人がかわいいです。女の子達がイチャイチャしてる所を見たい人におすすめです。

### ○ スケッチブック

**あらすじ**：高校に入って半ば強制的に美術部に入部することになった梶原空。今日も彼女やその他の美術部員達のまったりした日常が繰り広げられます。

**レビュー**：脱力系四コマ漫画。日常がふんわりとしたタッチで描かれています。実際に読んでみると身近な笑いやシュールなネタに思わずクスリとさせられました。美術ネタは少なめ。個人的には日常系のまさに日常を描いた作品だと思います。ほのぼのさを求めるならおすすめです。

### ○ キメらるクラブ

**あらすじ**：生徒会から部活動の合併を命じられた天文部の部長阿久野真緒。その合併先のオカルト研究部の部長丘路子、捏造部の部長螢所あい子、探求部の部長松戸さえの4人を中心としたドタバタ学園コメディが始まります。

**レビュー**：ドタバタコメディ。基本的には真緒以外の人間がボケて、真緒がつっこむといった構図です。効果的にシュールな表現が用いられることで読者の笑いを誘っています。路子の普段の厨二病<sup>\*9</sup>を発症している時とうろたえた時に素に戻った時とのギャップがかわいいです。

○坂本ですが？

**あらすじ**：入学早々、クラスのだけではなく学校中の注目を集める一人の男子生徒がいました。彼の名前は坂本。そんな坂本は今日もスタイリッシュかつクールに学校生活を送ります。

**レビュー**：普通の人には誰も真似できない、というか真似しないようなことを坂本は平然とやるのでクールな行動なのにかえってシュールさを際立たせています。坂本を取り囲む人たちの反応もかえって作品をシュールにする、そんな笑いがいっぱい詰まった作品です。

以上が、日常系漫画作品のレビューでした。ここでは紹介しきれなかった日常系作品はいっぱい存在します。もちろん、無理に日常系作品を見るとはいけません。しかしこの先、日常系作品に触れる機会があってこのレビューが参考になったら、僕としては嬉しい限りです。最後にフランスの写真家の名言を借りて文章を締めたいと思います。

**充実した一時間は忘却と不注意の数世紀より価値がある。**

—イリナ・イオネスコ

---

<sup>\*9</sup> 別名中二病。中学二年生で発症することの多い思春期特有の思想や行動のことを揶揄したネットスラング。症状としては自分を特別な存在だと思ったり、カルト的な存在を崇拜したりと自己顕示欲や自己陶醉が表面化した行動や思想が挙げられます。

# モバマスニツキ

11I 天麩レ

## 第一部 もばます道中 ～僕がプロデューサーになった理由～

興味ない人は第二部へ

### あらすじ

僕は大学に通うごく一般的な男の子。強いて違うところをあげるとすればモバマス<sup>\*1</sup>の興水幸子ちゃん<sup>\*2</sup>に興味があるってとこかな——名前は天麩レ

「興水幸子の CD デビュー<sup>\*3</sup>が決まったんだよ！」

Ω「な、なんだってー！」

こうして僕の生活は変わることになった……

という訳でネット巡回してて気に入ったキャラの CD が出るという事でモバマスを始めてみることにしました。<sup>\*4</sup>(´\_ゝ`)フーンって感じで見てもらえれば幸いです。

### 開始

5月28日にCDを購入する……が、うっかり始めるタイミングを見失ってしまい、寝落ちして一日遅れでゲーム開始。初日から躓く始末。最初に幸子が出てくる埼玉エリアのクリアを目標に設定しました。続けるかどうかはそれから考える、と言った気持ちで始めました。

始めにゲームを進める上で行うことが「お仕事」です。「お仕事」とはアイドルと日本各地に出張し、そこでアイドルの勧誘をしつつ仕事を進める、所謂クエストと呼ばれるようなものの一つです。モバマスはプレイヤーそれぞれに「スタミナ」と「攻コスト」、「守コスト」という3つのパラメータがあり、「お仕事」ではこの内、スタミナを消費してゲームを進めます。因みに3つのパラメータは時間経過で回復します。しかし慣れぬ新米プロデューサー、数回お仕事を進めるだけでヒヒイ言い出す始末。辛い。

5日目にはサークルの人達で構成されたプロダクションに入れてもらうことに。なにこのひとたちめっちゃ強い<sup>\*5</sup>。プロダクションに加入すると大量の振り分けポイント<sup>\*6</sup>が取得出来るので、これのお陰で以降のプレイが大分捗りました。

### 強アイドル交換の罫と初めてのイベントへ

10日目くらいにはそろそろ強いアイドル欲しいなーと思い、サークルの先輩からアドバイスを受つつアイドルを購入<sup>\*7</sup>。しかし、それはあまり強くなかった<sup>\*8</sup>という罫が……。良い勉強になりました。

この辺りで初めてイベントに参加。先輩にそこそこ強いアイドルを借りつつぼちぼち参戦しました。モバマスでは頻繁——というかほぼ毎日イベントが開催されているのでそれをやるとわりと飽きない感じです。チーム制なので他の人が強くてなんとか報酬ゲット。おおうまいうまい。

<sup>\*1</sup> モバマスとは、Mobage で好評配信中のソーシャルゲーム（以下、ソシャゲ）、アイドルマスターシンデレラガールズの略称。

<sup>\*2</sup> モバマスに登場するキュート属性のキュートなアイドルの一人。So Cute.

<sup>\*3</sup> 要はキャラソン CD が出るということ。決して同名の人物が居るわけではない。

<sup>\*4</sup> CD にはゲーム内カードが付属するので、CD を買えば欲しいキャラが簡単に手に入るのです。

<sup>\*5</sup> 「あ、この人こんなにやってるんだ……」と思う事多数。G 氏とか。

<sup>\*6</sup> 各種パラメータを上げるポイント。お仕事を一定量進めたり、プロダクション加入によって手に入る。

<sup>\*7</sup> 購入といってトレードしただけです。モバマスでは自分の手持ちのアイドルをバザー形式でトレードする事が可能です。ぶっちゃけ人身bおっと誰かが来たy

<sup>\*8</sup> 俗にいう MM 強化、ではなかったという事。要は成長で手を抜いたものという事です。

## そして埼玉へ……

開始1ヵ月くらいでやっと目的地埼玉へ。ちょっと寄り道し過ぎた感じ。多分イベントとか参加しなければもっと早かったと思います。ぐぬぬ。

この辺りで最初にCD購入で手に入れた幸子がレベルMaxに\*9。ひゃっほい。

2回目のイベントにも参加しますが、立ち振る舞いが分からず悪戦苦闘。前からやっている人とそうでない自分との差を感じます。

そして7月11日、ついに目的の埼玉エリアクリア。約1ヵ月、長くて苦しい戦いでした。くう疲。

## → (つづける) (がんばる)

一応目的は達しましたがそれ以降もプレイを続けています。その後は情報を集め直しつつ、戦力を整え、先日念願の別のSR+幸子を手に入れました。まあまだ上に2枚あるのですが……。

所感ですがこのゲーム、戦略が必要だなと感じました。既に開始から約2年が経過しているため、強アイドルやイベント等での立ち回り方等はだいたいネットに転がっています。ですのでちゃんと調べつつ、方針に従って進めていけばそれなりにサクサク進むのではないかと。少なくとも私のようなスローペースではなく。

遊んで一番思ったのは、実はこのゲームはお金よりも時間を食うゲームだということでした。少しガチャ回すだけの微課金ユーザだからという事もありますが、それでもアイテムが少ない初期では中々お仕事を進める事も出来ないため、こまめにプレイして進めるしかありません。多分そういうゲームなんだろうな—と思いつつも、お金同様時間も有限リソースですから辛いところではあります。

とは言え時間とリアルマネーが物を言う世界ではあるので、時間をかけただけではやはり限界はあります。流石に月n万円(n>0)を入れているユーザには勝てる筈ありません。ですがトレードがあるので、無課金でもある程度の戦力を整えてゲームを楽しむ事は出来ます。気に入ったら少しずつお金を入れていく……くらいが一番楽しめるのではないのでしょうか。

とりあえず気軽に遊べて、自分の経済状況、時間をよく考慮すれば楽しくプレイできるゲーム、というのが感想です。ああ次は天使幸子だ……。セクシー幸子は……無理。

## 第二部 魁！！ 考察塾

ここからはモバマスや別のソシャゲとかそうじゃないゲームを含めてソシャゲについての考察を少し。感想も絡めつつ、どこが面白いのか、少し考えてみたいと思います。私自身まだあまりソシャゲについてはプレイ出来てないので、少し遊んだ人間の感想と捉えて頂ければ幸いです。

### なぜ流行ったのか

結論から先に述べると、流行るゲームには面白い、と感じる幾つかの共通事項を感じました。それ自体だけでは到底面白いといえるものではありませんが、それが基礎になってゲームを支えている印象があります。それらがしっかりしているからこそ話題を呼べるのであり、それらが疎かになっていると何か特徴があってもどこか目立たない作品となってしまうのではないのでしょうか。

### モバマスで考える

モバマスというゲームの何が面白いのかというところを考えると、個人的に楽しいと思ったのはイベントの参加と、後はレベル上げやアイドル集め等の単純な事でした。

レベル上げやキャラクター収集はコンシューマゲーム(特にRPG)ではも見られるようなものですがそれらと何ら変わりないと思います。単純な作業の楽しいところの一つは、高い頻度でやってくる小規模な達成感だと思います。レベルが目標に達したり、欲しいキャラクターが手に入るというのは、一つの目標を達成する訳ですから達成感はあるでしょう。

強いて言うなら、このゲームでは比較的レベルが上げやすいのでそういった喜びが感じられや

\*9 ちゃ、ちゃんとMM強化でMaxにはしましたよ。

すい、というのがあります。つまり短いスパンで小規模の楽しい事を与えることで全体として楽しい、と感じられるものになっているのではないかと思います。

これに関連する要素の中で一つ重要なものとして、**数値と資本のインフレ**がありました。モバマスではアイドルが手に入り易かったり、レベルアップが早いという事があります。また格差も大きく、弱いアイドルと強いアイドルでは10倍以上の差が出てきます。これは謂わば1種のインフレであり、プレイヤーに自分がどんどん強くなっていくことをより強く感じさせることができます。プレイヤーに分かりやすい派手なインフレが、攻撃結果や報酬などの行動結果に対して使われると、プレイヤーは爽快感を覚えやすいでしょう。

これらの事は同じ会社で開発されている「神撃のバハムート」\*10（以下バハ）ではより顕著に感じられ、取得カード数、そしてレベルアップのタイミング、どちらもモバマスの数倍の量でやってきます。例えば先ほど説明したイベントでの入手カードについてですが、モバマスに比べてバハではかなりの回数を引くことが出来ます。モバマスなら10枚くらいカードを貰うところでも、バハなら同じくらいの時間で二倍以上のカードは出来ると思います。多少作業が多かったり報酬の質が悪くても、数で数えられるものというのは非常に分かりやすい要素であり、そういった資本インフレを起こすのは効果的な手法だと思います。こういった小規模な達成感を頻繁、且つ上手に使っているのがソシャゲだと思います。……やり過ぎはプレイヤー離れを引き起こすとは思いますが。

またモバマスとバハ、そして他の多くのソシャゲにも言える重要な要素として、ストレスを感じにくい手軽さ、というものがあると思います。ソシャゲは基本的には非常にシンプルに作られており、複雑な要素、システムは基本排除される姿勢にあります。だからこそ多少面倒な操作も耐えられる、というのはあるのではないのでしょうか。

## 突如として現れた CookieClicker

実は少し前に話題を呼んだゲームが、こういった要素を上手く使っていました。9月15日頃に**何故か** Twitter 上で話題になった **CookieClicker** \*11 というゲームをご存知でしょうか。私も話題を聞いて見つけてよく分からないうちにハマってしまい、その日は一日何度もクッキーのサイトを開いては操作する次第でした。非常にシンプルなゲームなのですが、何でこんなにも話題を呼ぶ人気が出たのかということ、先ほどの数値、資本のインフレが一つの大きなポイントだと思います。



ね、簡単でしょ？

このゲームは画面左のクッキーをクリックするとクッキーが1枚手に入り、そのクッキーを使って自動でクリックしてくれるカーソルを増やしたり、クッキーを焼いてくれるお婆さんを増やしたりして、クッキーを量産して施設を増やしていくゲームです。クリックして最初に手に入る量は1枚ですが、お婆さんを増やせば2秒で1枚、工場を増やせば秒間10枚と、プレイしているとあっという間にクッキーが増えていくようになります。これは単純に取得量がどんどん増えていくのでプレイヤーには直感的で分かりやすく、且つこのクッキーで新たな設備を増やせるのでプレイヤー側としては分かりやすい。このシステムのお陰で、このゲームは数値インフレと資本インフレを同時に提示していることになります。また一つ上の段階の設備を入れると分かりやすいレベルで一気に取得数が増えるのも大きなポイントでしょう。数値だけではなく、感覚的にも分かりやすいレベルなので2度おいしく感じられるという訳です。

\*10 URL - <http://shingekinobahamut.jp/>

\*11 URL - <http://orteil.dashnet.org/cookieclicker/>

また CookieClicker もモバマス、バハ同様に、手軽にプレイ出来るということが念頭に置かれて作られています。特にこちらはスコアと資本がクッキーという共通のもので作られている為、考え方こそ違えど、クッキーの枚数を増やすことがプレイヤーの目指すことであるため、プレイヤーとしては目標がとても分かりやすい特徴を持ちます。また CookieClicker が特に顕著ですが、放置しておくだけで進められるという手軽さは重要です。このシステムが CookieClicker の面白さを決定づけていると言っても過言ではないでしょう。ブラウザで遊べるというのも、手軽さを表していることのひとつだと思います。

## でも本当にそうなんです？

以上の事を纏めてみると、上に挙げたゲームの特徴は、

- プレイヤーに達成感を与えるシステム
- 感じ取れる数値、資本のインフレ
- ストレスの感じにくい手軽さ

となります。

挙げたのはゲームの面白さの一部——あくまでも基礎部分的なものです。面白いと感じる要素なら別なところにもっと色々あるでしょうし、これが面白さの全てかと言われるればほぼ間違いなく違うでしょう。例えばモバマスならイベントをこなしたり、アイドルを育てていくのが楽しいと思いますし、大ヒットしたパズルアンドドラゴンズ\*12 なら言うまでもなくそのパズルゲームででしょう。CookieClicker であればどのようにして効率良くクッキーを増やすかという戦略性等、上げればキリがないと思います。

またこれらの要素はソシャゲに限った事ではありません。例えば ACT なら大抵どこかに難所とか山場と呼ばれるところがあり（ボス戦とか）、それをクリアすることで達成感を得られるでしょうし、RPG ならドロップアイテムで射幸心を煽ります。インフレも具体例としては STG の怒首領蜂シリーズではよく表れており、HIT 数やスコアで数値のインフレ、取得アイテムのジャラジャラ感で資本のインフレがよく表されています。ストレスレスな設計は言うまでもありませんね。上の要素というものは昔から他のゲームで使われているものであり、また逆に言えばソシャゲだろうが何か新しいゲームだろうが、そういった古くからあるゲームの基本的なセオリーは踏襲しているという事です。

つまりソシャゲを支える要素というのはもっと一般的なものであり、何も射幸心を煽る事が全てとか、そんな事ではないと思うのです。繰り返しになりますが、別に上に挙げたことがあるから流行る、そして面白いという事を言うつもりはありません。ただゲームを作る上で気をつける事として、まず流行っているゲームを複数見て、そこから共通点を見て参考にすることができるのではないかと思います。共通点を取り入れていき、取り入れられそうにないなら別の視点で見ることで、自分の作るゲームの参考になるのではないかと私は思います。

え、私は出来てるかって？ 出来たら面白いゲーム作れてるでしょうね……（遠い目）

# やり過ぎに注意して楽しく遊みましょう！！

艦これ？ ああ BGM いいですよ（遠い目）

\*12 URL - <http://www.gungho.jp/pad/>

# 秋葉 来たの限界を知る

4629

自転車でもどこまで行けるか挑戦する、インドア学生による体温まるアウトドア企画です。

## 1. 動機

大学から横須賀中央駅まで何 km あるんだろうとちょっと気になった。Google 先生によると約 50km。50km って自転車で行けるか行けないかのギリギリの距離だよね。せつかくの夏休みだし、挑戦してみよう……的な。

## 2. 概要

いつも乗っている自転車を使って遠出してみました。前哨戦の「調布～秋葉原編」と本編の「調布～横須賀編」について、当日の様子を振り返りながら書いていこうと思います。ルールとマナーを守って楽しくサイクリング！

## 3. 運動プロフィール

高校の頃は片道 8km の平坦な道を自転車通学していました。しかし文化部で、それ以外の運動は皆無でした。最近では流石にマズイと思い、DDR<sup>1</sup>で定期的な運動を始めました。

1 DDR……DanceDanceRevolution。ダンスする音ゲー。疲れる。

## 4. 調布～秋葉原編

前哨戦なので、写真も無しにさっくりと書きます。(撮り忘れてました)

Google 先生によると電気通信大学から JR 秋葉原駅まではおよそ 23km。往復すれば約 50km なのでこれができるれば横須賀も行ける……筈。

①旧甲州街道(国道 20 号)を新宿方面に走って行くと半蔵門(皇居前)まで着く。

②そこから皇居を時計回りに回って靖国通りに出る。

③後は、道なりに秋葉原周辺まで行ける。

①がとにかく長いです、苦痛。

### 8/24 調布～秋葉原編当日

とりあえず西友で 1L 分の飲み物を買ってから大学の正門へ。

9:15 電通大正門からスタート！

道路の案内には『新宿 15km』の絶望の文字が！

10:00 明大前

京王線を使えば、調布から明大前まで 10 分強で行けるのですが、自転車だとその 4 倍の時間もかかってしまいました。仙川あたりで長めの上り坂もあり、このあたりで飲み物を半分以上消費してしまいました。

## 10:16 東京オペラシティタワー

調布から一時間で行けるのがここ。

## 10:28 新宿ルミネ前

明大前から意外と時間がかかってしまいました。ここまでで、73分経過。歩行者が多いので少し押して歩くことに。

## 10:44 JR 四ツ谷駅

意外と新宿からは時間がかからなかったです。

## 10:52 半蔵門

初めて生で見る皇居。ここに小さな公園があったのでトイレを済ませたり、水道水を飲んだりしました。

## 11:26 秋葉原到着

靖国通りあたりで迷ってしまい10分くらいロスしたものの、なんとか到着することが出来ました。累計131分。時速換算で10.5km/h。意外と時間がかかってしまいました。

## その後

秋葉原の有料駐輪場は1日停めっぱなしでも100円という良心的な所が多いです。

適当に停めた後、部長氏とチャーハンを食べたり、ビデオカメラのケーブルを探したり、キーマニ<sup>2</sup>をやったり

2 キーマニ……Keyboard Mania。ピアノの鍵盤の音ゲー。敷居がかなり高い。

してのんびりと過ごしました。

帰りは同じ道でしたが、150分かかりました。

ちなみに部長は電車で来た。ぐぬぬ。

## まとめ

電車代が惜しいってだけで秋葉原まで自転車で行こうって考えている電通大周辺に済んでいる諸君！後悔するぞ！

ジュース代等で結局チャラになりましたとき。

## 5. 調布～横須賀編

スタート地点は前哨戦と同じく。電通大正門。ゴールは横須賀中央駅。



Google Mapより

① 狛江市から多摩川を超えて、登戸駅を目指す(ここまで5.2km)。

② GoogleMapを最大限活用し、気合で新横浜駅を目指す(ここまで20.6km)。

③ 横浜駅を目指す(ここまで26.1km)。

④京急に沿った道路を走って金沢文庫駅を目指す(ここまで41.7 km)。

⑤国道16号(横須賀街道)で横須賀中央駅を目指す(ここまで52.0km)。

前哨戦との主な違いは、

- 1.ひたすら体力勝負
- 2.知らない道(特に登戸～新横浜間)なので、道を確認しながら進む必要がある
- 3.山道が多い
- 4.泊まる用意があるので荷物がやや重い

と言った点でしょうか。小さな山を登っては降りての繰り返し、神奈川に入ってから何回もあり体力を大きく削られました……。

途中でチェーンが外れたり、タイヤがパンクしたり、雨が降りだしたりしたら即中止にしようと考えていました(新横浜くらいまで来たら引き返すという選択肢が無くなりますが……)。

**9/21 10:37 電通大正門からスタート!**



とりあえず布田駅から、渋谷街道に沿って狛江市を目指します。この時

点では5時間で横須賀中央に着いたらいいな、と考えていました。

**11:06 多摩川を渡る**



さらば東京。

**11:16 登戸駅到着**



そこそこ大きい駅です。さて、ここからが地獄の始まり……。

ここからかなり迷います。とりあえず、GoogleMapを参照しながら行けば大丈夫……と思っていましたが、気がつくと、その道を外れていて、いつの間にか府中方面を目指していたり……と一筋縄では行きません。

ちょっと進んでは、GooglMapで確認していたので、端末のバッテリーが30%を下回る事態に。この時は、最悪死を覚悟していましたね……流石

にそれは大袈裟ですね、はい。

## 12:47 東山田ゆうやけ公園



実は本来のルートならば通ることの無い道なのですが、戻るのも面倒なくらい進んでしまったので、とりあえず見つけたこの公園で小休止することに。ちなみに、ここは調布から13.2kmの地点。あんまり進んでいないですね。

## 13:33 新横浜駅到着



途中、新横浜駅に向かうバス停を見つけることが出来、ひたすらその道を進むと大きなスタジアムの前に着きました。日産スタジアムというサッカー場みたいです、ここまで来れば、新横浜駅もすぐでした。

駅近くのマクドナルドで無料券を使

い、昼食を済ませます。

## 14:27 横浜駅到着



ここまで来れば、後は京急に沿って進むだけ……ウイニングランだ！ と思っていました。ちなみに、ここまでで3時間50分経過しています(時速換算で6.8km/h)。

## 14:48 とんかつのころ



というお店があるみたいです。特に深い意味は無いです。ちなみに、帰りも同じ店を別の場所で確認しました。チェーン店だったのか(井之頭五郎風に)。

この付近にあるマルエツで1L分の飲み物を買いました。

## 15:57 金沢文庫駅到着



とにかく長い道だった気がします。おそらく22号を走ってきたようなそうで無いような……。ここからは、見慣れた道、見慣れた景色なので精神的に楽になります。あとは横須賀街道をひたすら行くだけです。

## 16:03 開かずの踏切(金沢文庫～金沢八景間)



テレビで報道されるくらい有名な踏切<sup>3</sup>。とにかく開かないです。10分待たされるくらいで済みました。デレた。

ここからすぐのダイエーでジュースを買ったり、jubeat<sup>4</sup>やリフレク<sup>5</sup>をしたりして45分ほど潰しました。余裕モードですね。

3 今はそうでもないみたいです。

4 jubeat……音ゲー。16パネル。楽しい！

5 リフレク……REFREC BEAT。音ゲー。はじく！楽しい！

## 17:50 横須賀市街地！



いかにも横須賀な通りに来ました。ここまで来ればゴールはすぐです。なお、これより前に京急汐見駅あたりで、15分くらい潜水艦の写真を撮ってて時間潰していました。

## 17:55 ゴール！



京急の駅のフォント好きです。ということでゴールしました。

## まとめ

7時間18分の自転車の旅は終了しました。1時間遊んでいた時間を引いて、時速換算すると8.2km/h。調布～横浜の時よりも速いのは、横浜以降特に迷うことなく進めたからだと思います。

もう二度とやりたくないです。

## おまけ

行きもあれば帰りもあります。写真は無いのでダイジェストだけでお送りします。だいたいは行きに通った道を使う方針です。

9/23 10:10 横須賀中央駅

11:40 上大岡駅

12:20 横浜駅

ここまではかなり順調でした。

ここで、新横浜駅に向かえばいいものを、途中で謎のショートカットを試みようとして、横浜国立大学方へ……。このまま道を間違えていれば横須賀方面へ逆戻りしてしまうところでした。1時間くらい無駄にしてみました。

どうにか新横浜駅まで戻ります。ここから、どうやって帰ろうかなあ、と思った矢先道路標識に『登戸18km』の文字が。登戸と言えば、行きの最初のチェックポイントです。神奈川県道13号をひたすら進んで行きます。行きの時に、この道に気づけていれば……。

途中、マルエツ川崎宮前店で昼食も兼ねて寄ります。またマック……。ここまで来れば登戸駅はあと4kmです。

また、小さなゲームコーナーにポップン17<sup>6</sup>が1プレイ50円で置いてあったので、すかさずやりました。

## 15:55 ゴール！

行きと比べると1時間弱くらいは時間を削減できる結果になりました。ですが、途中で迷って無ければあと30分早く帰ることが出来た気がします。

## 6. 終わりに

道が分かったので次回以降は迷わずスムーズに行けるかと思われれます。やる機会があるかは分かりません。

何よりも、山道がきつかったですね。

これ以上の距離の挑戦とかは……あんまり考えたくないですね、はい。

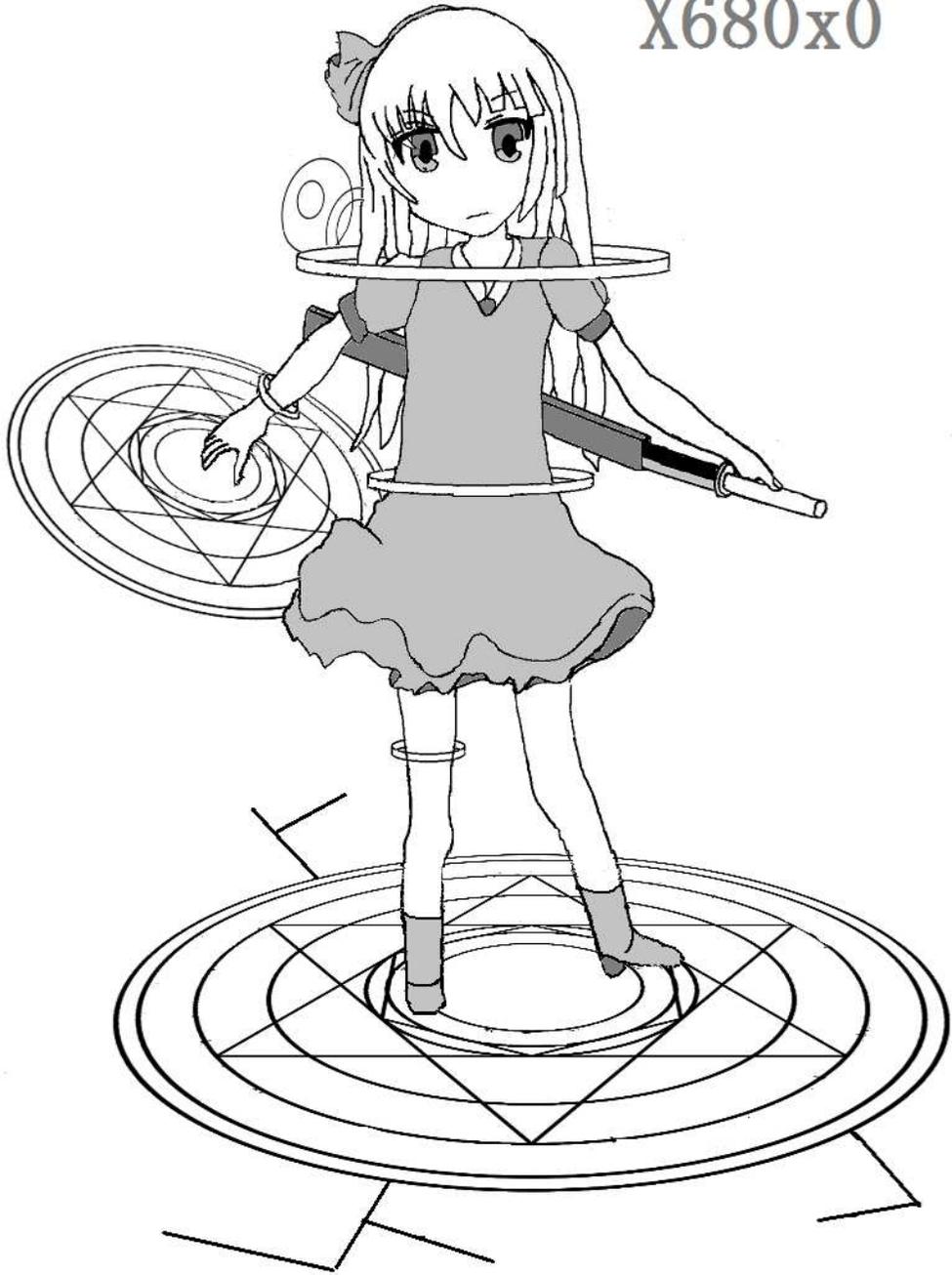


行きの終わり際に撮った横須賀の好きな景色の1つです。(汐入駅周辺ダイエー)

壊れなくてありがとう、自転車。雨振らなくてありがとう、お天道様。好きです、横須賀。

6 ポップン……pop'n music。音ゲー。17は最新作の4作前。

X680x0





knowledge

Android向けミニゲーム制作ライブラリ

# Amanatsu を作ってみた

Hiroki

これは Android 向けの OpenGL を用いたゲーム制作ライブラリ Amanatsu の開発記事です。

## 1 Mikanっぽいライブラリ

Android でゲーム開発をしようと思った時に立ちふさがるのは、端末ごとの性能差や GC<sup>1</sup> 描画方法だったりします。X680x0 同好会でも何度か簡易講座をやってみたものの、あまり普及しませんでした。そもそも私自身も毎度毎度 FPS<sup>2</sup> 制御や描画管理等を行うのは面倒なのでやりたくありません。やはり FPS 制御をざっくりとやってくれ、準備フェーズなどが用意されたライブラリが欲しいわけです。そんな気軽に使えるのがない？ならば作りましょう。それが開発者というものです。

そんなわけで、今回は Amanatsu<sup>3</sup> と名付けた Android 用のゲーム制作ライブラリを作ることとしました。このライブラリは私と何人かの協力者で作っている PC 向けゲーム制作ライブラリの Mikan<sup>4</sup> 系列の新ライブラリという位置付けで、基本的な設計方針等を取り入れていきます。

しかし、ライブラリを作ると言っても、

最近 Android 版もフリーになった Unity や 2D ゲームを作成する Cocos2d-x などの高機能なゲーム制作ライブラリがたくさん出てきています（正確には、C++ を用いたクロスプラットフォーム向けの開発環境）。ただゲームが作りたいなら、そこら辺の高機能で流行ってるものを扱えるようにした方がいいので、独自ライブラリを作る前にコンセプトを明確にする必要があります。

そこで初めにコンセプトを設定しました。今回のコンセプトは単純明快、たった 1 つです。

- Java のみで使える。

Java のみで使えるとは、JAR<sup>5</sup> 単体で読み込ませるだけで使えるようにするという事です。なぜこれをコンセプトにしたかといえば、最近の Android の開発環境事情が関係してきます。スマートフォンは高機能・高性能なため、最近ではお絵かきや作曲のみならず、アプリ制作や DAW による音声加工も可能になってい

<sup>1</sup>不要になったメモリを開放する処理。

<sup>2</sup>1 秒間に何回画面を更新するか単位。

<sup>3</sup>Mikan 系列のライブラリには柑橘系の果物の名前を与えることとしました。今回は Android の頭文字である A を含む甘夏を採用しました。

<sup>4</sup>C++ 向けの DirectX を用いた PC 用ゲーム制作ライブラリ。詳しくは 2011 年の 68 通信 (<http://www.x68uec.org/other/press/2011/>) や MikanWiki (<http://mikan.azulite.net/>) を参照してください。

<sup>5</sup>Java のアーカイブで Java の class ファイルや画像などのリソースを ZIP 圧縮したもの。



ないと端末の方向を変えただけでゲームが再起動してしまったり、ゲームを別のアプリに切り替えると終了してしまうこともあります。そのため、このライフサイクルは絶対に守られる類のものではなく、ゲームを作りやすくするものと割ってきて扱う必要があります。

### 3 描画

描画には OpenGL を採用しました。Android には Canvas、SurfaceView、OpenGL の 3 つの描画方法が提供されています。Canvas は簡単ですが描画が重く、SurfaceView と OpenGL は描画の処理が面倒です。SurfaceView は描画能力は Canvas と同程度のようなので、同じく面倒ならより描画速度が早く 3D も扱える OpenGL<sup>6</sup>を採用することとしました（DirectX で得た知識という資産を扱えるメリットもある）。ゲームライブラリの作成においてこの描画は最も重要な部分となります。使い方は Mikan ライブラリと似たような使い方になるようにしつつ、Mikan ライブラリの引数の不具合を修正することとしました。また、Java や OpenGL 特有の問題も多々発生したので、いくつか重要な部分について書いていきたいと思います。

#### 3.1 引数

実際の描画を行う引数は画像の番号や読み取り開始座標、大きさ、描画先座標

<sup>6</sup>グラボを使って 3D や 2D 描画を行う API。Android 等の携帯機器に搭載する場合用の組み込み向けの OpenGL ES が利用されている。

<sup>7</sup>実際の利用者の声

や描画情報を扱う必要があります。かなり多くの値を扱うことにはなりますが、画像の一部分を切り取って画面の任意の場所に貼り付けるにはこの程度の情報が必要となります。Mikan ライブラリでは読み込みと描画の引数関係がぐちゃぐちゃだったため、今回は読み込み情報を先、描画情報を後になるようきっちりと設計しなおしました。読み込み情報が先の理由は、読み込みは開始座標と大きさという要素が増えたり減ったりはしないからです。これにより多少は引数の与え方に秩序が持たされるようになりました。また、Mikan 系列のライブラリの特徴である、引数のパラメーターを増やすとより詳細に描画を指定できる特徴をより理解しやすくなりました<sup>7</sup>。

#### 3.2 画像の読み込み

グラフィックボード（以下グラボ）を使って高速描画するためには、グラボのメモリに画像のデータを転送する必要があります。そのデータの管理方法として、正方形かつ  $2^n$  の画像サイズでなければならないというルールがありました。最近のグラフィックボードで DirectX を使用する場合はそのような制限はなくなりましたが、Android の OpenGL ではその条件が必須となります。

このようなルールがあるため、ユーザーにその制限を課したいところではありますが、それでは知らない人がいつまでも正常に描画できないことが想定され

るので、とても不親切です。今回は無理矢理にでも描画できるように、必要があれば画像の大きさを Amanatsu 側で修正することにしました。今回はミニゲーム用のライブラリで画像の一部を切り取って使う利用方法ということで、拡大縮小は行わず、画像サイズから正方形かつ  $2^n$  の画像サイズに満たない部分を透明な色で埋め尽くすことで画像サイズを適正な大きさにします。

また、Android での画像などのリソースに関しては、2 つの保存方法があります。res フォルダ等の決められた場所にファイルを入れる方法と、assets フォルダを作りその中に入れる方法です。前者は画像ファイルが端末のサイズによって勝手にリサイズされてしまうので、assets フォルダに入れる方法を推奨することとしました (res 内の画像も扱えるようにはしている)。

ちなみに捕捉ですが、Android 搭載端末はハードウェアの個体差が非常に激しいです。もちろん OpenGL で扱える画像サイズの限界も様々です。そうなるほどの程度の画像サイズであれば扱えるのが非常に重要となりますが、Android の OpenGL では 1024 のサイズまでなら作れるようにすることと明記してあるため、基本的には 1024 までのサイズであれば作れると考えてよいでしょう。

## 4 入力と音声

ゲームライブラリでは入力や音声も重要な要素です。描画ほどではないですが

こちらもしろいろと面倒な問題が出てきます。

### 4.1 入力

Android の入力はタッチとその他センサーの 2 種類に分かれます。Amanatsu ではその他センサーに関しては傾きを検知する機能だけ管理しています。傾きは Android の API で Roll、Pitch などの名前で簡単に取得できる値でしたが、Android の端末と API がしっかりと連携が取れていない関係上、このやり方では端末によって得られる値が全く異なるという事態<sup>8</sup>になりました。そこで、少し面倒ではありますが磁気とジャイロセンサーから端末の傾き具合を検出する方法<sup>9</sup>を採用しました。これにより端末間の差異を考えなくてもよくなります。

タッチ入力についてはタッチしているフレーム数を返す管理機構を作成しました。これにより、より長くタッチしているなどの情報が入手可能です。一方マルチタッチを考慮した場合、指には ID が割り振られるため、それを元にハッシュで管理することでユーザーがマルチタッチでも個別の指を認識できるようにしました。その際、後述するメモリの問題もあるため、あまり大きくないハッシュサイズにしつつも、ハッシュテーブルの大きさが足りなくなって再確保するようなことがないサイズにする必要があります。こちらは指が 10 本までと仮定し、ハッシュテーブルの領域再確保の目安の数を調べた上で適切に設定を行いました (資料を

<sup>8</sup>多分開発会社の認識の違い。

<sup>9</sup>このやり方を推奨。少し面倒だが処理のサンプルはたくさん転がっている。

紛失してしまったが、確か現在のテーブルサイズの  $2/3$  を超えると領域の再確保が行われる。10本を上限として超えない事を考えると16が安全最小値となるので、Amanatsuでは16に設定している)。

ついでではありますが、Androidではキーボードやゲームパッドの入力にも対応しているので一応対応させておきました。キーボードの上下左右キーとゲームパッドの上下左右ボタンが同じ割り当て番号だったりとしおもしろい関係になっていたことと、Windowsの仮想キー割り当てのようにQWERTYキーの左上から順番ではなく、(プログラミングしやすいようになるのか)A~Zが連番になっていたことを捕捉しておきます。

## 4.2 音

音は3つの方法で再生が可能です。BGMのように長い曲を再生するMediaPlayer、効果音のように短い音を再生するSoundPool、そしてバイトデータを再生するAudioTrackです。初めの2つはAudioTrackを利用しているとのことなので、大元はAudioTrackのようです。しかし、AudioTrackは自分で音声ファイルの解析やループ制御等を行わなければならないので、とても面倒です。MediaPlayerは再生に遅延が生じたり、SoundPoolは長い音を再生できないなどデメリットはありますが、この2つを採用することとしました。特にSoundPoolは端末ごとの再生可能秒数が異なり1秒程度でないと安心できませんが、AudioTrackでの実装は時間もあまりかかたくなかったのでやめました。後日音について何とかしたい

と感じたら、本腰を入れてAudioTrackを用いた再生機構を作ろうと考えています。

## 4.3 GC対策

Androidでゲーム制作といえば、最大の敵はGCです。GCはメモリ管理機構のことで、メモリが圧迫されてきたら不要になったメモリ領域を使えるようにするありがたい機能です。最近の言語にはデフォルトで採用されている場合がほとんどです。この問題点は、このGCの作業が少し重いのと、アプリの実行を止めてしまうことです。ユーザーが書く部分に関してはどうしようもありませんが、ライブラリ部分はGCを積極的に起こさない工夫をする必要があります(少し話は変わるが、C#ではGCを任意タイミングで実行可能。そのような機構がある場合は、シーン遷移など少し止まっても問題ない場所でGCを積極的に行うという選択肢もある)。

では、GCを起こさないためには何をすればよいでしょうか？ 答えは簡単で、変数の使い回しを行うことです。入力のところではハッシュテーブルのサイズを最適に.....などと書いていましたが、これもGC対策の一環です。メモリの再確保=不要になるメモリ領域があるということですし、不必要に大きなテーブルを確保するとメモリが足りなくなってGCが発生してしまうので、極力それを避けるよう努力します。簡単な部分で言えば、ループ変数を全てメソッド内ではなくクラス内に定義します。こうすることでインスタンスを削除しない限りはメモリを不要に確保しなくなります。特にOpenGLで

は float 配列を 3D の描画に使いやすいように FloatBuffer というものに変換して扱うのですが、それも作った後は再利用します。具体的には値は put() メソッドで書き込み後は必ず position() メソッドでバッファの位置を 0 にします。また float 配列も予め必要個数を確保しておき、その配列にデータを書き込んでから FloatBuffer に put() メソッドで書き込みます。

こんな対策で効果あるのかと疑問に思う方もいるかもしれませんが、FloatBuffer に対してこの改善を行うだけで処理落ちが緩和されました。例えば 1 枚の画像を貼り付けたとき、座標、頂点色、UV 座標で FloatBuffer 4 頂点分使うので、少なくとも  $3 \times 4 \times 4$  バイトは消費することとなります。Amanatsu のデフォルト FPS が 30 なので、1 秒間に約 1KB は消費することとなります。実際にはもっといろんな場所でメモリを確保しまくっていると考えられるので、その数倍にはなるでしょう。携帯端末はそんなにメモリ容量があるわけではないのでそれだけのメモリ領域を毎秒確保していると、あっという間に GC が実行されてしまいます。アプリの実装言語である Java は C 言語などと比べてリッチではありますが、動かし環境が PC などと比べると貧弱ですので、メモリ管理は重要な要素となります。

捕捉ですが、Android のバージョン 2.3 以降では、平行 GC という GC が採用されています。この GC を採用しているバージョンでは GC によるアプリ停止がかなり緩和されています。しかし、ゲームでは ms 単位の処理の遅れが致命的であり、平行 GC を採用したとしても対策

なしでは処理落ちが多々確認されています。GC 対策は頑張りましょう。また、NativeActivity などを用いて Java の世界を抜けだして C++ で実装すれば、GC の影響下から逃げることも出来ます。恐らく他の C++ で Android ゲームを作れるライブラリは、これを利用していると思われるます。

## 5 その他

実装の大体の苦勞部分は終了しました。後はいくつか補足的な内容について書いていきます。

### 5.1 描画捕捉

Windows + DirectX と Android + OpenGL では画面の座標の設定方法が異なります。具体的には Windows + DirectX では左上原点 Y 軸下方向ですが、Android + OpenGL は左下原点で Y 軸上方向です。ここは Mikan ライブラリが左上原点なのでそちらに合わせるようスクリーンの設定を行いました。これに加え、描画の向きも標準では DirectX と OpenGL の向きが異なるため、ある種の資産再利用のために逆向き (Mikan ライブラリと同じ方向) にしました。

また、描画モードについては、加算や乗算は良いのですが、減算などが OpenGL のバージョン依存のようなので、減算は今のところ用意しないこととなりました。Android のバージョンごとのシェアからそこら辺の機能拡充について考える予定です。

次に、このライブラリは画像の描画がある種のメインとしているため、基本図形の描画は若干重くなる可能性があります。というのも、基本図形と画像の描画は根本では同じ板ポリゴンの描画で実装されています。しかし、内部的には画像を貼り付けるモードとそうでないモードを毎回切り替える必要があります。OpenGL的にはそれらをちゃんと設定する方がよさそうなのですが、今回は基本画像描画で設定を保持し、基本描画時のみその設定を無効化して描画し、再度有効化するため、若干処理が重くなるのではないかと考えられます。また、これらの事情によりオープンソースで公開しているのですが、一部分を切り取って使っても正常に使えない可能性があったりします。今後の課題として、ここら辺の実装についてはもっと調査して適切な状態にしていきたいと思います。

## 5.2 ライブラリについて

ライブラリは初めに書いたように JAR ファイル単体のみで稼働するようにしました。AIDE といった Android 上で Android アプリを作るアプリでも libs フォルダに Amanatsu の JAR ファイルを入れ、適切に import したプロジェクトを作ったところ、無事稼働しました。これで当初の目的は達成しました。

また、起動時にデフォルトでは Amanatsu のロゴを表示しています。これは無効化も可能なので単純なお遊びでもあるのですが、JAR ファイル内に入れておいた画像を描画するテストでもあります。

さらに、このライブラリは先ほど書いたようにオープンソースです。GitHub にて公開しており、誰でも改変及び利用が可能です (利用だけなら JAR ファイルをダウンロードするだけで良い)。

## 6 最後に

大体このような形で開発が進みました。ざっくりと 1 週間程度で基板ができ、次の 1 週間で最低限使えるものに仕上げ、2013 年度の春合宿でサンプルゲームを作る規模になりました。全体では約 1 ヶ月ほどかかりました。OpenGL 直叩きは初めてでしたが、DirectX よりは準備も少なく、また DirectX で培った基礎知識と楽なライブラリの揃った Java だからこそ、この期間で使い物になるものができたかなという感想です。特に画像の大きさ補正やハッシュなどでは大いに楽ができました。やはりオブジェクト指向が前提の設計、豊富なライブラリや機能がデフォルトで提供されていると開発がとても楽で良いです。

一方メモリ管理やライフサイクルには苦しめられました。Java にはデストラクタが無いため終了時に処理を行うには一工夫必要ですし、GC は専用の項目があるくらいには苦労しました。それでもアプリ終了は Android 任せなので、後片付けフェーズは正常に実行されないことが多々あります。細かなチューニングが必要な場合には言語や GC 等の仕組みを理解していなければいけませんし、場合によってはいろんな部分を自分で実装する必要もあるので、そこら辺は GC 搭載言語の悪い部分でしょう。

今後についてですが、Mikan 系列のライブラリは私という需要がはっきりとありますので、いろいろな方面に勢力を拡大させつつあります。Amanatsu の後は WebGL という OpenGL の Web 版を JavaScript で使えるようにしたものがあるので、それらを使ったライブラリも開発中です。本当は JavaScript には明確な型情報がなく、クラスなども非常に面倒な実装方法になるので私は大規模開発に手を出したくありませんでした。しかし、TypeScript<sup>10</sup>と呼ばれる JavaScript の上位言語で実装し、JavaScript に変換する方法を採用することで、楽に開発をすすめることに成功しました（変数名や構造もほぼ残るので分かりやすい）。残念ながら開発し始めてすぐ WebGL の動く端末がないことに気がついたので作業凍結中ではありますが、FPS 制御及びライブ

ライブラリの基盤はできました。WebGL が動く端末を入手次第、続きの実装を TypeScript で行っていく予定です。さらに、最近 PSVita を入手し、ちょうどライセンスキーが 1 年無料となるキャンペーン中だったこともあり、そちらでも似たようなライブラリの作成を行っています（忙しくて出来はまだまだですが）。

これから先、個人でも携帯端末やゲーム機での開発がどんどん可能になっていくでしょう。数年前には考えられなかったことです。Web もリッチになっているので、いろいろなプラットフォームでゲームが作られることでしょう。私は今年で X680x0 同好会を卒業しますが、ここで学んだことを生かし、少しでもいろいろな環境で、ゲーム制作を楽しんでいきたいと考えています。

## 7 リンクとか

Amanatsu - GitHub - <https://github.com/HirokiMiyaoka/Amanatsu>

Amanatsu 本体のダウンロードやソースファイルなど。利用する場合は Amanatsu.jar をダウンロードして下さい。

68 通信 vol.17 (2011 年度版) - <http://www.x68uec.org/other/press/2011/>

Mikan ライブラリを作ってみた記事があります。

---

<sup>10</sup>Microsoft が開発した。VisualStudio 以外でも Emacs や Vim、SublimeText で連携可能。

# 理地のめたの督提

腋酉 一十

月十年五廿成平

## 1 はじめに

最近艦これ流行ってますね。ありえないくらい流行ってますね。僕の知り合い集めてそこに向かって石を投げたら、たぶん提督に当たるくらい皆やっています。僕もやっています。今回はそんなみんなの大好きな艦これに出てくる艦娘たちの名前の由来、すなわち旧日本海軍の艦の名前の由来について話していきたいと思います。

船、特に大型船というのは車や列車のように量産できるものではなく、おおまかな形式が決まっていたとしても、個々の船というものの詳細はかなり違っており、一つ一つの船に対して名前がついていることが多いわけです。世界史で出てきた鉄剣宰相ビスマルクからとった超弩級戦艦ビスマルクとかは聞いたことあるんじゃないかなと思います。

では旧日本海軍の軍艦の名前ってのはどんな命名基準だったのでしょうか。適当に艦これに出てきた艦の名前を出してみましょう。

駆逐艦: 「吹雪」「舞風」

軽巡洋艦: 「長良」「多摩」

重巡洋艦: <sup>あたご</sup>「愛宕」「羽黒」

空母: <sup>ずいほう</sup>「瑞鳳」「<sup>ずいかく</sup>瑞鶴」「赤城」

戦艦: 「比叡」「伊勢」「長門」

名前を見てみると、駆逐艦は自然現象的な名前など雅な日本語が多く、空母は鳥の名前とか飛ぶやたらかっこいい感じなのが多い。また巡洋艦と戦艦は地名とか地域っぽい名前が多いように見えますね。正式には

駆逐艦: 天気や海、季節などに関連する単語

軽巡洋艦: 河川の名前

重巡洋艦: 主に山の名前、  
河川の名前も用いられた

空母: 神話に出てくる飛行動物、  
後に山の名前も使用可

戦艦: 律令体制時代の旧国名、  
「扶桑」は日本の雅称でこれは例外

となっています。

今回は山や川・地名地域に絞って、つまり巡洋艦と戦艦がメインとなるようにお話を進めていきたいと思います。説明はその艦の説明ではなく、その地域等の説明になります。あらかじめ、阿賀野型重巡洋艦や戦艦武蔵は、艦これではまだ実装されていません(2013年10月現在)が、近い未来実装されると思うので一緒に書いておきました(追記: 武蔵は11月のイベントで実装されるみたいですね)。地図に関しては google の maps engine で作りました。



図 1: 樺太 (サハリン)

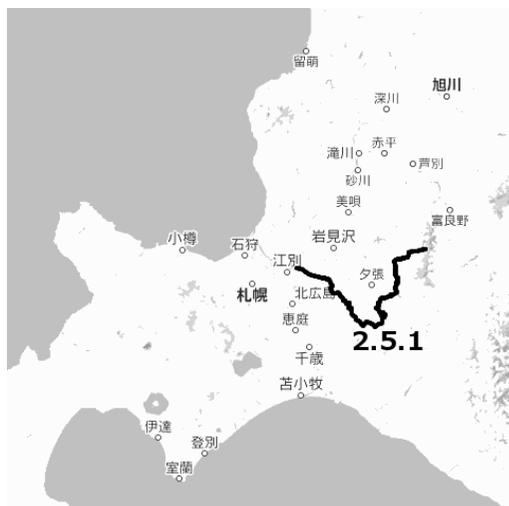


図 2: 北海道

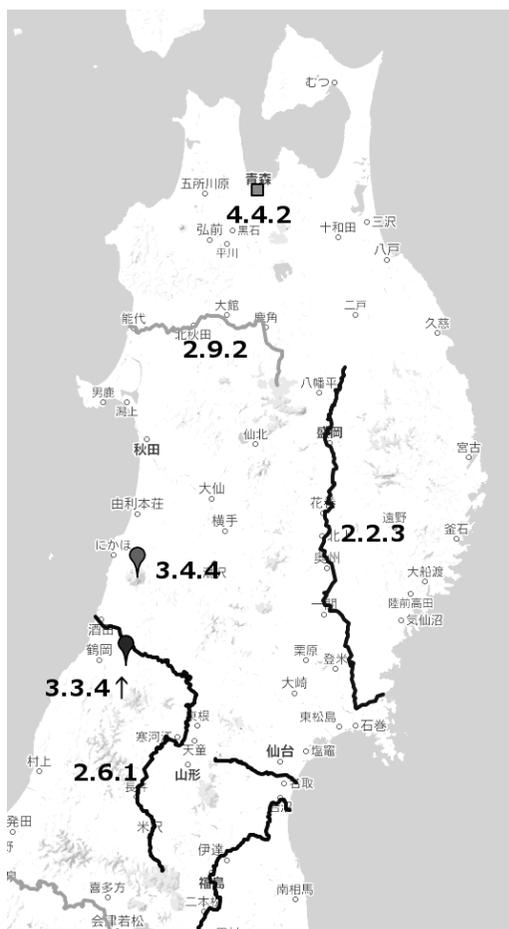


図 3: 東北地方等

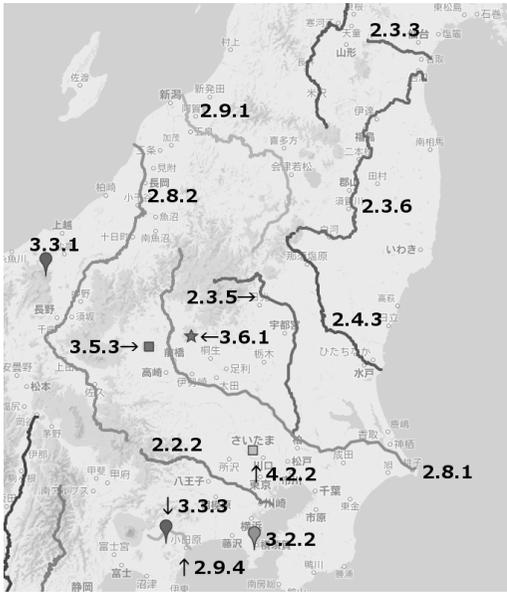


図 4: 関東地方等

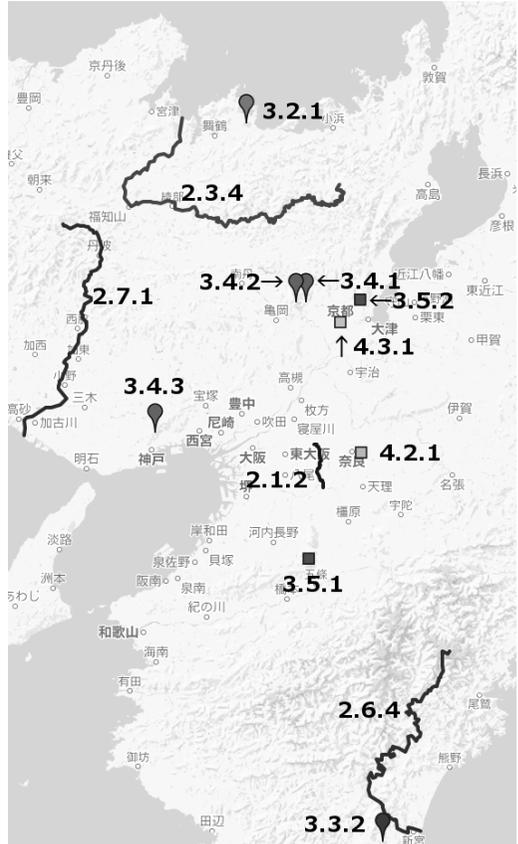


図 6: 近畿地方等

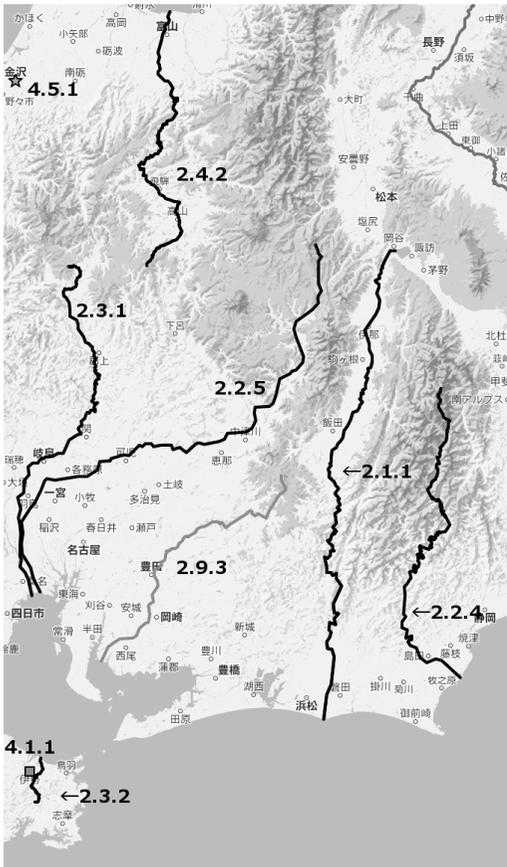


図 5: 中部地方等



### 2.3.5 鬼怒川

個人的難読河川シリーズ。源流は鬼怒沼、標高2000メートルの湿地帯。上流部は温泉が沢山あり湯治客も多い。昔は「絹川」「衣川」と書かれたが、水害が多く「鬼」が『怒』っているように思われたところから、現在の名前となった。常磐道守谷SAあたりで利根川と合流する。

### 2.3.6 阿武隈川

個人的難読河川シリーズ。福島県を縦断するような形で流れる川。まるで福島県の川のような言い方をすることが多いが、河口は宮城県岩沼市。長さは東北で北上川に次いで二番目。軍艦の方もぶつかるし、川の方は長さには敵わないしと仲が悪そうではある。

## 2.4 川内型軽巡洋艦の名前の由来となった河川

### 2.4.1 川内川

個人的難読河川シリーズ。上から読んで下から読んで川内川。九州で二番目の長さを誇る川。下流には「川内駅」があり、「川内から仙台」までの新幹線の切符が買える（どうでもいい）。

### 2.4.2 神通川

富山県と岐阜県を流れる川で、岐阜県では宮川と呼ばれる。飛騨や高山などの隠田集落による独自文化が栄えていた。そのうちが世界遺産の白川郷や五箇山にあたる。河口の富山市は市街地の交通機関の発展が（車社会が根付いている地方都市にしては）著しく全国から数多くの地方自治体から視察が来ることも有名。

### 2.4.3 那珂川

ここもアユが有名。築漁がおこなわれるところでもある。源流は阿武隈川と同じ山的那須岳。河口はひたちなか市（ちなみにひたちなかを漢字で書くと「常陸那珂」。パッと見読めないね。）と大洗町の境になっている。大洗町の大洗磯前神社には軽巡洋艦・那珂の祈念碑がある。那珂川の説明したんで那珂ちゃんファンやめます。

## 2.5 夕張型軽巡洋艦の名前の由来となった河川

### 2.5.1 夕張川

石狩川水系の支流である夕張川。江別市で本流と合流する。ところでメロンで有名な夕張市であるが、財政破綻で大変な中、大夕張ダムところに夕張シューパロダムを建設中。新たな観光資源になるといいね。支流にかの有名なヤリキレナイ川がある。

## 2.6 最上型重巡洋艦の名前の由来となった河川

### 2.6.1 最上川

日本三大急流の一つ。一つの県にしか流れない川のうちで一番長い川。「五月雨をあつめて早し最上川」という芭蕉の句があまりにも有名であるが、芭蕉は「あつめて涼し」かで迷ったそうない。急流だから「早し」になったのかも知れない。

### 2.6.2 筑後川

最上型二番艦は三隈であるが、三隈川は筑後川の一部区間、大分県日田市にかかる部分のさらに一部がそれである。筑後川は三大暴れ川の一つで「筑紫二郎」とも呼ばれる。ここ、クイズに出るから覚えとくといいよ。

### 2.6.3 鈴谷川

南樺太（サハリン）が日本領だった時の川。鈴谷岳（チエーホフ山）が源流で豊原市（Южно-Сахалинск）郊外を経由して亜庭湾（アニヴァ湾）まで流れる川。個人的には南樺太が今も日本領なら、唯一の陸続きの国境線になるから面白いと思うんだけどなあ。そしたらパスポートもって豊原まで行きます。

### 2.6.4 熊野川

熊野神社の周囲の流域はいわゆる「熊野古道」の一部として世界遺産の一部になっているすごい川。水害が多い。

## 2.7 古鷹型重巡洋艦の名前の由来となった河川

### 2.7.1 加古川

重巡加古はもともと川内型軽巡の予定だったらしく、川の名前になったらしい。加古川には加古川線が並走っていて、最近までトラウマ級の変な塗装の103系が走ってた（参照：眼のある電車）。ちなみに流域の西脇市は東経135度、北緯35度が交差していて、日本のへそ公園とかいう謎の公園がある。

## 2.8 利根型重巡洋艦の名前の由来となった河川

### 2.8.1 利根川

日本三大急流の一つで「坂東太郎」の異名がある。ここクイズに出るよ。首都圏の水瓶でも有名。作ろうとしたり中止したりまた作ろうとしたりと地元民がいろいろ可哀相な目にあってるハツ場ダムも利根川（に作ってる）のダム。どうでもいいけど「ッ」を「ん」って読むのは無理あると思う。流域面積日本一、長さ日本二。

## 2.8.2 信濃川

利根型重巡洋艦の二番艦は筑摩<sup>ちくま</sup>だけど、由来である千曲<sup>ちくま</sup>（筑摩）川は信濃川の長野県領域の名前。信濃だと千曲川。越後（新潟）だと信濃川。すっごく紛らわしい。日本一の川の長さ。源流は甲武信ヶ岳。甲州・武蔵・信州の境の山だから甲武信ヶ岳（諸説あり）。やる気ないね。

## 2.9 阿賀野型重巡洋艦（艦これには未実装）の名前の由来となった河川

### 2.9.1 阿賀野川

福島県内だと阿賀川<sup>あが</sup>（磐越西線あたり）とか大川（会津鉄道線あたり）とも言われる。中下流は第二（新潟）水俣病の惨劇があった川でもあるが、今は綺麗な川。水系にはダムが多い。福島は電源村だからねしょうがないね。

### 2.9.2 米代川

阿賀野型2番艦の名前は能代<sup>のしろ</sup>。米代川の下流部の名称が能代川ってところから。源流は八幡平<sup>はちまんたいら</sup>という山々の集合体。流域には鉱山が多くこの川を用いて運んでいた。

### 2.9.3 矢作川

個人的難読河川シリーズ。阿賀野型3番艦の名前は矢矧<sup>やはぎ</sup>であるが、今は読みが同じで「作」の字を使う。おきやはぎの「やはぎ」はこの字を書く。河口は三河湾。

### 2.9.4 酒匂川

個人的難読河川シリーズ。静岡県内は鮎沢川<sup>あゆざわ</sup>と呼ばれる。源流はそうかえんで有名な御殿場市の滝ヶ原駐屯地・東富士演習場の近く。川も演習場を横切っている。御殿場市内はひげ根のように支流があり、御殿場線駿河小山駅あたりで一体化。御殿場線と小田急線を並走して小田原市内で駿河湾に出る。

## 2.10 大淀型重巡洋艦（艦これには未実装）の名前の由来となった河川

### 2.10.1 大淀川

淀川は琵琶湖から大阪に流れ込む川、これとは関係ない。河口は宮崎県は日向灘。宮崎県を流れる川。数年前にそのまま東が県知事になってから一躍話題の県になった。このマンガーは冗談抜きにおいしい。

## 3 山の名前による艦名

### 3.1 古鷹型重巡洋艦の名前の由来となった山

#### 3.1.1 古鷹山

広島県の離島、江田島<sup>えたじま</sup>にある古鷹山。旧日本海軍の要所の島で、海軍兵学校があった。男塾の塾長江田島はここから名前をとってるらしい（出展：民明書房『男塾塾長 江田島の数奇な人生』）。呉もそこそこに近いついていうか橋でつながってるから離島って感じがしない。現在も海自の第一術科学校があり、自衛隊の卵がここから生まれる。

### 3.2 青葉型重巡洋艦の名前の由来となった山

#### 3.2.1 青葉山

個人的には青葉山と言えば仙台を思い浮かべるけど、重巡青葉はこの山が由来。京都府と福井県の境の山。この辺一帯はリアス式海岸でいい漁場になっている。古くから寺があり信仰の山。

#### 3.2.2 衣笠山

横須賀市内、崖の上側にある衣笠山。それこそ、横須賀鎮守府の目と鼻の先にある。今も昔も軍港として栄えている。今の横須賀市内はアメリカ海軍第七艦隊横須賀基地があるせいか、日本のベッドタウンとアメリカ的な雰囲気<sup>きぬがさ</sup>が交差する面白い街。カレーは美味しい。

### 3.3 妙高型重巡洋艦の名前の由来となった山

#### 3.3.1 妙高山

ゴジラ・モスラ・キングギドラ 大怪獣総攻撃の一番初めのシーンはここの山。活火山。山麓は温泉、山はスキーのメッカとリゾートには事欠かないところ。

#### 3.3.2 那智山

熊野三山の一つ那智山。熊野神社まわりで世界遺産。信仰の山。天然記念物の那智原始林もあり、自然豊かな山でもある。天気が良いければここから富士山が見えるらしい。

#### 3.3.3 金時山

妙高型重巡洋艦二番艦は足柄<sup>あしがら</sup>。足柄山は金時山から足柄峠にかけての区間の山々。昔話の金太郎の本名は坂田金時で、こころの出身らしい。球磨<sup>はぐる</sup>に乗るのか。

#### 3.3.4 羽黒山

山伏がいっぱい、修験道の聖地出羽三山の一つ羽黒山<sup>でわさんざん</sup>。湯殿山と月山が残りの二つ。出羽三山神社があり、そこにお参りすることで、3つすべてにお参りしたことにできる。昔は羽黒山以外女人禁制だった。

### 3.4 高雄型重巡洋艦の名前の由来となった山

#### 3.4.1 高雄山

愛宕山の隣にある山。東京都八王子市の方は「高尾山」。台湾にも「高雄」って地名があるけど、これは台湾併合時に「打狗」に発音が似ている、京都の名所であるこの名前を使って「高雄」としたらしい。そしたら漢字そのまま中国語で読まれるようになってこうなったらしい。

#### 3.4.2 愛宕山

京都市と亀岡市にかかる山。日本各地の愛宕神社はこの山の愛宕神社を総本社とする。「伊勢へ七度、熊野へ三度、愛宕様には月参り」ということわざ（信仰が深いたとえ）があるが、その愛宕さん。火事予防とかそのへんの神様。今このことわざ聞くと違った意味に聞こえる。

#### 3.4.3 摩耶山

元祖百万ドルの夜景といえば神戸市内の夜景であるが、隣の六甲山から摩耶山にかけて見える夜景がそれである。山頂まではケーブルカー・ロープウェイで簡単にいける。摩耶ちゃんさくらいだよ（なんかちがう）。

#### 3.4.4 鳥海山

東北で二番目の高さの山。美味しい水がわき出る。信仰の山だったがゆえに、山頂がどこ所属かで揉め事になったことのある山。うちの県も山形と山頂争ってわけのわからん県境作ってたなあ。

### 3.5 金剛型戦艦の名前の由来となった山

#### 3.5.1 金剛山

金剛型戦艦は、もともと巡洋艦として開発が計画されていたために、山の名前からとっている。金剛山は金剛山地の最高峰。奈良県と大阪府にまたがる。ここも修験道の聖地。神社とお寺がある。山頂まで行くのは、ロープウェイがあるので結構楽。大阪府最高点も金剛山にある。

#### 3.5.2 比叡山

古文で「山」と言ったら8・9割はこの山って古文の先生が言ってた。そんな比叡山。「比叡山・延暦寺」として歴史的に著名なお寺がある山。天台宗総本山で全国から仏門に入るうとする若者が次々とここで厳しい修業をする。世界遺産。

#### 3.5.3 榛名山

頭文字Dで有名な山。上毛三山のひとつ。信仰の山。山頂にカルデラ湖がある火山。側火山（山体側面に噴口がある）がいっぱいある。この辺の温泉だと伊香保温泉が有名。

#### 3.5.4 霧島山

頭に「黒」をつけると芋焼酎。つけなくても芋焼酎。日本神話における天孫降臨（「天」照大神の「孫」である瓊瓊杵尊が天界から「降臨」した）の山であり、由緒正しい信仰の山。活火山で噴煙は常時出ている。温泉も出る。

### 3.6 赤城型航空母艦の名前の由来となった山

#### 3.6.1 赤城山

頭文字Dで有名な山その二。上毛三山のひとつ。色々な火山が合体したような複成火山。そのため一口に赤城山と言っても色々山頂がある。群馬県や埼玉県では赤城風と呼ばれる冬の風も有名。

## 4 旧国名・その他地名による艦名

### 4.1 伊勢型戦艦の名前の由来となった旧国名

#### 4.1.1 伊勢国

伊勢神宮の御膝元。伊勢神宮は正式には「神宮」と称する。現在の三重県の大半、伊賀と志摩以外の部分。古来から神宮のお膝元して朝廷や幕府に手厚く保護された。今年平成25年は20年に一度の式年遷宮のある年で、なし崩し的に言えばお社の建て替えの年。宮大工の技術を後世に残すために20年に一度というスパンで行っているらしい。

#### 4.1.2 日向国

日向国は現在の宮崎県全体をさす。ぱっと思えば浮かんだのは日向夏。正確には鹿児島県の志布志市あたりまで日向国だったようで。志布志市といえば志布志市志布志町志布志に志布志市役所志布志支所があるってトリ〇アの泉とかで紹介されたことがあった記憶がある。

### 4.2 大和型戦艦の名前の由来となった旧国名

#### 4.2.1 大和国

日本全体をさす大和だが一番初めの大和はこの辺。ヤマト朝廷時代を考えればめちゃくちゃ歴史のあるところ。シカせんべいは美味しくない。

#### 4.2.2 武蔵国

東京都と埼玉県、神奈川県川崎市横浜市あたりがだいたい武蔵国。自分の周りだと逆に書くことが少ない。

#### 4.3 扶桑型戦艦の名前の由来となった旧国名

##### 4.3.1 山城国

山城国は京都南部のエリア。いわゆる畿内。京都といえば伏見の酒。あとウズラとスズメの丸焼き。伏見稲荷大社の山をのぼるとどんどん自販機の値段が上がるので要注意。

#### 4.4 長門型戦艦の名前の由来となった旧国名

##### 4.4.1 長門国

長門国は、現在の山口県の日本海側のエリア。いわゆる長州。本州の西の端である長門は、古来から本州側の下関しものせきと九州側の門司もんじ間の関門海峡を挟んで九州との交流が盛んであった。ちなみに関門海峡は最狭区間で500メートルでとっても狭い。調べた僕が驚いてる。泳げそう。こんなに狭いのに船の往来はとても多く昔から交通の要所。

##### 4.4.2 陸奥国

陸奥は「みちのく」とも呼び現在の福島県以北の東北地方（出羽国の山形県・秋田県除く）全体がそれであった。古来から白河の関なごそ。勿来の関以北のエリアは（言い方はアレだが）僻地であり、そこの関所が破られないように厳しい監視の目が敷かれていたのは時代劇でもよく聞く話。明治時代以降はおおまか青森県が陸奥国。津軽鉄道のストーブ列車でスルメを焼くのが僕の野望の一つ。

#### 4.5 改装空母の名前の由来となった旧国名

##### 4.5.1 加賀国

空母加賀はもともと戦艦として開発刷する予定であったため（下の信濃も同じ）、命名規則に乗っ取らず旧国名から名前が決まっている。加賀百万石。昔から金が有名なところで、今でも金箔の99%が石川県産。ちなみに金箔は100nmレベルの薄さ。すごい。

##### 4.5.2 信濃国

信州こと信濃は今の長野県全域を指す。北部（長野）は善光寺の門前町として栄え、中部（松本）は全国的にも珍しい黒い城の松本城（城が国宝）。また、軽井沢や上高地などもあり観光地として主に関東圏から四季を通じて観光客が訪れる。艦これでは未実装の艦。

#### 4.6 その他

##### 4.6.1 間宮海峡

樺太（サハリン）とユーラシア大陸との海峡。ロシア語名Татарский пролив。日本語による名前の由来は、江戸時代に樺太が島なのか半島なのかという論争が起きた際に、樺太に派遣された人物である探検家・間宮林蔵からとったもの。

##### 4.6.2 扶桑

「はじめに」でも述べたように扶桑は日本の雅称。もともとは、古代中国で日の出る東海の中にあるとされた神木とのこと（出展：大辞林）。すっこい壮大。トラックとかバスとかを生産している三菱ふそうの「ふそう」もこれ。

# iPadで 絵を描こう

書いた人:じいる

## ・はじめに

お絵かき、楽しいですね！でも忙しい大学生には講義中くらいしかお絵かきの時間がない……でも大丈夫！iPadがあれば、教室の後ろで板書を取るふりして絵が描けちゃいます！しかも指で描くから音が立たない！無敵です！

ということで、iPadでここ1年間絵を描きまくってました。描き方のコツみたいなものもつかめてきたので、ここで紹介しようと思います。

## ・アプリ選び

いろいろあります。どれもだいたい500円以内でフルに機能が使えるようになってるので、ぜんぶ試してみるのが一番です。

### ◆Procreate 450円(無料版なし)



iOSのお絵かきアプリではおそらく最強。レイヤーが(キャンパスサイズ依存で)30枚程度使えたり、不透明度保護からブラシの作成まで自由にできたりと、PC用ペイントツールで使えるだいたいの機能を搭載している。最近 Apple Design Award なるものを受賞したらしく、その流れで課金要素だった追加ブラシが全開放された。わりとこまめに機能追加される。

## ◆Bamboo Paper 無料(追加ブラシなどに課金)



落書き用に気持ちよく書けるツール。パレットが固定されているが色選びのセンスは良い。ノートをモチーフにしており、パラパラとめくるようにどんどん次の紙に落書きしていける。半透明ペンと普通のペンのみ無料で使えるが、その二つだけで描くほうが頭を使わず気持ちよく描いていけるのでブラシには非課金がおすすめ。何気に手ぶれ補正が強くかかる。

## ◆ibispaint 無料(広告を消すのに 350 円)



アプリ内に SNS を持っており、クリエイターだけが集まっている空間を楽しめる。drawr の空気に近い。そこに絵を投稿すると製作時間、作画工程 (youtube に勝手にアップロードする)、ベクターファイルも一緒にアップロードする男前仕様。ただ残念なのが UI で、絵自体が描きづらい。SNS は面白いのでアップデートに期待。

## ◆i せんせい 350 円(無料版なし)



タカラトミー社開発の、磁石でボード内の砂鉄を吸い寄せて描くアレである。消しゴムがない。黒と赤の二色で描ける。しかし消しゴムがない。なぜか 4 コマ漫画作成機能がある。でも消しゴムがない。塗りつぶしていく時に砂鉄の目がうごうごと動く部分まで再現されている。それでも消しゴムがない。本物だって裏から描けば消えるのに。

## ・実際に描いてみる

最初に紹介した Procreate を使って、実際に 1 枚描いてみることにします。

## ◆ラフ

構図さえ決まれば一番楽しい部分です。正確に描く必要がないのでずばずば好き放題できます。

紙ではできない、「ちょっと胴長になっちゃったから腰削って縮める」とか「足がはみ出したから全体を上を持ち上げる」だとかできるので、紙でラフ作ってスキャンするより楽ちんです。地のデッサン力をごまかせるデジタルの力に感謝。

## ◆ペン入れ

中くらいにめんどくさい部分です。

iPadはタッチパネルでしかないので、筆圧分解能は2段階です(指が付いているか、離れているか)。なので線の強弱は指運びの緩急で表現するほかありません。書道です。

ペン入れが終わりました。慣れると1時間もかからず終わらせられます。



## ◆色塗り

最高にめんどくさい部分です。細かい部分を塗り分けるのが、指で描くという性質上難しいので、エロゲ塗り(\*1)、アニメ塗り(\*2)はかなりめんどくさくなります。

(エセ)水彩塗りならかなり楽にできます。15%くらいの不透明度で広めの面積をざっくり塗っていきましょう。こだわらなければ10分で済みます。もともとハード的なスペックに限界があるので、凝った絵を描ききるのは難しいです。さっくり落書きをするものだと割りきって楽に描いていきましょう。



## ◆完成

最背面のレイヤーにそれっぽい模様を加えて完成です。色塗りさえ割り切ってしまうと2時間もかからず1枚描き上げられます！



## ・おわりに

楽しいですよ、iPad お絵かき。スペック制限はたしかに厳しいけど、液晶に直接触って描いていけるのは直感的で気持ちいいです。場所を選ばないのもメリットで、iPad さえあればいくらでも暇潰せるっていうのはいいものです。iPad を音ゲーマシーンとしか使っていない人たちなんかには特にお勧めです。450円ですよ？ music pack ひとつ分ですよ？ これを期に始めてみませんか、お絵かき。

(\*1)アニメ塗り:アニメ絵の色遣いに準拠した描き方。ベースカラー+影レイヤーを使って塗りつぶし主体で描いていく。グラデーションをあんまり使わない。

(\*2)エロゲ塗り:エロゲ絵の色遣いに準拠した描き方。彩度の高い色をたくさん使う。グラデーションもたくさん使う。みばえがいい。肌がえろっちく描ける。

# Fami Tracker

## 事始め

11 五目

### 0 はじめに

---

「ファミコン」と聞くと、もはや遠い昔の遺物のような印象を受けるかと思います。事実、マシン自体の性能や使える画像も音源も、現在のゲーム機と比べるとチープなものです。しかし、「そこがいい」という人も数多くいるわけです。

ファミコンの内部音源や、あるいはそれらを連想させるピコピコした音色を用いた、いわゆる「チップチューン」と呼ばれる音楽ジャンルは、現在でも一定の地位を築いています。ニコニコ動画で検索してみると、様々な曲のファミコンアレンジなどが投稿されています。著者が数年前に投稿した東方のファミコンアレンジも何処かに埋もれていると思いますが探す必要はありませんまいや止めてくださいマジで。

今回紹介する「FamiTracker」というソフトは、「あまり難しいことを覚えたくない」「環境導入が面倒くさい」という人にオススメできる、気軽にファミコン風の作曲やサウンド制作ができるものです。MML (Music Macro Language の略。プログラミングのように打ち込みで作曲するための機械言語) を用いるよりも、より直感的でグラフィカルに作曲を行うことができます。この記事では、このソフトの扱い方や FC 音源自体について、軽い感じで説明したいと思います。

### 1 FC 音源について

---

詳しい仕様を詳しく解説するにはページも時間も著者の気力も足りませんので、ここでは軽い説明に留めておきます。

ファミコンの CPU (RP2A03) に内蔵されている音源には、

矩形波(2ch)・三角波(1ch)・ノイズ(1ch)・DPCM(1ch)

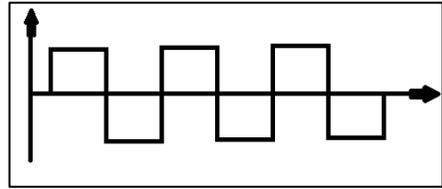
の計 5 チャンネルが用意されています。そのうちノイズ、DPCM を除くと、メロディラインに使えるのはたったの 3 チャンネルとなります。FC 音源がよく「3 和音」と呼ばれる所以ですね。

ここで各チャンネルの役割について見てみましょう。

#### i. 矩形波

「くけいは」と読みますね。「短形波」ではないです。覚えておきましょう。カクカクした波。電子工学のスイッチング回路とかでも使います。フーリエ級数を扱うとき

にサイン波の重ね合わせとかで見たかも。山と谷の比率（デューティー比）の違いによって音色が変わります。右の図だと50%（1:1）です。



50%（1:1）矩形波

FC音源では 12.5%、25%、50%、75%の4つが扱えます。

メインパートの要。それぞれを表現すると、

12.5%：一番荒い音。トランペットとかに使いやすいかな。

25%：一番安定感がある音。ピアノとかのメインメロディに使いやすい。

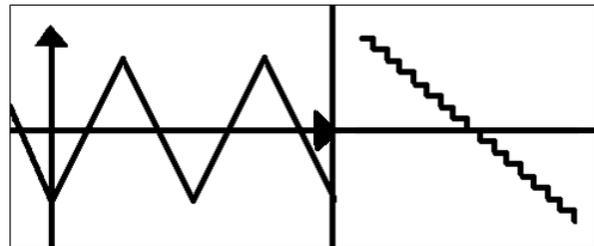
50%：最もファミコンっぽい「ピコピコ」音。笛っぽい？

75%：25%とほぼ同じ。

という感じですかね？

## ii. 三角波

FC音源の場合、綺麗な三角波ではなく、右の図の拡大図のように、16段階の階段状のギザギザがついた形になっています。ベース音に使ったり、高音域でメイン楽器として利用したりします。



三角波

（拡大図）

また、このパートの特徴として、音量は常に固定で変更できません。

## iii. ノイズ

長周期（ザーッと鳴る。テレビの砂嵐のようなノイズ）と短周期（キンッと鳴る）の2種類があり、主にリズム系の楽器や効果音に用います。

## iv. DPCM

サンプリング音源を鳴らせるパートです。主にドラム等に使われます。

以上がファミコンの内部音源の全てです。実機の場合、同時発音数の問題から、効果音が鳴る時はメロディの1パートがなくなってしまいます（プレイ動画などを観てみるとわかりやすい）。単純にFC音源で作曲をする場合は気にする必要はありませんが、効果音などの小ネタを仕込む場合は覚えておくといいかも。

FamiTrackerでは、これらのチャンネルを全て使うことができます。加えて、「VRC6」「MMC5」などの拡張音源を使うこともできます。そっちに関しては、今回は詳しくは述べませんが、まあチャンネルを増やしたいときとかに使ってみましょう。

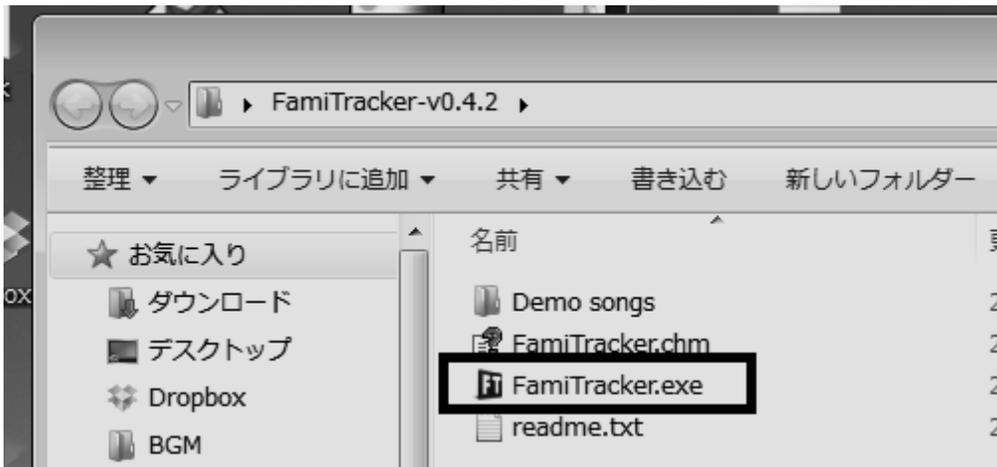
次のページからはいよいよFamiTrackerの導入とその使い方に入ります。

## 2 導入編

さて、実際に FamiTracker を使ってみましょう。FamiTracker は jsr 氏が制作・配布しているフリーソフトで、<http://famitracker.com/> からダウンロードできます。



サイトに飛ぶと全て英語ですが、「最新版のダウンロード」がトップページの一番上（上図参照）にあるはずですが（2013/9/20 現在で ver0.4.2）。それをクリックして保存するか、「Downloads」のページから好きなバージョンをダウンロードします。



zip フォルダを解凍して開くと（上図参照）インストーラの類はなく、exe ファイルが直に置いてあります。フォルダを置く場所はどこでも OK です。アンインストールするときは、このフォルダごと削除します。「Demo songs」のファイルはその名の通り、デモ曲の数々が入っています。後で聴いてみるのもいいかもしれません。

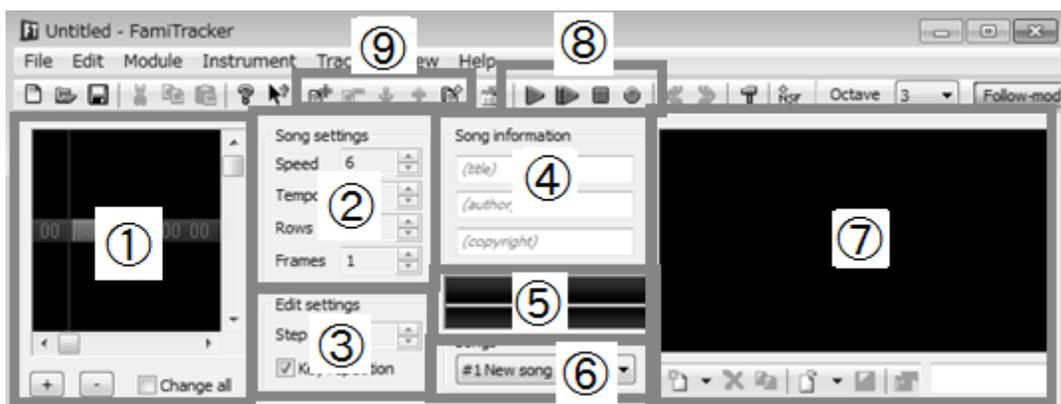
次項では実際に FamiTracker の画面を見ながら使い方の説明をしていきます。

ちなみに、FamiTracker では 16 進数を使います。これは 0~9 までの数字に加えて A~F までの文字を数字に対応させます。 $0A_{(16)} = 10_{(10)}$ 、 $10_{(16)} = 16_{(10)}$  という感じで、10 進数では 9 を超えると繰り上がるのに対し、16 進数では  $0F_{(16)}$  を超えると繰り上がることとなります。それを踏まえて次へどうぞ。

## 3 操作編

- 起動してみる

前項の「FamiTracker.exe」をダブルクリックして起動すると、下図のようなウィンドウが立ち上がります。画面は本来繋がっていますが、説明の都合上、上の設定画面と下のパターンエディタの2つに分割します。



設定画面（上画面）。やはり全て英語ですね。こればかりはどうしようもないです。

① トラックリスト。各パートを管理します。

パートごとに 00~FF（16 進数）までのパターンを確保することができて、同じ数字に設定すると同じパターンが鳴ります。

② 曲全体の設定です。曲のテンポや 1 フレーム辺り何行にするか、などの調整です。

③ 矢印キーでパターンエディタの行を移動するとき、何行ずつ移動させるかを設定するというもの。~~調べるまで知らなかった~~

④ 曲の情報。NSF ファイルなどに書き出す際に反映されます。

⑤ 波形が表示されます。クリックで表示を切り替え可能。

⑥ 曲の切り替え……らしいけど、まあ普通は使わないよ。うん。

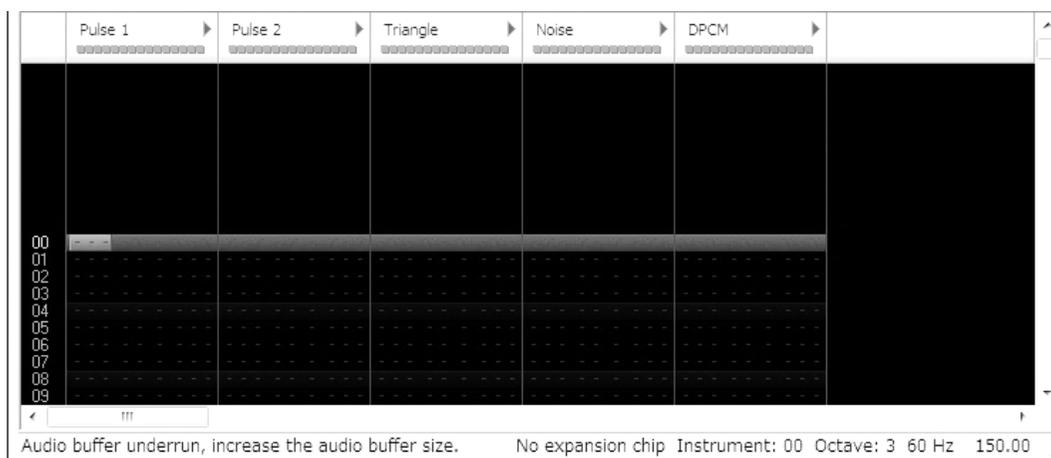
⑦ 作った音色のリスト。音色の作り方については後で。

⑧ 再生・停止など。二重の再生マークは「その場から再生」で、同じパターンのみを繰り返します。録画マークは「編集可・不可」を切り替えます。

⑨ 左から、

- ・ 選択中のフレーム（一パートでなく全体）の次の行に新しいフレームを追加。
- ・ 選択中のフレームを消去。
- ・ フレームの選択を下（上）に移動。
- ・ 選択中のフレームをコピーして次の行に追加。

となっています。

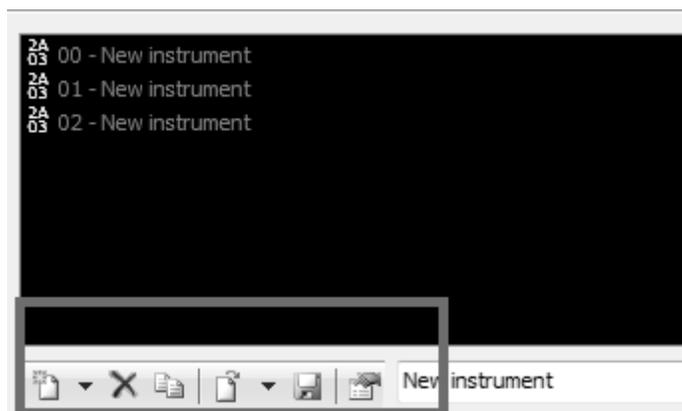


パターンエディタ（下画面）。実際にメロディなどを打ち込んでいく場所です。左から矩形波（2ch）、三角波、ノイズ、DPCMです。またメニューバーの「Module」→「Module properties」から設定することで、拡張音源も表示されます。対応したキーを押すことで、音を鳴らしたり、音を打ち込んだりできます。

……しかし、起動した直後は、押しても何も反応がありませんし、音も鳴りません。これはどうしたことでしょう。答えは簡単、「鳴らす音色がないから」です。

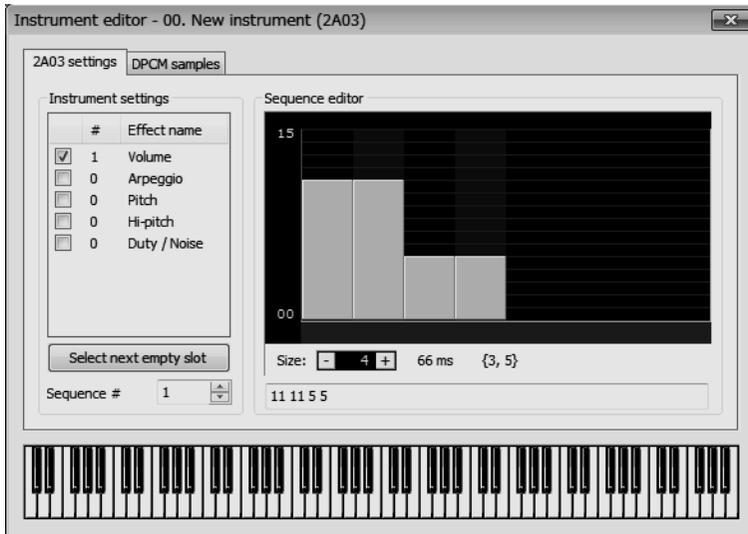
- 音色を作ってみる

前ページの⑦（音色リスト）の下にある、一番左のアイコンをクリックすると、「New instrument」として新しい音色が作られます。他にも、左から順に音色の削除、コピー、音色ファイルのインポートとエクスポートが可能です。右の入力スペースで音色の名前を設定できます。音色をダブルクリックすると、詳細な設定の編集ができます。



- 音色を弄ってみる

音色をダブルクリックすると、その音色のエディタが表示されます。「ボリューム」「アルペジオ（音の高さをずらす）」「ピッチ（音の調子の上げ下げ）」「ハイピッチ（ピッチより細かい調整）」「デューティー比」の5つを設定できます。



それぞれ左のチェックを入れることで、選択中の番号の設定が反映されます。例えば音量を（4 5 5 6 6 7 ……）のようにだんだん大きくすると、ストリングスのように変化させることができますし、ドラムだったら（10 6 4 2 0）のようにすぐに止めてしまいます。また、「|」（パーティカルバー）を途中で挟むことで、音色やテンポなどをその位置からループさせることもできます。なお、ここで使う数字は10進数です。

● 鳴らしてみる

先ほど音色を作った段階で、音が鳴らせるはずですが、適当にキーを押してみましよう。分かりづらいのは「\」キー(右シフトの隣)が休符（音を止める）、「;」「:」のキーが選択中のパターン番号の上下、「]」「[」のキーで「ファ」「ファ#」を表す、ということですかね。オクターブを上げてても配置は変わりません。

ここで、録画マークを押してみると、パターンエディタの選択中の位置が赤くなり、パターンエディタの編集が可能になります。



「Pulse2」を例にとると、一番左から「note（音階とオクターブ）」「楽器番号」「音量」「エフェクト類の設定（fx1 ~ fx4 の4つまで）」を表します。note を選択してキーを押すと音が配置されます。その他のパラメータは対応する0~Fまでのキーを押すことで設定できます。

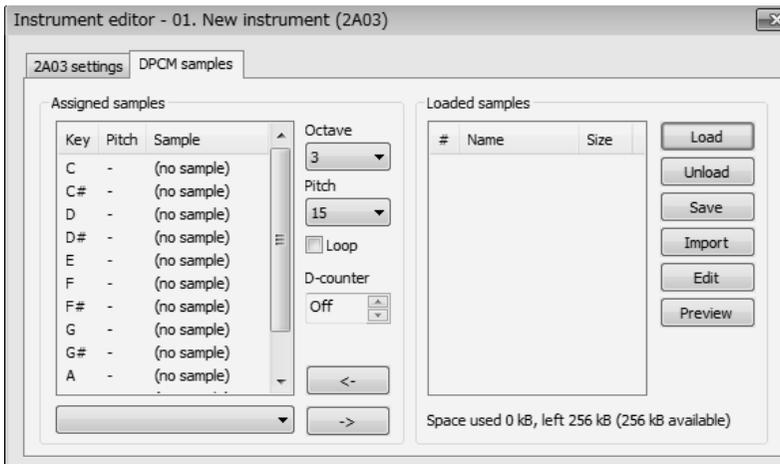
● エフェクトについて

音色自体をどう鳴らすかはエディタで設定しますが、それとは別にパターンエディタ上で「エフェクト」を用いて音色に様々な変化を起こすことができます。様々な種類があるので、よく使うものだけ解説（ヘルプから翻訳）します。

- 
- 0xy アルペジオ (x,y :二番目・三番目の音をどれだけずらすか。)
  - 1xx (2xx) ピッチを上げる (下げる) (xx : 変化スピード。00 にはできない。)
  - Axy 音量のスライド (A0x : x だけ下げていく。 Ax0 : x だけ上げていく。)
  - Bxx xx 番目のフレームにジャンプ (ループに使う。)
  - Dxx 次のフレームにスキップ (xx で何行目から始めるか設定可能。)
  - 3xx ポルタメント (次の音との繋がりを滑らかにする。 xx : 変化スピード。)
  - 4xy ビブラート (x : 変化スピード。 0 にはできない。 y : 変化の深さ。)
  - 7xy トレモロ (音量を上下させて小刻みに鳴らす。値は 4xy と同じ。)
  - Fxx スピード・テンポの設定 (00~19 : スピードの変化。 20~FF : テンポの変化。)
  - Qxy (Rxy) 音のピッチを上げる (下げる) (x : 変化スピード。 y : どこまで変化させるかの設定。)
- 

音色を弄るものから、曲全体の設定を司るものまで、色々あります。これらのエフェクトを上手く活用すれば、同じ音色でも場面ごとに全く別物にすることも可能ですし、音色自体の設定も簡単に済ませることができます。どのような場面でどんなエフェクトを使うと効果的なのかは、私自身も上手くつかめていない部分がありますし、個人の技量に左右される部分もあるので、使いながら慣れていきましょう。

● DPCM の使い方



音色エディタを開き、上の「DPCM samples」を選択すると、このような画面になります。「Import」を選択することで、外部の WAV ファイルを取り込むことができます。そのときに「Quality」や「Gain」で設定される数値は、それぞれ「音の質」と「音

の大きさ」を示しています（右図参照）。

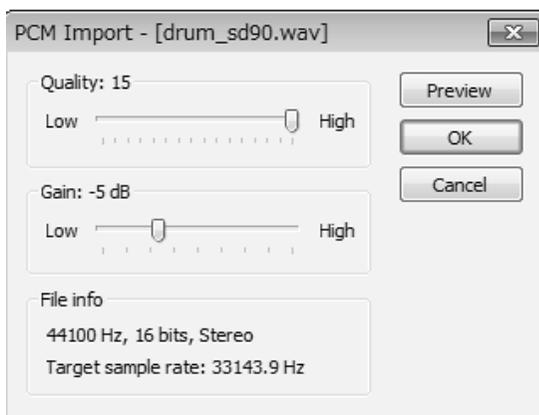
「Preview」から自分で聴いてみて調整するのが良いでしょう。また、「Edit」で波形の切り貼りなどの調整が行えます。

右側の設定したい音と、左側にある好きなキーをそれぞれクリックして選択し、「<-」をクリックすると、（設定中のオクターブの）選択したキーに音が登録されます。その他にもループなどの様々な設定ができそうですが、基本的な使い方はこんな感じで以上とします。

ノイズからドラムっぽいものを生成するのも一苦労ですし、何よりノイズの1パートしかないためにハイハットやらシンバルなどと両立できないのが厳しいので、積極的にDPCMを使っていきましょう。

- WAV ファイルを出力してみる

「File」→「Create WAV」からWAVファイルの出力ができます。色々設定できますが、そのまま「Begin」を押すと、1ループ分の曲が出力されます。また、NSFファイルを出力することもできます。



## 4 おわりに

---

今回は初めて FamiTracker に触れる人を対象に、最初から丁寧に進めていったつもりでしたが、いかがでしたか？ 使い方さえ解ってしまえば、割と簡単に曲らしく仕上げることはできますし、高度な技術も再現できるはずです。古の調べに心惹かれし者共よ、集うがいい！（FC音源が好きな人は、ぜひ試してくださいね！）

では、オタッシャデー！

## 5 参考文献・サイト

---

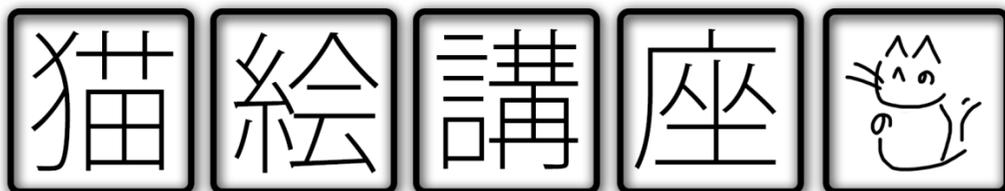
1. mck Wiki FamiTracker/チュートリアル <http://wikiwiki.jp/mck/>
2. FamiTracker Wiki（英語です） <http://famitracker.com/wiki/>
3. 「わかりづらい！東方音楽解説本」より 蓬屋ふらんさんの解説記事 <http://www.dobuusagi.com/tondemo/>（同人誌の紹介ページ）  
東方FCアレンジ界の重鎮。

あとニコニコ大百科の「FC音源」の記事が何気に充実しています。それに限らずニコニコ動画には様々な作品がアップされているので、聴いてみるといいですね。



「へのへのもへじ」で猫を描く

by ナスビ



「じかきうた」とは？

「じかきうた」というものをご存知でしょうか？ これは絵描き歌の一種で、主に文字を利用して絵を描くもので、言葉遊びの要素もちょっとあります。有名な例としては「つるさんはまるまるむし」や「へのへのもへじ」などが挙げられます。ぶっちゃけ、「じかきうた」をひと言で言うなら「へのへのもへじみたいなもの」といった感じです。



↑ つるさんはまるまるむし(左) と へのへのもへじ(右)

この「じかきうた」は非常に奥が深いもので、実際に「へのへのもへじ」で色々な絵を描いている芸人さんもいるほどです。

「へのへのもへじねこ」が流行



@nedikes

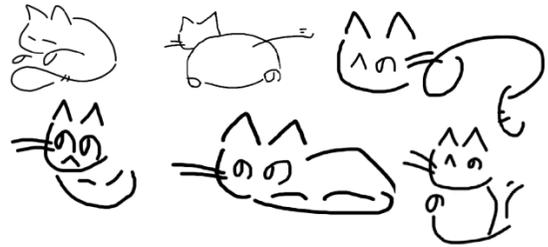


2012年4月、Twitterで「へのへのもへじ」を使って描かれた猫が巷で話題を呼び流行しました。この猫は「へのへのもへじねこ」と呼ばれ、その可愛さから、あらゆる派生猫が作られたり、Tシャツが作られたりしました。

↑ へのへのもへじねこ(左) と そのTシャツ(右・nesnoo®販売)

実はこの「へのへのもへじねこ」、絵の構造がかなりシンプルであるために誰でも簡単にアレンジを効かせたりして創作することが可能です。絵心がない人でも関係なく可愛い猫が描けます！ 実際、「俺、へのへのもへじ世界一うまく描けるぜ？」なんて自慢してるヤツどこにもいないですよ。

という訳で、可愛い「へのへのもへじねこ」を描くためのコツなどをのんびり書いていきたいと思います。



↑ いろいろ描いてみた「へのへのもへじねこ」

**「へのへのもへじねこ」の解析**



「へのへのもへじねこ」と言うからには、使える文字は「へ」×3、「の」×2、「も」「じ」×1の計7文字だけです。これらの文字をうまく猫のパーツと対応させるのが大事な所です。この文字と猫のパーツとの対応をまとめてみます。

例えば、猫には欠かせない「猫耳」を考えてみます。猫耳というと「角のところがた耳」ですよ。使える文字の種類は「へ」「の」「も」「じ」の4つ。この中でとがった部分を持つのは「へ」だけ。よって、猫耳に関しては「へ」でしか描きようがないことになります。

他にも「ひげ」を考えてみます。ひげがあるだけで猫らしさが出てくるので大事なパーツですが、横線が欲しくなるパーツでもあります。横線として使いそうなのは「じ」の濁点部分と「も」の横線ですね。よく使える手段として、こちらから見て顔の右側にひげを描くなら「じ」、左側に描くなら「も」を使って、両者に共通する「し」の部分を使って顔の輪郭を描くと、綺麗に猫っぽくなります。



↑ 「も」を使ったひげの描き方例(左)と「じ」を使ったひげの描き方例(右)

次に「しっぽ」について考えてみましょう。しっぽは長い線で表現ができそうです。無理やり長い線をかけそうなのは「へ」「も」「じ」の3つですね。しかし、「へ」は猫耳で既に2つ消費しており、残りの1つは必要な線の補助としてよく使われます(よくあるのは胴体の一部で使われるパターン)。



↑ 「じ」を使ったしっぽの描き方例

つまり、「へ」をしっぽで使うのはあまりない例です。「も」か「じ」のうち、ひげで使われなかった方をしっぽに持つのが最もスタンダードなパターンです。

このように消去法的に考えていくと、猫と文字の対応を整理しやすいです。これ以上書くと、ちょっとグダグダになるので、私が適当に猫パーツと文字との対応を表にしてみました。

猫パーツ	対応可能な文字(例)	説明
猫耳	「へ」	多分これでは描けないと思います。
ひげ	「も」「じ」	「し」部分で顔の輪郭を表現。
しっぽ	「も」「じ」、希に「へ」	ひげで使っていない方を主に使用。
目	「の」「へ」「じ」	基本は「の」。ウィンクは「へ」。ひげを犠牲に「じ」を使って、眠ったかのような横線の目を表現するのもあり。
前脚	「の」	「の」を「o」のように描いて前脚っぽく。
胴体	「へ」「の」	例外はあるものの、だいたいはこの2文字。

### 創作ワンポイントアドバイス



実際に「へのへのもへじねこ」を創作する際、頭の中でどの文字をどこに置くかを考えるよりも、適当に手を動かしながら考えるのが効率的です。また、「へのへのもへじ」の順番に従って文字を置く必要はありません。この際、「へのへのもへじの7文字を使って猫を描く」と頭の中で割り切って、その順番については頭から切り離しましょう。

個人的に一番効率がいいと思う方法として、先に顔の部分にじっくり時間をかけて描き上げていき、残った文字のパーツでなんとかして胴体やしっぽを完成させる手順があります。私の場合、ペイントソフトにマウスで描いているため、気を抜くと猫さん顔面崩壊という惨事になってしまうため、最初に顔を描いて、後は楽したいという理由からこのような手順をとっています。ちなみに私の場合、耳から描くのか、顔の輪郭から描くのかはその時の気分によって適当な順番になります。

慣れてきたら、「へのへのもへじ」の順番に沿って文字を置いていく縛りプレイなどをして面白いかもしれませんね。



↑ 「へのへの…」の順番通りに文字を置くより、顔から適当に描きながら作るのがやりやすい。



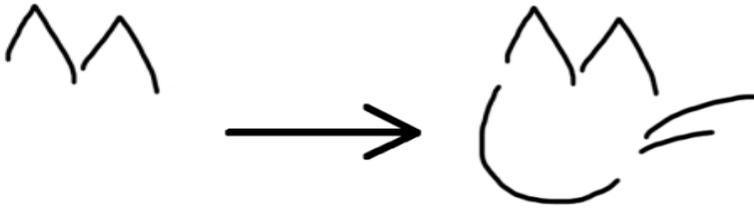
← 「言い出しっぺの法則」という事で、順番縛りで描いてみた。

実際に描いてみよう

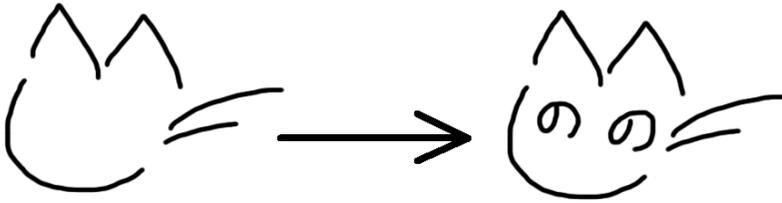


さて、実際に「へのへのもへじねこ」を描く手順の一例をお見せしましょう。あくまで今回私が描いてみたらこんな手順になった程度のもので、「こんな描き方があるんだな」程度に読んでください。

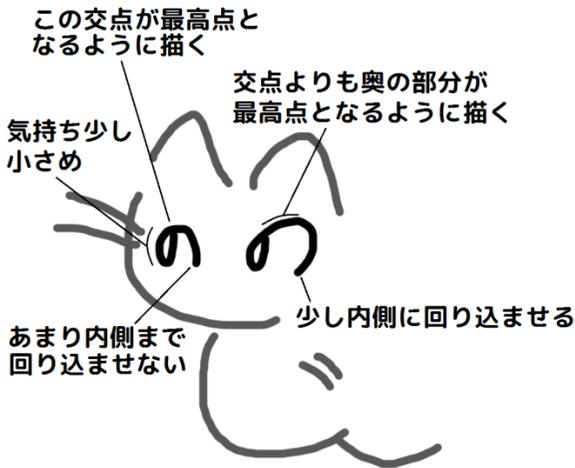
1. 顔の整形



まずは「へ」を2つ消費して耳を描き、次に今回は「じ」を使って顔とひげを描いてみました。ちなみに、言うまでもなく、マウスで一発で描けたわけではなく、アンドゥを連発して何回も描き直しまくってます。



次に「の」を2つ消費して目を描きました。「の」を使って目を描くのは結構難しく、可愛く見せるのには少しコツがいります。具体的に言うと、こちらから見て

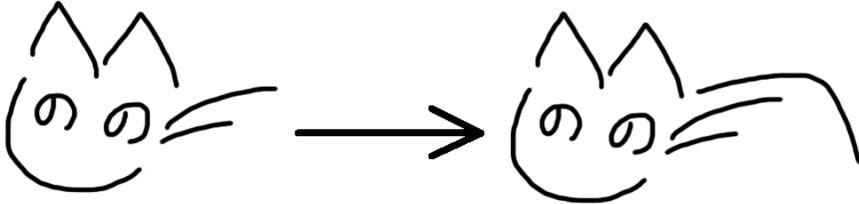


左側の目は少し小さめに書き、「の」の交点の部分が最高点になるように描く、右側の目は交点の部分より奥側が最高点となるように描く、といった具合なのですが、これに関しては左の絵を見てもらったほうがわかりやすいと思います。

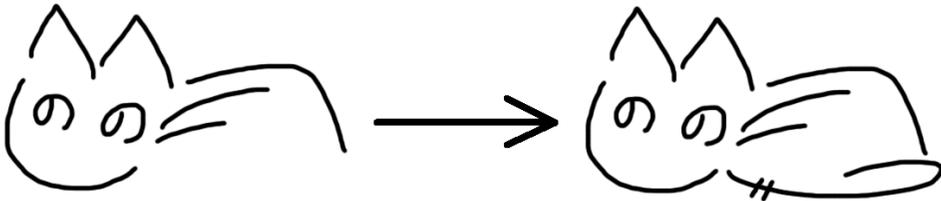
ですが、初めの内はあまりこれに凝り過ぎず、気楽に何度も描きなおすのが良いでしょう。

## 2. 胴体の生成

この時点で「へ」と「も」だけが残りました。この2文字で胴体を表現していきます。

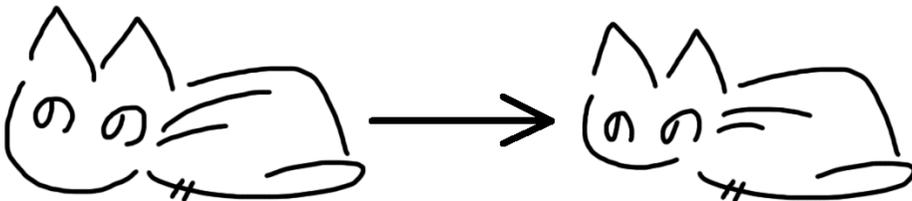


とりあえず、「へ」を使って胴体の一部である背中を描きました。しかしこの時点で猫の顔がカバミみたいになっていると気づきます。とりあえず、今描いているのは構想決定段階と割り切って作業を進めていきます。

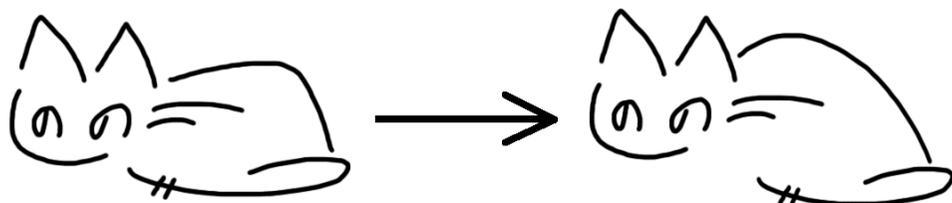


最後に「も」を横向きに配置して、しっぽが身体全体を巻いているかのように表現してみました。一応これで完成です……が、顔はカバミみたいだし、頭でっかちだし、ひげと背中がほぼ平行のおかげで横線が密集しているみたいでだらしがないし、目が可愛くないしで、あんまり出来としては良くないですね。ブサイクです。ここからアナログに描き直したり、デジタルに拡大縮小・移動などを行いながら修正をしていきます。

## 3. カワイイをつくる



まずは顔の部分を修正してみました。主な修正点は、目をさっき言ったコツの通りに描き直し、顔を小顔化。耳は微妙に描き直して縮小処理を施しました。ひげも綺麗な線になるように描き直しました。これだけで随分と猫っぽくなりましたね。



胴体部分も修正してみました。背中はもっと丸みをつけて、耳の後ろから線が出ているように描き直し、しっぽは小型化して位置調整しました。これで修正の作業も完了です。最後に、なんか見た目がなにか啜えてそうな感じの猫に仕上がったので、「へのへのもへじさかな（適当に命名）」を描き加えて完成です。



## 最後に



「へのへのもへじねこ」の面白さが少しでも分かってもらえたでしょうか。「じかきうた」のひとつであった「へのへのもへじ」が進化したらここまで面白くて、より奥深いものになります。本家である「へのへのもへじ」とは違って、「へのへのもへじねこ」の方は創作の幅がかなり広いところもより味わい深い点だと思います。ぜひ、この魅力に惹きつけられたなら、見るだけでなく自分だけの「オリジナルへのへのもへじねこ」を作ってみてください。

実は猫だけでなく、なんと有名なアイツも「へのへのもへじ」で描けてしまうそうです……？



# かんたん！ RPN電卓を使おう

’ 11 7tuv

## 0. はじめに

RPN電卓とは普通の電卓とは違い、ちょっと特殊な操作を必要とする電卓です。その特殊性のせいか、現代の日常生活ではほとんど見かけることはありません。しかし一旦その直感的な操作に慣れてしまうと、二度と普通の電卓が使えない“カラダ”になってしまうほど<sup>\*1</sup>、とても癖の強い代物でもあります。まずは基本的なところを紹介します。

## 1. RPNってなに？

RPNとは日本語で逆ポーランド記法 (Reverse Polish Notation) <sup>\*2</sup>と呼び、略して“逆ポ”なんて言われたりもします。ここでRPNとは何かを、普通の電卓の打ち方 (中間記法) と比較して説明します。例えば「2 と 5 を加算する」という演算をするとき、それぞれの記法で表すと次のようになります<sup>\*3</sup>。

- 2 + 5 (中間記法)
- 2 5 + (RPN, 逆ポーランド記法)

このようにRPNでは、演算記号を2つの数値の後ろに置きます。そのためRPNは後置記法とも呼ばれます。初めて見る人には奇妙でややこしいと思われるかもしれませんが。しかしこの表現は、日本語のようなSOV型の言語の文法とよく似ており、「2 と 5 を

<sup>\*1</sup> アニメ「ロケットガール」第2話より。実際俺もたまに間違える。

<sup>\*2</sup> 当然、ポーランド記法も存在します。これはLISPに使われてたりします。

<sup>\*3</sup> ポーランド記法は前置記法とも呼ばれ、“+ 2 5”のように式を記述します。

加算する」と読み下した順序で“ 2 5 + ”と記述したと考えると多少は理解しやすいと思います。

もう少し複雑な例を出してみます。まずは以下の中間記法の例を見て下さい。

- ( 6 \* 4 - 9 ) \* ( 2 + 3 ) (中間記法)

これをRPNで記述するところになります。

- 6 4 \* 9 - 2 3 + \* (RPN, 逆ポーランド記法)

ちょっとややこしいですね。見ての通りRPNでは括弧を用いることはありません。実際に日本語で読み下してみると「6 と 4 を掛けたものから 9 を引き、それを 2 と 3 を足したもので掛ける」となり、見事にRPNの順序と一致します。順々に計算していくと以下のようになります。

$$\begin{aligned} & \underline{6\ 4} * \underline{9 - 2\ 3} + * \quad (\text{下線部を計算}) \\ \rightarrow & \underline{24\ 9} - 5 * \rightarrow \underline{24\ 9 - 5} * \\ \rightarrow & 15\ 5 * \rightarrow 75 \end{aligned}$$

このように式の表記が簡潔になるのがRPNの利点です。

なお本記事において特に説明なく“ X + Y ”などと書いた場合は、中間記法であるとしています。

## 2. RPN電卓の使い方 (基礎編)

実際にRPN電卓を叩いてみましょう。この記事で用いているRPN電卓はHP (ヒューレット・パッカー) 社のHP-35sです。次ページの上図はHP-35sの全体図です。



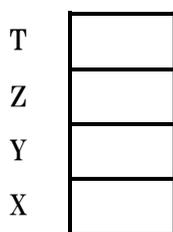
HP-35s

いろいろ文字やらキーやらありますが、よくみると“=”キーがありません！\*4 そうです。これがRPN電卓の最大の特徴です。では一体どのように計算をするのでしょうか。ここで“スタック”という概念を利用します。

## ◆ スタック

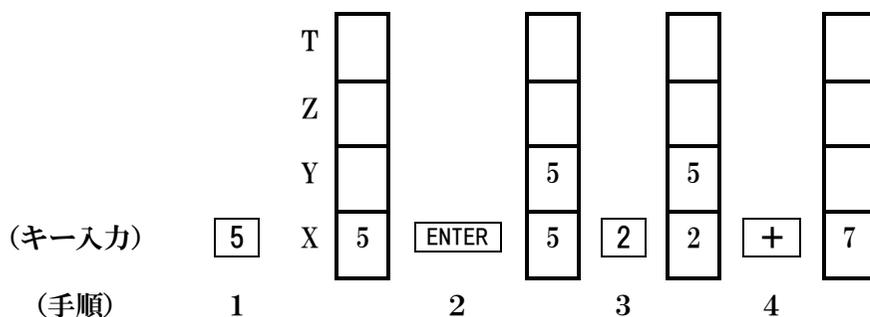
次ページの上図がRPN電卓におけるスタックの概要です。

\*4 真ん中にある“+/-”キーの上に“=”とありますが、これは式モード（変数を含んだ式を入力し、方程式を解いたり積分を行う機能）で“=”の文字を入力するときに使います。



スタックには数値を保存する場所が4個あり、これらをレジスタと言います。スタックとは日本語で“積み重ねる”の意味であり、文字通り4個のレジスタはスタックに「積み重ね」て保存されます。4個のレジスタには順番にX, Y, Z, T<sup>\*5</sup>という名前がついています。HP-35sでは、画面の下段にXレジスタの値が、上段にYレジスタの値が表示されます。

まずは“2 + 5”計算をしながらスタックについて説明します。



上図が概略図です。まずは電卓の 5 を押すとXレジスタに数値が保存されます。次に電卓の真ん中の左端にある ENTER キーを押すと、Xレジスタの値が一つ上のYレジスタに移動します。ENTER キーはそれぞれのレジスタに保存されている数値を、一つ上のレジスタに移動する役割を持っています。しかしXレジスタの値は ENTER キーを押した後も変わりません。またTレジスタに元からあった数値は押し出されて消えてしまいます。

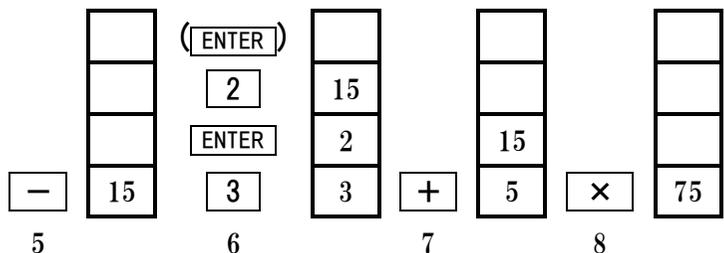
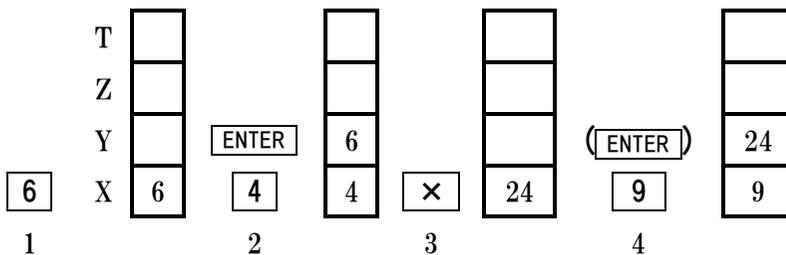
<sup>\*5</sup> Tはtopを表します。

次に **2** を押すとXレジスタの値は 2 になります。ここで2つの数値が揃い加算ができるようになりました。**+** キーを押すとX, Yレジスタの値が足されて、Xレジスタの方に演算の結果として 7 が保存されます。その際、Z, Tレジスタにある数値は下がって、それぞれY, Zレジスタへと移動します。Tレジスタについては値がそのまま保持されます。この値の下降を利用した電卓の使い方があるので後々説明します。

+ を含む四則演算について、2つの数値の順番には決まりがあります。X, Yレジスタの数値をそれぞれx, yとすると、“ x **演算子** y ” の順番で計算されることに注意しましょう。

まとめると、 **5** **ENTER** **2** **+** で 2 と 5 の加算の計算ができます。

では次に “ ( 6 \* 4 - 9 ) \* ( 2 + 3 ) ” を計算してみましよう。



キーだけの順番を示すと

**6** **ENTER** **4** **×** **(ENTER)** **9** **-** **(ENTER)** **2**  
**ENTER** **3** **+** **×**

となります。**(ENTER)**と書かれてあるところがありますが、ここでは **ENTER** の

入力を省略できます。演算子やその他関数のキーが押された後、Xレジスタにある数値は途中までの計算結果ですから **ENTER** を押す必要がありません。以降この入力は省略します。省略した場合のキー入力は次の通りです。

**6** **ENTER** **4** **×** **9** **-** **2** **ENTER** **3** **+** **×**

## ◆ いろいろな関数・機能

RPNとは関係ありませんが、関数について少しだけ書きます。HP-35sには実数を演算するうえで必要な関数はだいたい揃っています。指数、対数、三角関数、べき乗など、ほとんどの計算はこれらの関数で済んでしまいます。何かしらの説明が必要になったらその都度補足します。

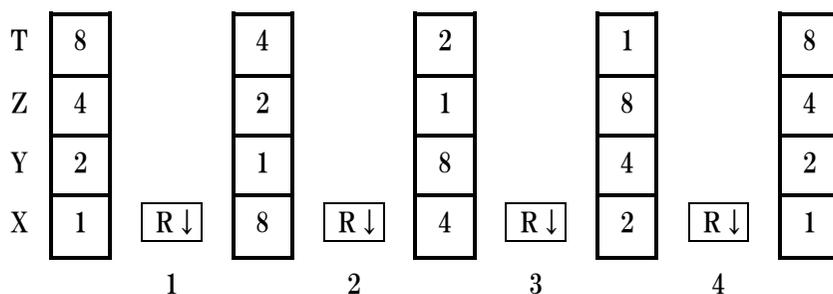
スタックを操作する上で便利な機能がいくつかあります。それらについて紹介します。

### ● シフトキー

HP-35sの左下の方に曲がった矢印のキーがあります (  と  )。これはPCのキーボードでのシフトキーと同じ使い方をします。キーにはそれぞれ三つの機能が割り当てられており、それぞれを使い分けるためにシフトキーを用います。

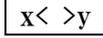
### ● ロールダウン

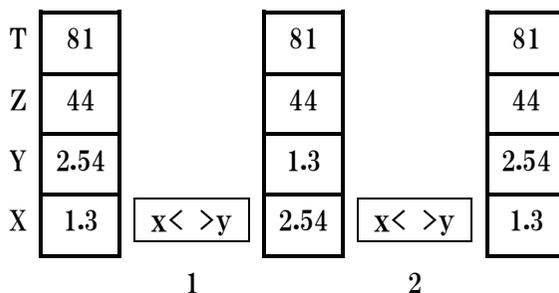
**R↓**(roll down)キーを押すとスタック中のすべてのレジスタの中身を下方方向に回転させます。これによりZレジスタとTレジスタの中身も確認することができます。



ロールアップという機能もあり、 **R↑**(roll up)キーを押すと、スタック中のすべてのレジスタの中身を上方方向に回転させます。

● XレジスタとYレジスタの交換

 キーを押すとX, Yレジスタの中身が入れ替わります。そこからもう一度  キーを押すと元に戻ります。引き算や割り算など、2つの数字の演算の順番を変えたい時に用いられます。



### 3. RPN電卓の使い方 (応用編)

スタック操作を用いたRPN電卓におけるテクニックについて紹介します。

● スタックを全て同じ値で満たす

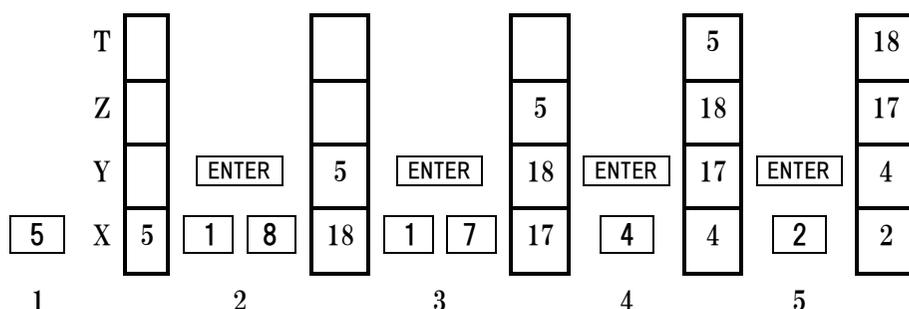
最初に数値を入力し  を3回押すと、全てのレジスタが同じ数値によって満たされます。ここで  や  を連打すると数字がどんどん増えていきます。Tレジスタに保存されている値は、常に保持されるためこのようになります。あなたも普通の電卓で    と押した後  キーを連打しまくる\*6遊び、やったことありますよね？ あれがRPN電卓でも可能です。あれものすごい楽しいですよ……

\*7。

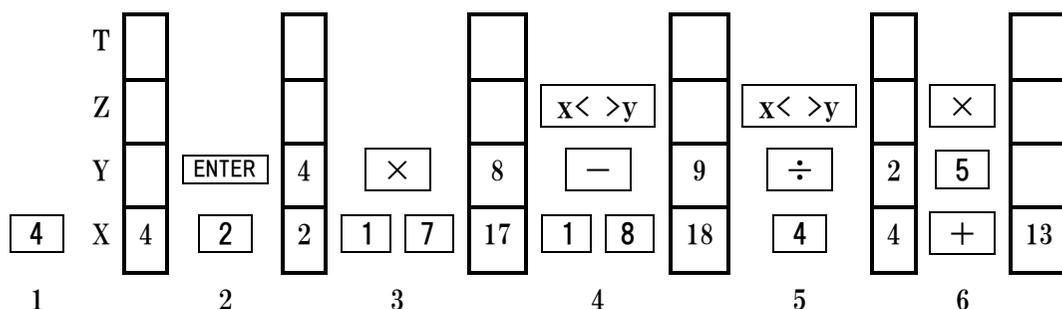
\*6 定数計算と言います。もちろん本来の使い方とは異なりますよ。

\*7 Cookie Clickerが流行った理由がよく分かります。



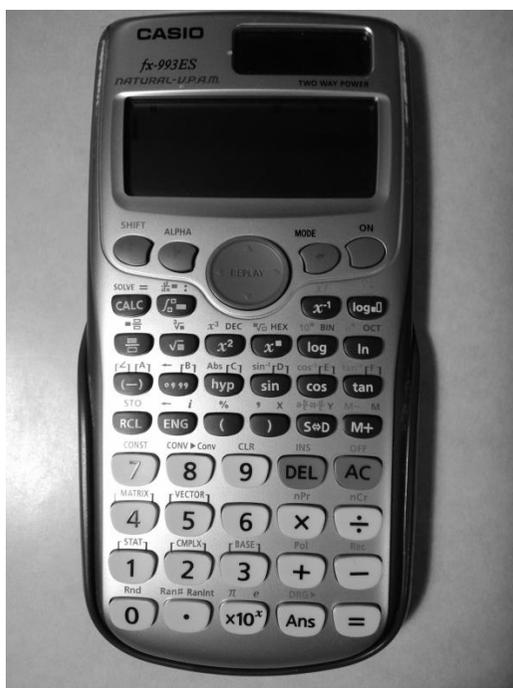


最初にスタックに入れた 5 は、2 を入れた時点で消えてしまいました。これではいけません。このように入れ子になっている式では、出来るだけ内側にある括弧から計算した方が必要なレジスタも少なく済み、計算の手順も楽な場合が多いです。



#### 4. RPN電卓の長所・短所

RPN電卓はどういった場面で力を発揮するのか、他の関数電卓と比較してみます。比較対象として、電気通信大学生協が推薦している関数電卓、カシオ社のfx-993/995 ESを選びました。電通大生のほとんどはお世話になっているはずです。



fx-993ES

このタイプの電卓は「自然表示」タイプと言って、数式がそのまま表示されるタイプ\*8の関数電卓です。計算するときは、式を見たまま入力すればいいので操作はとても簡単です。

- 打鍵数の少なさ

基本的にRPN電卓は、他の電卓と比べて打鍵数が少ないです。さきほどの“ $5 + \{ 18 / [ 17 - ( 4 * 2 ) ] * 4 \}$ ”を例に挙げます。

fx-993ESで計算式を入力すると打鍵数は、式の文字数とイコールを押した分を合わせて20回です。HP-35sでは、打鍵数は16回とfx-993ESよりも少ないです。このようにRPN方式では括弧を記述する必要がないため簡潔に入力が行えます。

- 長い計算式での演算

例えば次のような計算式を考えます。

---

\*8 この方式をWYSIWYG (What You See Is What You Get) と言います。呼び方は“ウィジウィグ”です。

$$\sqrt{\frac{a^2}{b^2} + \frac{c^2}{d^2} + \frac{e^2}{f^2} + \frac{g^2}{h^2} + \dots}$$

電通大の科目、基礎科学実験Aで不確かさを求めるときにこんな式を見た人、もしくは見ることとなる人がいるでしょう。これを二つの関数電卓で解く\*9場合を考えます。

まずはfx-993ESについて、入力文字数について制限があり、99文字まで入力できるようになっています。すると、その制限を超える長さの式だと一気に計算できない場合があります。

HP-35sの場合、ルートの中にある二乗分の二乗の項をまとめ、連続的に足すことができます。つまり式がどれだけ長くなっても計算の操作に支障は出ません。

## ● 積分計算

Fx-993ESにおいては積分キーを押し、それぞれに入れたい数字、関数を入力すれば。マニュアルを読まずとも積分計算が直感的にできます。これこそ「自然表示」タイプの長所と言えるでしょう。

逆にHP-35sにも積分できる機能はついてはいますが、数式の表示領域が一行しかないためFx-993ESに比べてかなり操作が面倒です。積分計算には不向きと言えます。

## 5. おわりに

ネット上にはRPN方式の電卓アプリが多数あるので、興味のある人は実機を最初から使うよりもまずそちらから触れてみてはいかがでしょうか。

RPN電卓（基本的にはHP社製）は現在も入手可能ですが、HP社が日本での電卓事業から撤退しているため、ほとんど市販されていません。輸入代理店から通販で購入するのが一番簡単な方法でしょう。ちなみにHP-35sはAmazonで6,990円です。

なお、今回使用したHP-35sには、方程式を解く、統計、プログラミングなど様々な機能が搭載されています。あ、あと今さらですがRPNじゃなくてALG（普通の関数電卓

\*9 実際に計算するときは無難にExcelとかを使いましょう。

の入力方法) も使えますよ。

## 6. 参考文献・サイト

- HP 35s Scientific Calculator 日本語ユーザーズガイド
- 関数電卓マニアの部屋 <http://teamcoil.sp.u-tokai.ac.jp/calculator/index.html>

## 7. おまけ

練習問題です。暇な方はどうぞ。

$$\textcircled{1} \quad 1 + 1 * 1$$

$$\textcircled{2} \quad \sqrt[3]{30 - 1.1 * (1 - 0.9^2)}$$

$$\textcircled{3} \quad 2 * \left( \frac{(7^2 - 2) * 7^2 * (1011 - 100)}{10^5} \right)^3$$

# Oculus Rift 入門講座

'10 魯鈍ジョン

## 0. この記事の概要

Oculus Rift (おきゅらすりふと) というヘッドマウントディスプレイがあります。これは、バーチャルリアリティー(VR)と呼ばれる、仮想空間を現実っぽく見せるような技術で、これによりゲーム体験をよりリアルに見せることが可能になります。

今回、Oculus Rift で遊ぶ方法、Oculus Rift を DirectX11 で扱う方法について書きます。会誌 DVD には申し訳程度のサンプルを付属しているので参考にしてください。

## 1. Oculus Rift の基礎知識

### 1.1. Oculus Rift の購入方法

9月現在、Oculus Rift にはいわゆるエンドユーザー向け（一般向け）の製品はなく、デベロッパー向けの Development Kit のみ販売されています。日本の輸入代理店で買うとぼったくりな価格になるので、普通に公式サイト注文ページから買ったほうがいいです。

<http://www.oculusvr.com/> （公式サイト）

<https://www.oculusvr.com/pre-order/> （注文ページ）

日本で購入する場合、私が購入したときの価格は\$300+送料\$56.10でした（その時のレートで換算すると 36,976 円）。それに、関税 800 円と通関手数料 200 円が受け取り時にかかります。

支払いには、クレジットカードが PayPal を利用します。VISA が使えるので、V プリカを買ってそれを使うことができます。

## 1.2. 注文ページの書き方

- Order Form
  - Product
    - ◇ Oculus Development Kit – 300.00 USD を選択
  - Qty
    - ◇ 買う個数を入力
  - I understand this hardware~
    - ◇ 開発者向けであることを確認するためのチェックです。  
買うにはチェックを入れる必要があります。
- Payment Method
  - クレジットカードか PayPal かを選び、クレジットカードの場合は必要な情報を入力します。
  - Promo Code
    - ◇ 空欄にしておきましょう。
- Billing Information
  - Country
    - ◇ 国名を選択します。Japan を選択。
  - Company
    - ◇ 個人で買う場合は空欄
  - First Name, Last Name
    - ◇ 下の名前, 上の名前  
(例) 電通太郎さんの場合 : [ Taro ] [ Dentsu ]
  - Phone
    - ◇ 電話番号。国際番号になるので、電話番号の先頭の 0 を 81 に変えて入力します。携帯電話の番号も同じです。  
(例) 電通大 042-443-5000 の場合 : [81424435000]
  - E-mail
    - ◇ メールアドレス
  - Address

- ◇ 住所の一番後ろの部分を、日本の書き方と逆順に入れます。
- ◇ 電通大（東京都調布市調布ヶ丘 1-5-1）の場合：[ 1-5-1 Chofugaoka ]
- Address 2
  - ◇ 建物名とか部屋番号とか。なければ空欄。
- City
  - ◇ 市・郡を入れます。
  - ◇ 電通大の場合：[ Chofu-shi ]
- State / Province
  - ◇ 都道府県を入れます。
  - ◇ 電通大の場合：[ Tokyo-to ]
- ZIP / Postal Code
  - ◇ 郵便番号を入れます。日本のそのままの書き方で大丈夫です。
- Shipping Information
  - Same as billing information
    - ◇ 入力した住所と発送先が同じ場合はチェック。
 そして Continue ボタンを押すことで注文できます。

### 1.3. Oculus Rift の基本原理

Oculus Rift の基本的な構造は、1 枚のディスプレイを、2 つのレンズを通して見るような形になっています。開発版では、ディスプレイの解像度が 1280x800 となっており、これを左右に分割するので、片方の解像度は 640x800 になります。1 枚のディスプレイに、左目用と右目用の画像をレンダリングして表示することで利用できるということになります。しかし、ディスプレイとレンズが近すぎるので、中央はちゃんと見えても外側はぼやけて見えてしまいます。そこで、光学的な補正を行うために画像をゆがませます。理論的な話は第 3 章あたりで説明しますが、基本的にはデバイスからとってきた値を使って計算をして、レンダリングした画像をピクセルシェーダーでゆがませてディスプレイに表示します。

また、頭部の回転を検出するセンサーが搭載されているので、そのセン

サーを利用して 3D 空間のカメラを回転処理させれば、まるでその場にいるかのような素晴らしいバーチャル空間を体感できます。たぶん。

Oculus Rift 対応のゲームを作る際、自力でプログラミングするか、Unity を利用するか、Unreal Engine (UDK) を利用するかに絞られると思います。今回は、修羅の道っぽそうな自力でプログラミングをしてみようと思います。

## 2. Oculus Rift であそんでみよう

すでに、Oculus Rift 向けに作られたデモが結構あります。だいたい Unity 製だったと思います。いくつか紹介するのでダウンロードして遊んでみよう。

・「[OculusWorldDemo](#)」

Oculus SDK 付属のサンプル。

・「[Blue Marble](#)」

<https://developer.oculusvr.com/forums/viewtopic.php?f=28&t=1158>

宇宙旅行っぽいもの。酔いにくい。酸素がなくなったら終了。

・「[Mikulus](#)」

<http://bowlroll.net/up/dl19888>

初音ミクが目の前に現れるデモ

・「[Parrot Coaster](#)」

<http://www.mtbs3d.com/phpBB/viewtopic.php?f=140&t=17848&sid=ab9515b3c26a1b11add96d50d1098b95>

ジェットコースターを体験するデモ

・「[Rift Coaster](#)」

<http://www.theriftlist.com/Home/Game?name=Rift%20Coaster>

ジェットコースターを体験するデモ。終了は Alt+Tab でほかのウィンドウに切り替えてから×を押す。

### 3. Oculus Rift の理論

行列計算を理解している前提で書きます。説明ではすべてのベクトルは縦ベクトルですが、DirectX の場合は横ベクトルになるので行列の掛け算順序が逆になります。 \* の記号は掛け算を表します。

#### 3.1. 頭部回転の適用

センサーから取れる値は、ロール(Z 軸回転)、ピッチ(X 軸回転)、ヨー(Y 軸回転)の 3 種類で、単位はラジアンです。Z 軸回転行列を Roll、X 軸回転行列を Pitch、Y 軸回転行列を Yaw とすると、回転行列 M は、

$$M = \text{Roll} * \text{Pitch} * \text{Yaw}$$

となります (DirectX の場合は当然かける順序が反対になります)。これに、もとのビュー行列 V に対して回転行列を適用した新しいビュー行列 W を求めるとき、

$$W = M * V$$

となります。また、頭部の回転をモデルに適用する際は、回転行列 M の逆行列をとってから掛け合わせます。

#### 3.2. 立体視画像 (ステレオ画像) の生成方法

1 枚のディスプレイに 2 つの 3D 画像を描画するので、ビューポートを個別に設定します。左目用は (0,0)-(640,800)、右目用は (640,0)-(1280,800) とし、それぞれに同じシーンを描画します。と思うじゃん？

同じシーンを描画しますが、微妙に違う視点から見た画像を描画することになります。微妙に視点を変えるとんでも、ビュー行列を変えると計算が面倒になるので、ここはプロジェクション行列を変えます。どう変えるかという、X 軸方向に一定の平行移動行列をかけることで実現します。どの程度の平行移動をすればいいのかは、デバイスから取得した値から計算することで平行移動する量を決められます。

$$h_{\text{meters}} = \frac{(\text{横スクリーンサイズ[m]})}{4} - \frac{(\text{レンズの間隔[m]})}{2}$$

$$h = \frac{4h_{meters}}{(\text{横スクリーンサイズ[m]})}$$

プロジェクション行列 P を、左目用の平行移動行列 Tl、右目用を Tr とすると、左目用のプロジェクション行列 Pl と右目用のプロジェクション行列 Pr は、

$$Pl = Tl * P$$

$$Pr = Tr * P$$

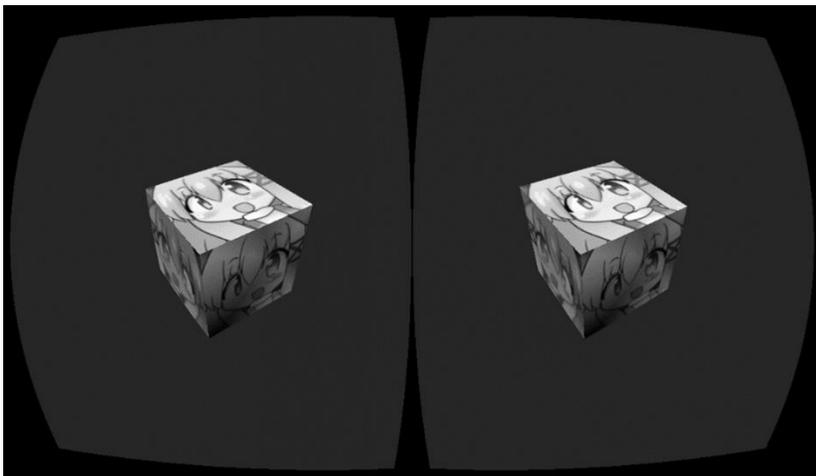
となります。この行列 Pl, Pr を使ってそれぞれの目を描画することで立体視画像を生成できます。

### 3.3. 画面をゆがませる

ここまでで、左目用と右目用の画像がレンダリングされたものができあがります。これをゆがませます。どうゆがませるかということ、テクスチャ座標における、中心の座標からその点への距離を r とすると、移動後の点 f(r) は次のようになります。

$$f(r) = a_0 + a_1r^2 + a_2r^4 + a_3r^6$$

わけわかんないですね。でもこう書いてあるから仕方ありません。これをピクセルシェーダーで処理をします。この中の a<sub>0</sub>, a<sub>1</sub>, a<sub>2</sub>, a<sub>3</sub> はデバイスから取得した値をそのまま入れます。



## 4. DirectX11 を用いた Oculus Rift プログラミング

理論がわかればあとは組み上げるだけです。難しくはありません。面倒かもしれないけど。

### 4.1. 必要なモノ

- Oculus Rift Development Kit
  - 当たり前である。
- Visual Studio
  - 今回は 2010 を使います。
- DirectXSDK
  - 最新版は June 2010 です。適当に落としてください。
- LibOVR
  - Oculus Rift SDK 中にあるはずですが。
  - Oculus Rift の公式サイトからデベロッパー登録すれば手に入ります。
  - この記事を書いている時点での最新版は 0.2.4 です。
  - ダウンロードしたら、ビルドして lib ファイルを生成します。

### 4.2. おおまかな説明

LibOVR を利用する際、「OVR.h」「OVRVersion.h」をインクルードし、「libovr.lib」をリンクします。これで Oculus Rift デバイス固有の値をいろいろ取得できるようになり、頭部の回転もとれるようになります。

従来の 3D アプリケーションに対応させるだけなら、左目用右目用それぞれのビューポートとプロジェクション行列を設定し、ビュー行列に対してヨー・ピッチ・ロールの順番で回転行列を掛けます（第 3 章と逆なことに注意）。最後に、片目ずつピクセルシェーダーに処理をさせてディスプレイに表示します。これだけです。

### 4.3. 実装してみよう

いよいよ本番です。ここでは、LibOVR を利用した実装方法がメインなので、DirectX や XNAMath の細かい部分は各自で調べてください。

#### 4.3.0. 変数

ネームスペース OVR に色々用意されているので使います。

```
OVR::DeviceManager* pOvrManager = NULL;
OVR::HMDDDevice*    pOvrHmd = NULL;
OVR::HMDDInfo      hmdInfo;
OVR::SensorDevice* pOvrSensor = NULL;
OVR::SensorFusion  SFusion;
```

ここでは、OVR::SensorFusion は静的に宣言されていますが、OVR::HMDDDevice オブジェクトが先に解放されているとプログラム終了時に実行時エラーを吐くので、その際は OVR::SensorFusion を先に破棄できるようポインタとして宣言し動的に使用するのがいいかと思えます。サンプルではそうしています。

#### 4.3.1. LibOVR の初期化

```
//Oculus Rift(libovr)初期化
OVR::System::Init();
pOvrManager = OVR::DeviceManager::Create();
if (pOvrManager)
    pOvrHmd = pOvrManager->EnumerateDevices<OVR::HMDDDevice>().CreateDevice();
if (pOvrHmd)
{
    //デバイスの情報を取得
    pOvrHmd->GetDeviceInfo(&hmdInfo);

    //センサーを取得
    pOvrSensor = pOvrHmd->GetSensor();
    if (pOvrSensor)
    {
        SFusion.AttachToSensor(pOvrSensor);
        pOvrSensor->SetRange(
            OVR::SensorRange(4 * 9.81f, 8 * 3.14159265358979f, 1.0f));
    }
}
else
{
    //初期化できなかった
}
```

この通りやってくれれば問題ないです。

サンプルでは、初期化できなかった場合でもデバイスの初期値を適当に代入しておいてあるので、とりあえずは描画できます。

もし、SFusion をポインタで宣言してある場合、new OVR::SensorFusion();でオブジェクトを生成しておきましょう。

#### 4.3.2. LibOVR の解放

```
#define SAFE_RELEASE(x) {if(x) {(x)->Release: (x)=NULL;}}

SAFE_RELEASE(pOvrSensor);
SAFE_RELEASE(pOvrHmd);
SAFE_RELEASE(pOvrManager);
OVR::System::Destroy();
```

SAFE\_RELEASE マクロを使っています。もし、SFusion をポインタで宣言してある場合、SAFE\_RELEASE(pOvrSensor);の前に delete SFusion; しておきましょう。

#### 4.3.3. 頭部回転センサーの取得

```
float eyeYaw, eyePitch, eyeRoll; //グローバル変数

static float LastSensorYaw;
static int SensorInitialized;

OVR::Quatf hmdOrient = SFusion.GetOrientation();
float yaw = 0.0f;
hmdOrient.GetEulerAngles<OVR::Axis_Y, OVR::Axis_X, OVR::Axis_Z>(&yaw, &eyePitch, &eyeRoll);

if(!SensorInitialized)
{
    SensorInitialized = 1;
    LastSensorYaw = yaw;
}

eyeYaw += (yaw - LastSensorYaw);
LastSensorYaw = yaw;
```

SFusion はこういうふうに使います。Yaw は基準となるもの(重力)がないため初期化時に現在向いている方向を正面とします。Pitch, Roll は重力を基準に値が決まるので特に何も考えずそのまま取得できます。

#### 4.4.4. ビュー行列の回転

```

XMMATRIX mView:           //最終的なビュー行列
XMMATRIX mViewRotate;

mViewRotate = XMMatrixRotationY(dg->eyeYaw) * mViewRotate,
              &XMMatrixRotationX(dg->eyePitch) * mViewRotate, &XMMatrixRotationZ(-dg->eyeRoll);

XMVECTOR vEyePt = XMVectorSet( camx , camy , camz , 0.0f);
XMVECTOR vLookatPt = XMVectorSet( lookatx, lookaty, lookatz , 0.0f );
XMVECTOR vUpVec = XMVectorSet( 0.0f , 1.0f , 0.0f , 0.0f);

mView = XMMatrixLookAtLH( vEyePt , vLookatPt , vUpVec ) * mViewRotate;

```

DirectXでは行列の掛ける順序が逆になるといいました。つまり回転行列は Yaw \* Pitch \* -Roll の順番で掛けます。ビュー行列の変換後は原点が基準になるので、そのあとに回転行列を掛けることで回転が適用できます。Yaw, Pitch, Roll がすべて0で（正面を向いている）とき、つまり回転行列が単位行列になるときはそのままのビュー行列になるので、それを理解すれば調整は容易です。たぶん。

#### 4.4.5. ステレオ画像の生成

まずはビューポートを設定します。左目用と右目用それぞれにビューポートを設定し、それぞれにほぼ同じで微妙に視点が違うシーンを描画します。プロジェクション行列を微妙に変化させ、それを使って描画します

```

XMFLOAT4X4 matProjection;           //プロジェクション行列
XMFLOAT4X4 matProjectionLeftEye;    //左目用プロジェクション行列
XMFLOAT4X4 matProjectionRightEye;   //右目用プロジェクション行列

matProjection = XMMatrixPerspectiveFovLH( fovy, aspect, zn, zf );

OVR::HMDInfo hmd = dg->GetHMDInfo();
float viewCenter = hmd.HScreenSize * 0.25f;
float eyeProjectionShift = viewCenter - hmd.LensSeparationDistance*0.5f;
float projectionCenterOffset = 4.0f * eyeProjectionShift / hmd.HScreenSize;

matProjectionLeftEye = matProjection *
                      XMMatrixTranslation( projectionCenterOffset, 0, 0);
matProjectionRightEye = matProjection *
                       XMMatrixTranslation( -projectionCenterOffset, 0, 0);

```

これだけでOK。左目用にはX軸の正の方向に、右目用にはX軸の負の方向に動かします。

#### 4.4.6. ゆがませるためのピクセルシェーダー

ステレオ画像をゆがませるシェーダーはこれです。DirectX11 用です。

```
Texture2D g_Texture: register (t0);
SamplerState g_Sampler : register (s0);

cbuffer global
{
    float2 LensCenter;
    float2 ScreenCenter;
    float2 Scale;
    float2 ScaleIn;
    float4 HmdWarpParam;
};

float2 HmdWarp(float2 in01)
{
    float2 theta = (in01 - LensCenter) * ScaleIn; // Scales to [-1, 1]
    float rSq = theta.x * theta.x + theta.y * theta.y;
    float2 rvector= theta * (HmdWarpParam.x + HmdWarpParam.y * rSq +
        HmdWarpParam.z * rSq * rSq +
        HmdWarpParam.w * rSq * rSq * rSq);
    return LensCenter + Scale * rvector;
}

float4 PS_OculusRift(float4 oPosition : POSITION0, float4 oColor : COLOR0, float2
oTexCoord : TEXCOORD0) : SV_Target
{
    float2 tc = HmdWarp(oTexCoord);
    if (any(clamp(tc, ScreenCenter-float2(0.25, 0.5), ScreenCenter +
        float2(0.25, 0.5)) - tc))
        clip(-1);
    return g_Texture.Sample(g_Sampler, tc);
}
```

3.3.で説明した式まんまのことをやっているだけです。定数バッファでいろいろパラメータを渡してそれで描画しています。いたって普通のことしかしていません。

このシェーダー自体は、Oculus SDK 内のドキュメントに記述があります。それを DirectX11 用書き換え、クランプした後の処理を clip 関数（引数が負の場合はそのピクセルの描画をスキップする）に任せています。

それでは、実際に設定している部分のソースコードを見てみましょう。DirectX11 で書かれたサンプルをそのまま転載しています。DirectX11 の知識がないとちょっと難しいかもしれませんがなんとかがんばってください（投げやり）。

```
//Oculus のデバイスから定数を計算する
OculusRiftSettings ocrSet;

//ゆがみ定数
ocrSet.HmdWarpParam.x = g_hmdInfo.DistortionK[0];
ocrSet.HmdWarpParam.y = g_hmdInfo.DistortionK[1];
ocrSet.HmdWarpParam.z = g_hmdInfo.DistortionK[2];
ocrSet.HmdWarpParam.w = g_hmdInfo.DistortionK[3];

//スケール
ocrSet.Scale.x = 0.5f;
ocrSet.Scale.y = 0.5f;
ocrSet.ScaleIn.x = 2.0f;
ocrSet.ScaleIn.y = 2.0f;

//左目の描画
//スクリーンの中央位置
ocrSet.ScreenCenter.x = 0.25f;
ocrSet.ScreenCenter.y = 0.5f;
//レンズの中央位置
ocrSet.LensCenter.x = ocrSet.ScreenCenter.x +
    (1.0f - 2.0f*g_hmdInfo.LensSeparationDistance/g_hmdInfo.HScreenSize);
ocrSet.LensCenter.y = 0.5f;
g_pImmediateContext->UpdateSubresource( g_pConstantBufferOculus, 0, NULL, &ocrSet,
0, 0 );
g_pImmediateContext->Draw( 4, 0);

//右目の描画
//スクリーンの中央位置
ocrSet.ScreenCenter.x = 0.75f;
ocrSet.ScreenCenter.y = 0.5f;
//レンズの中央位置
ocrSet.LensCenter.x = ocrSet.ScreenCenter.x -
    (1.0f - 2.0f*g_hmdInfo.LensSeparationDistance/g_hmdInfo.HScreenSize);
ocrSet.LensCenter.y = 0.5f;
g_pImmediateContext->UpdateSubresource( g_pConstantBufferOculus, 0, NULL, &ocrSet,
0, 0 );
g_pImmediateContext->Draw( 4, 0);
```

前提として、頂点シェーダーに渡す行列はすべて単位行列で、渡す頂点は(-1,-1),(1,-1), (-1,1),(1,1)の4点でトポロジは TRIANGLE STRIP を指定し、ビューポートは画面全体です。

Scale は出力倍率を設定し、ScaleIn は、入力時に[-1,1]の範囲になるように拡大するように値を設定します。それ以外は「こうやって設定するもんだ」という程度で問題ないと思います。

Oculus Rift 入門講座はおしまいで次から CD 焼き講座になります。

# きれいな CD 焼きをしよう

'10 魯鈍ジョン

## この記事の概要

同人ゲームを作るとき、完成版ならともかく、体験版だと業者に CD プレスの依頼するのちょっと躊躇するよねっていうときに手焼きにしようと思う時があります。せっかく手にとってくれるものだからきれいに仕上げたいときにどうするのかを書きます。たぶん。

## きれいに作るために

CD レーベル面も真っ白でジャケットも用意してない状態でイベント頒布はちょっとアレな感じがします。そこで、「これを使えば綺麗にできる」というのを紹介します。

- レーベル面印刷に対応したプリンタ
  - 筆者は Epson EP-802A を使っていますが今はもっと新しい製品が出ています。
- レーベル面印刷が可能な CD-R/DVD-R
  - おすすめは、右の写真の製品です。
    - ◇ 太陽誘電製 That's 写真画質
    - ◇ 結構綺麗に印刷出来ます。
- スリムケース
  - エレコム製のケースを使っています。
    - ◇ 100 枚パックは取手がついていて便利です。
- ジャケット用紙
  - サンワサプライ製のスリムケース用カードを使います。
- 動作環境表示用のラベルシール（宛名・表示ラベル）
  - ケースのうらに貼り付けます。エレコム製のものを使います。



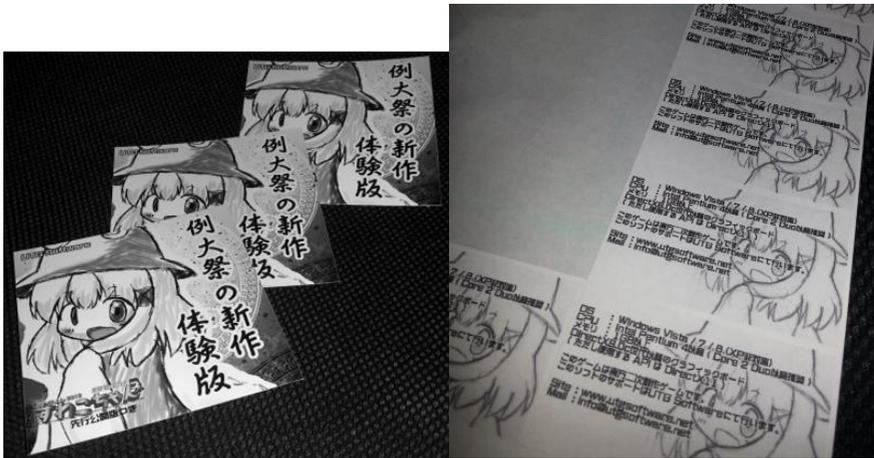
## こんな感じで作業

今回は PC2台使って焼いていきます。CD を焼いてからプリンタに入れてレーベル面を印刷します。だいたい、PC2台とプリンタ1台で同じ速度くらいになると思います。

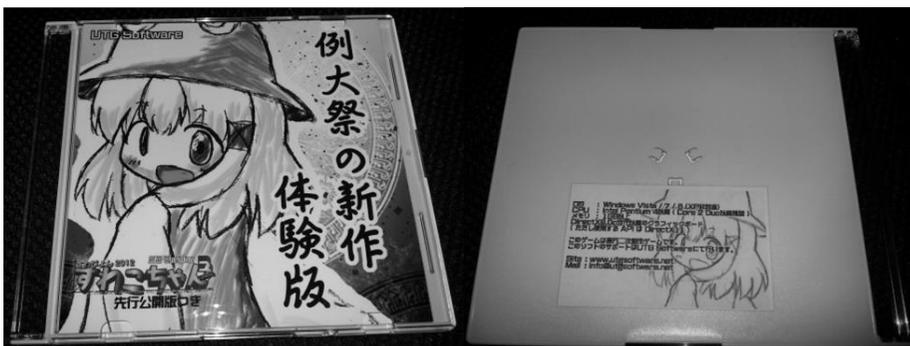


今回は 130 枚ほど焼きましたが、4時間程度かかりました。やっぱり時間かかるなあと。

レーベル面・ジャケット・動作環境表示用ラベルはテンプレートがありますので、例えば Photoshop とかで作って Word で貼り付けて印刷とかやれば作れます。全部の印刷が終われば、いよいよケースに CD を詰めます。



## 完成品





これを枚数分（今回は 130 セット）作ります。

## かかった費用について

このように、きれいに作ろうとすると結構費用がかさむものです。今回の材料はすべて新宿西口のヨドバシで買いましたが、これだけかかっています。

- CD-R メディア（太陽誘電製 That's 写真画質）
  - 50 枚 pack (2,380 円) ×2
  - 30 枚 pack (1,680 円) ×1
- スリムケース（エレコム製）
  - 100 枚 pack (1,680 円) ×1
  - 10 枚 pack (270 円) ×3
- スリムケースカード
  - 10 シート（カード 20 枚） pack (610 円) ×7
- ラベルシール
  - 20 シート（ラベル 180 枚） pack (700 円) ×1
- 送料（重かったので甘えました）
  - 500 円

合計すると 14,500 円になります。1 枚あたり 111.5 円くらいの原価になります。これにさらに電気代・インク代などがかかるのでもうちょっと多くなります。ちなみにこれを 1 枚 300 円で頒布しましたが、完売したもののあげちゃった分も多いので利益自体は数千円程度です。

## あとがき

いつもの人はお久しぶりです。はじめての人ははじめまして。魯鈍ジョンと申します。一個目の記事は Oculus Rift の実装について、二個目の記事は同人ソフトジャンルの同人活動で欠かせない CD 焼きについての解説？を書きました。Oculus Rift に関しては、もしかしたら間違ってる部分はあると思うのでそのときはゆるしてください。なんでもしますから！

ヘッドマウントディスプレイ、とくに Oculus Rift は特に没入感が高く、本当にその世界にいるような錯覚をおぼえます。400ドル以内で買えますし、ジェットコースターのサンプルで遊ぶだけでも十分元がとれると思うくらいすごいのでみんなもぜひ買ってやってみてください。プログラミングは……しらんなあ。

二個目の記事では、さぁみんなて CD 焼きやろう！というところなのですが、この方法結構お金がかかるので頒布価格が上がってしまう欠点はあるかもしれません。きれいに作るとそれだけ手に取られやすくなるというのはあるので、たまにはこういうちょっと手を凝った手作りもしてみると、同人らしくていいんじゃないかなって思います。

某同人作家いわく「500枚までなら手焼き行ける」って言ってたし、筆者の知る限りで東大生の某個人サークルは、エクストリーム会場焼きというコミケ当日の会場にノートパソコンと CD を持ち込んで CD 焼きをしてその場で頒布するというのもやっていたし、同人界隈コワイナーって傍から見てて思います。

ほら、できる気がしてきた。  
いつやるの？ 今でしょ！

# others



# 目指せブレイブルー脱初級者？！

## “ちょっと出来るようになってきた人向け” 講座

編集：はがね (twitterID:@STEEL\_PLUS)

### 1. はじめに

ブレイブルーとは、全国のゲームセンターで稼働している 2D 対戦格闘アクションゲームです。最新バージョンである「ブレイブルークロノファンタズマ」は 10 月に新キャラを引っさげて家庭用として PS3 から発売されました。

ここではブレイブルー好きの私、はがねが独断と偏見によって

「コンボも出来るようになってきて、システムもひと通り目を通した。対戦もするようになった。けど次に何をすれば良いんだろう？」という人向けに、色々紹介して行こうと思います！

\*ブレイブルーにおける基本的な用語は知っている事を前提として話を進めます。

### 2. 同じ相手に何度挑んでも全く勝てない！！

大雑把な悩みですが、これが一番最初に当たる壁であると思います。

理由は大体次のうちのどれかです。

#### ① 単純に力量の差

これはどうしようもないです。格ゲーは自分の中に秘められし何かが発醒して一瞬で上達したりはしないゲームですので、日々の練習あるのみです。

#### ② 動きを読まれている

これが一番多いパターンです。常に同じ動きしかしない相手ほど戦いやすい相手は居ません。

ダッシュして相手に近づいて技を出していなければ駄目なゲームでは無いので、時には一歩引いて冷静に状況を見極めるのも大切です。

#### ③ 自分のミスが多すぎる

これも多いと思います。昇竜を打とうと思ったら 6C になった、なんて経験は誰もが

通る道です。トレーニングモードではコマンドミスしないのに何で？ と不思議に思う方も多いと思います。

理由は、トレーニングモードではコマンドの練習しか行ってないからです。ニュートラルから昇竜コマンドを入力するのと、ガード中から昇竜コマンドを入力するのでは難易度がかなり違ってきます。

トレーニングモードで相手を CPU にして、対戦中のような状況でのコマンド入力精度の練習をしてみると良いでしょう。また、防御中のミスも勝負の決着に大きく影響します。

次項からはプレイブルーにおける重要なテクニックをいくつか紹介します。

## 3. プレイブルーの差し合い・固め・崩し

### ① 差し合い

プレイブルーはリーチの長い技や、置ける小技を持ったキャラが多いので、技を振りながら立ち回る事が基本になります。しかし、技を考えなしに振っているだけでは隙に技を差し返されて相手にターンを奪われてしまいます。そこで立ち回りのテクニックをいくつか紹介します。

#### ・「ダッシュキャンセルバリアガード」

本来ダッシュから停止する際にはダッシュ慣性が乗ったガードが出来ない時間が存在しますが、停止したいタイミングでバリアガードを繰り出すとガードをしながら相手に近づく事が可能になります。急な停止で相手の牽制技を空振りさせるのが主な使い道です。ハクメン、アズラエルなどのステップキャラはダッシュが出来ないのでこのテクニックは使えません。

#### ・「置き JA」

プレイブルーは空中ダッシュからジャンプ攻撃を使った奇襲が強いゲームです。そこでジャンプ A 攻撃を相手が来そうな位置に置いておきます。

空中ダッシュ中はガード不可な事や、奇襲に使う B 系統 C 系統の技の発生が A 系統より遅い事から、相手の奇襲を潰すことが出来ます。隙が少ないので対空技を振るよりリスクは若干少なめです。潰した後はコンボでめて起き攻めまで持っていければ御の字です。

### ・「二段ジャンプ」

自分が空中に居る状況の場合、相手はジャンプ攻撃に合わせて「頭属性無敵フレーム」を持つ 6A、2C などの「対空技」を打ってくる事があります。頭属性を持つ多くのジャンプ技ではどう頑張っても対空技には勝てません。そこで、そもそも対空技に付き合わず二段ジャンプで仕掛けるタイミングをずらしてみましよう。技が空振れば隙が生じます。そこに攻撃を当てに行きましょう。二段ジャンプを嫌がった相手が次はジャンプ攻撃にジャンプ攻撃で反撃してきたり、ジャンプを潜って来る事があります。こうなると読み合いなので状況を判断して適切な対処が必要です。

## ② 固め・崩し

相手に何かの技をガードさせればこちらのターンです(こちらが不利になる技でない限り)。相手の防御の癖を観察しつつ有効な固め・崩しを仕掛けましょう。

### ・「ディレイ固め」

固めの最中の「2A2A2A」や「2A5B5C」などの間に僅かにディレイをかけます。すると、ガード中に A で暴れていた相手(暴れつぶしになる)やレバーを上に入れていた相手(ガード不能のジャンプ移行フレームに刺さる)は崩れます。中段や投げを見せた後、相手は暴れなくなっている筈なので、この択を試してみましよう。

### ・「クラッシュトリガー」

BBCP から追加された新要素で、バリアガード以外ではガードが出来ない全キャラ共通の技です。重要なのは、殆どのキャラにおいてクラッシュトリガーをガードさせた後の状況が五分～微有利であるということです。崩しは相手に読まれないように択をバラけさせる事が重要なので、とりあえず一回は打っておくと良いでしょう。

### ・「当て投げ・遅らせ投げ」

ガード中に ABC を同時押しすることで、バリアガードと投げ抜けを両立させることが出来ます。これを投げ抜け仕込みと言います。相手が投げに来そうなタイミングで投げ抜けを仕込んでいるプレイヤーは多いので、スローリジェクトミスを誘って「2A→投げ」や、「遅らせ投げ」を試してみましよう。投げが通ってスローリジェクトミスの文字が出れば、相手は投げ抜けを仕込んでいます。投げられた相手は仕込みを続けるか止めるか迷います。こうした読み合いを発生させることで有利に崩しを進める事

が出来ます。逆に相手が仕込んでいなかった時は、その場面だけ仕込んでいなかったケースも警戒しつつ直で投げを試みたりして様子を見ましょう。

## 4. ブレイブルーの防御

ブレイブルーは人間には見えないような高速の固め、崩しを仕掛けることが出来るキャラが多数存在します。ガードが上手かったとしても、その他の防御手段でも相手の攻めを凌げなければいけません。

### ・「上入れっぱ」

ダウン起き上がり時やガード中にジャンプ方向にレバーを入れ続ける事です。相手の投げや中段といった択を無効化することが出来ます。当然、起き上がりに技を重ねられた場合や、連続ガードの連携、先述のディレイ固めなどに引っかかってしまうリスクもあります。

### ・「カウンターアサルト」

別名「ガーキャン」。ガード中に「6+AB」を入力する事で、ヒートゲージを 50%消費してノーダメージの無敵技で相手を吹き飛ばす事が出来ます。

ゲージを温存したまま崩されて死ぬくらいなら、使ってしましましょう。ガーキャン入力時にガード中じゃなかった場合は 6B などの大振りな技が暴発してしまうので、隙の少ない投げでフォローできるよう、ガーキャンの入力は「6+ABC」にしておくのが吉です。ガーキャンはカウンターヒットで当てないと相手は空中復帰が可能で、状況は微不利なので注意が必要です。

### ・「ギリギリガードからの行動」

ギリギリガードは別名「直ガ」。通常ガードの直ガ(以降「直ガ」)ではガード硬直が減少し、バリアガードの直ガ(以降「バリ直」)は相手を大きくノックバックさせることが出来ます。方法は、相手の攻撃に合わせてガード入力をするのですが、お勧めは「2入力→ガード」です。ニュートラルからするよりも精度が良いです。

直ガで狙うのは、直ガバクステや直ガ上入れっぱや、直ガ暴れです。

バリ直で狙うのは相手の固めの連携に隙を作ることです。普通なら当たる距離で技を振っているつもりが、ガード側のバリ直により技が空振りしてフォロー不可になってしまいます。そこを突いてジャンプで逃げるなり、小技を差し込んだりで反撃してみましよう。

どちらも普通のガードでは通せない行動が通るようになるので、難しいですがここぞという時に試してみましょう。

## 5. 上達するには…

プレイブルーに登場するキャラクターはどれも強みを持っています。まずは上手い人のプレイを見て、どの行動をどういう風に使っているかをよく観察してみてください。きっと行動ごとにそれなりの理由が見つかるはずです。あとは自分なりに考えて実践の繰り返しです。私はこうやって頑張っています。

また、対戦相手と互いに感想を伝え合うのも大切です。自分では分からなかった癖や、思いもよらない強みが見つかるきっかけになります。

## 6. さいごに

この文章はあくまではがねの私的なチョイスと解釈によって書かれています。テキトーな事言ってる部分も多いかと思うのでご了承ください！

BLAZEBLUEwikiにもっと有意義で詳しい事が書いてあったりするので、ぜひ参考にしてみてください。

それでは良いプレイブルーライフを！



11I 金井

## あらすじ

「そうだ、かき氷を食べよう」

一人が言ったこの一言が、我々をかき氷へと駆り立てた。

普通のかき氷で良いのだろうか。

いや、違う。

普通のかき氷では物足りない。

では何を求める。

そうだ、**天然氷のかき氷**を食べねばならぬ。

そうと分かれば話は早い。早速レンタカーの手配をし人を集め、埼玉県秩父郡長瀬町の天然氷蔵元 阿左美冷蔵<sup>\*1</sup>を目指すことにした。男5人の小旅行である。

## 白き山を目指して

9月某日、我々は大学近くのレンタカー屋を出発し、練馬 IC から関越自動車道を利用して、花園 IC を目指す。当日は平日の昼ということもありそれほど混んではおらず、快調に車を飛ばす。

深谷市を通過し寄居町に到着すると、警察署の前で3人のカエルが我々を出迎えてくれた。今後の運転で事故が起こらないことを祈る。その後は有料道路を通過し、目的地の長瀬町で降りた。駅前に駐車場があったのでここで停車。少し歩くと目的の阿左美冷蔵に到着した。

阿左美冷蔵に到着してまず目に入ったのは行列である。9月の平日の正午にも関わらず、既に何十人もの人が並んでいた。列に居たのは年配の方や家族連れが殆どであったが、よく見ると我々と同じ大学生くらいの年齢の方もちらほら確認できた。どうやら考えることは一緒のようである。渋々列に並ぶと、すぐさま10人程の団体客が我々の後ろに一気に並んだ。少しでも遅れていたなら先を越されていただろう、非常に運が良かった。

さて、目的のかき氷であるが我々を非常に驚かせるものであった。まず高さ30cm程度はありそうな白き山に我々は圧倒された。我々はそれに果敢にも挑みだす。氷は非常にきめ細やかで、我々の手は留まるところを知らずサクサクと進む。しかし食べ進めていくと、そこはかき氷特有の頭痛が起こらない。このかき氷、食べても全然頭が痛くならないのである。筆者は少し



これが天然氷のかき氷だ！！

<sup>\*1</sup> 今回訪れたのは金崎本店。他に山の中腹にある宝登山道店もあるが、こちらは現在営業停止中。

痛くなったが、それでもすぐに引く程度のものであった。天然氷故なのか分からぬが、どこか神秘性を感じさせるものであった。

……少し内容を盛った気がしたが、とにかく非常に美味しいかき氷であった。それは間違いない、うん。

## 河原の冒険

かき氷を食べて満足した一行は、予定に組んでいた長瀬のライン下りを堪能することに。しかし、先日あった台風の影響で川が増水し、残念ながらライン下りの予定は中止。そこでどの程度川が増水しているか気になった我々は、実際に近くの川の様子を見に行くことに。どうみても死亡フラグ\*2である。良い子は決して真似してはいけない。

河原に着いた我々は濁った川を見て、なるほど、これは無理だなと納得。勿論川には入らなかったが、折角なので河原で遊ぶ事に。久々に童心に返り、ひたすら水切りや水の堰き止めをして遊ぶ男子大学生5人なのであった。

## 鍾乳洞を登る

さて、河原遊びにも飽きた大学生一行は次の目的地、秩父市影森にある橋立鍾乳洞を目指す。長瀬町から秩父市まではバイパスで一本の為、殆ど迷うことなく目的地に到着。

さてこの鍾乳洞、タイトル通りなのだが、実は鍾乳洞の中では珍しく縦に長い竪穴である。本当に登っていくのである。入園料の200円を払い鍾乳洞へ。最初は下ったと思ったら、その後は時折梯子を使ってひたすら上へ。鍾乳洞内は狭く、入り口で貸して貰えるヘルメット無しでは危いところであった。内部は湿っぽく、石筍や石柱もよく見られた。菩薩に例えられた鍾乳石も見かけたが……バチとか当たらないのだろうか。中々面白い鍾乳洞であった。

## 浦山ダムへ

時間が余った一行は鍾乳洞近くにあるダム、浦山ダムへと向かう。浦山ダムは堤高156.0m\*3、重力式コンクリートダムとしては奥只見ダムに次ぐ全国2番めの高さを誇るダムである。またダム湖としてさくら湖を有する。こちらも大きい。但し、湖は赤くはない\*4。この事からも非常に大きなことが伺えるが、その後我々はその高さを身を以て体験することになった。

このダム、上から下へのエレベータがあるのだが、それだけではなく階段も存在する。我々日頃特に運動しない男子大学生5人は無謀にもこの階段を登ってみることにした。繰り返すが、浦山ダムは堤高156.0mである。大体東京タワーの階段を登るようなものである。



お手製ロックフィルダム



HASHIDATE 鍾乳洞の前に鎮座する秩父札所28番橋立寺

\*2 実際、長瀬町の荒川沿岸は毎年死者が出ている非常に危険なところである。今回は日数も経っているという事もあり、水量は大したものではなかったが、台風の日には本当に近づいてはいけない。飛び込みとか以外の外である。

\*3 某ロックバンドのhyd O約100人分である。

\*4 SCE社から発売された某ホラーゲームの舞台モデルが秩父だったりする。まあ実際モデルになったのはもっと奥地の方なのだが。



青年は階段を辿る

## おわり

今回の小旅行であるが、様々な発見があり非常に楽しい旅行であった。特に阿左美冷蔵のかき氷は非常に美味しかったので、まだ天然氷のかき氷を食べたことのない方は食べてみて欲しい。少し値ははるが、それだけの価値はあるものだと思うので是非。

また他の場所も色々な発見が出来て楽しかった。今回はライン下りは残念ながら中止となってしまったが、他にも長瀬町の岩畳や秩父市のまつり会館等、見るところはまだあるのでまた行くのも一興かもしれない。

今回は時間が無かったが

次こそは……  
 珍石館\*6 ……  
 珍石館へ行きたい……

と誓ったメンバーなのであった。ちゃんちゃん。

まあ東京タワーと同じくらいなら、と誰一人エレベータを使わずに階段を登りはじめた。単調な階段であり、最初は特に問題なくサクサクと登っていく。しかし登っても終わる気配のない階段を目の前に徐々に絶望が見え始める。最初走って登っていた輩も力果てていき、ダムの高さの半分に達する頃には既に息が上がっていた。なんとか全員登り切ったが、登り切ったときに全員疲れ果てていた。体力不足を感じさせられる結果となった。正直しんどい。

その後は事務所に行ってダムカードを貰った。プラスチック製の何の変哲のないカードであるが、ちゃんとダム毎に貰えるカードは違うのでコレクション要素はある。ダムに行く人は是非とも集めてみて欲しい。ちょおかつこいいぞおたぶん。因みに先日台風がきたという事で、もしかしたらダムから放流が見れるかと期待したのだが今回は見れなかった。残念。

その後は秩父神社を軽く参拝した後には帰宅。行きと違って渋滞に巻き込まれ、レンタカーの返却時刻に間に合うかの瀬戸際であったが、メンバー一人のSAN値\*5を生贄に何とか間に合わせる。帰るまでが遠足、という言葉が久々に感じた一瞬であった。



寄居の守護神

\*5 TRPG「クトゥルフの呼び声」で使用されるパラメータの1つで、「正気度」を表す。つまりSAN値が少ないと正気ではないという事。いあいあ。

\*6 秩父市影森にある、人面石を飾る施設。こちら参照：<http://www006.upp.so-net.ne.jp/chinseki/>

# Single Journey

## 音ゲー入旅

12 屏真太朗

### 合宿先からダッシュ～石打-長岡～

X68の合宿2日目、ちょうどトリプルジャーニー(※1)期間だったので、頑張ってコンプリートしちゃおう！ということで、合宿地からゲームセンターのある長岡までの旅を敢行することにしました。



← 今回の行程。

GoogleMapによれば1時間34分かかる。

JR 石打駅→

近くには上越線を作るために奮闘した方の像や、ド○○もんの像もありました。



この行脚は「ひたすらダッシュした」旅でした。

【ダッシュすることになった理由】

- 16:00にあるミーティングまでに戻らないといけない
- 石打駅付近で迷ってしまい、電車を1本乗り過ごす
  - 次の電車は1時間後
  - 結構時間がなくなり、ミーティングまでの時間がカツカツに
- 長岡駅からアピナ長岡店まで5~6km
  - 悠長に歩いてたら時間がかかる

ということで長岡について途中からダッシュでゲーセンに向かいました。

**上越線発車時刻表**  
Timetable of Joetsu Line

石打駅  
Ishidate Sta.

上り 越後湯沢・水上方面 for Echigo-Yuzawa & Minakami				下り 長岡方面 for Nagasaki			
発車時刻 Time	行先 Destination	のび Track	記事 Remarks	発車時刻 Time	行先 Destination	のび Track	記事 Remarks
6 05	水上	2		6 24	新潟	3	※上越線駅スキー場前通過
7 45	越後湯沢	2		7 20	新潟	3	※上越線駅スキー場前通過
8 55	越後湯沢	2		8 07	長岡	3	※上越線駅スキー場前通過
9 40	越後湯沢	2	※水上行まで運転する日があります	9 06	長岡	3	
10 58	越後湯沢	2	※土曜日・休日及び12/30~1/3運転	10 37	長岡	3	
11 42	水上	2		11 31	長岡	3	※土曜日・休日及び12/30~1/3運転
12	—	—		12 26	長岡	3	
13 43	越後湯沢	2		13 18	長岡	3	
14 57	水上	2		14 27	長岡	3	
15 44	越後湯沢	2		15 22	長岡	3	
16 37	越後湯沢	2	※水上行まで運転する日があります	16 26	長岡	3	
17 41	水上	2		17 17	長岡	3	
18	—	—		18 49	長岡	3	
19 06	越後湯沢	2	※水上行まで運転する日があります	19	—	—	
20 04	越後湯沢	2		20 38	長岡	3	
21 07	越後湯沢	2		21 51	長岡	3	※上越線駅スキー場前通過
22 09	越後湯沢	2					

お知らせ



石打駅構内と駅時刻表。  
ほぼ1時間に1本。

近くでお祭りがあってい  
たようで、駅前には出店も  
数軒ありましたし、ちょん  
ど子供神輿とも遭遇する  
ことができました。

## つぶやき

途中の六日市駅の近くにメジャーロード六日町店があったのですが、こちらは今回の目的のゲーム GITADORA(※2)がないため今回はスルーしました。今度の合宿の時にいきたいですね！



↑ 今回のルート（イメージ図）。橋のあたりからダッシュ。

本当に久々にマラソンをしたのでゼイゼイ言いながら何とかアピナ長岡店に到着。しかし、この時点でかなり時間が迫ってました。

本当は式寺(※3)もやりたかったけど、グッと堪えてGITADORAをプレイ。1回で解禁終了して、急いでアピナ長岡店を後にしました。



←アピナ長岡店。中は結構広い。式寺やGITADORA以外にも各種音ゲーが揃っていた



← 戻る途中、ダッシュに疲れてバスを利用。  
結果的にこれに乗らないと電車に間に合わなかった。

帰りに乗った電車を見送る。石打駅にて。→



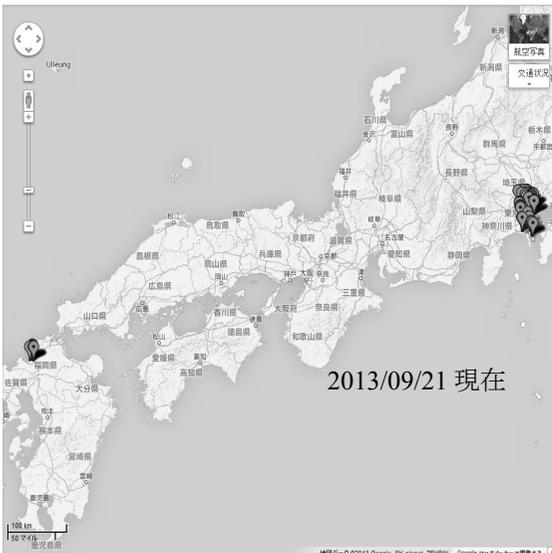
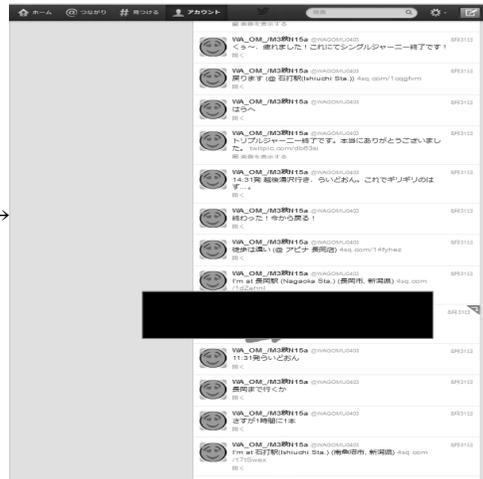
そんなこんなで私のどたばた音ゲー旅は終わりました。今回思ったことは、「旅は時間に余裕を持ってやろう！」ということでした(当たり前)。

ダッシュで戻ったところ、まだミーティング前で無事に戻ることができました。帰ってきて思っていたのは「とにかくすぐに風呂に入りたい！」でした。

# つぶやき

1時間に1本しかない各駅停車での旅でしたが、乗客はそこそこ（1車両に4~5人）いたように感じます。  
 おじいちゃん、おばあちゃんはもちろん、子供連れや女子高生など地域の足として活躍してるのを感じました。良いですね。

道中の私の Twitter。  
 あまり呟いてないけど、かなり焦っていた。→



← 現在の式寺行脚 MAP。  
 帰省していたので福岡も回っている。

その後も行脚(※4)し続けた結果  
 ・ 4 エリア  
 ・ 94 店舗  
 ・ 104 台  
 となりました。  
 (この前調布アドアーズで1000店舗超  
 してる皆伝さんがいたので、さすがで  
 すね……)

## 【その後】

他にもたくさん行脚しました！

- 「事業所見学の帰りに」京急本線沿線行脚
- 「完全徒歩でテクテクと」三軒茶屋～下北沢～明大前行脚
- 「行脚に夢中でアイマスライブに遅刻しました！」福岡遠征行脚

などなど……。ちょこちょこ備忘録とか書いていきたいですね！

駄文ではありましたが、読んでいただきありがとうございました！

## ～注釈表～

この記事に出てくる※をつけた単語について、分かりにくそうなところを補足しておきます。

1. トリプルジャーニー
  - コナミのアーケード音楽ゲームでのイベントのこと
2. GITADORA
  - コナミのアーケード音楽ゲーム。ギターとドラムを模した筐体を使って遊ぶことができる
3. 式寺
  - コナミのアーケード音楽ゲーム beatmania IIDX の略称
4. 行脚
  - 様々なゲームセンターを回っていくこと。beatmania IIDX にはゲームを行った店舗数およびその店舗があるエリアを記録する要素があり、それをランキング表示するシステムもある。そのためゲームセンターを目的に全国を飛び回るプレイヤーもいる。

# 華麗なる泥棒術

11M fumakura

## 1. 概要

風来のシレン(\*1)というゲームをご存知だろうか。風来のシレンとは、チュンソフトが開発・発売しているローグライク(\*2)ゲームで、1995年にシリーズ第一作がスーパーファミコンで発売されてから現在に至るまで、多くの人間を楽しませ続けている。

今回は本シリーズで、「ダンジョンにランダムで出現する店から、お金をかけずに商品を奪い、次の階まで逃げ切る」というローグライクの華、「泥棒」の方法を考察した。

本シリーズをプレイしたことのある初級者～中級者向けの記事となっているが、可能な限り用語の解説はしていく。なお、今回検証に用いたのは、最新タイトルである「風来のシレン5 フォーチュンタワーと運命のダイス」である。

## 2. 仕様

泥棒をする際、最初に考えなくてはならないのが仕様である。仕様を知っていれば対処の仕方にも幅が利くが、知らなかった場合・勘違いしていた場合は一転、高い確率で死に至る。

### 2.1 泥棒

①いかなる理由でも(\*3)「商品を購入しないまま店を1歩出た場合」

「紹介状を持たないまま高級店に侵入した場合」に泥棒扱いされる。

②店主を倒した状態で店を出ても泥棒扱いされる。

③泥棒後は現在フロアにいる全ての敵が店主に変化し、その後出現上限である19匹になるまで店主が一気に湧く。

以降は店主を倒しても5ターン後に新たに湧く。

### 2.2 店主

①店主のステータスは高く、序盤では1発殴られたら確実に死ぬ。

②黄色い店主は倍速1回攻撃である。

③ステータス異常は無効化されない。一部の即死攻撃は無効化する。

④聖域の巻物(\*4)は効かないが、ギタン投げ(\*5)は効く。

### 2.3 他

①罨は、移動して踏んだ場合作動しないことがある。足元で作動させれば確実に作動する。

②水路に吹き飛ばされると同じフロアのどこかにワープする。

などなど。他にも、アイテムの仕様やその階に出現する敵など、知識は多いに越したことはない。

(\*1)何故シリーズの原点である「トルネコの大冒険」でも最近人気の「ポケモン不思議のダンジョン」でもはたまた「ディアボロの大冒険」でもなく、風来のシレンなのかといえば、単純に筆者が一番好きだからである。

(\*2)ターン制 ARPG、と云えば分かりやすいだろうか。レベルが引き継がれなかったり、マップが毎回違うことが大きな特徴。名前は1980年に公表されたRogueに由来する。

(\*3)敵のせいでも。

(\*4)足元に置くと、すべての物理攻撃を無効化する巻物。遠距離攻撃は防げなかったり、ボス級には無効化されたりと、頼もしいのか頼もしくないのか分からないアイテム。

(\*5)ギタンとはお金のこと。ギタンを敵に投げて当てた場合、その1/10のダメージを与えることができる。

(\*6)割ると、足元に容量分の落とし穴の罨を作る。

(\*7)シレン5に登場。アイテムを入れると山彦状態になる。山彦状態とは、魔法が反転される状態である。リフレク。

### 3. 基本テクニック

#### 3.1 底抜けの壺

底抜けの壺(\*6)を投げて落とし、落とし穴から落ちる。恐らく一番安全かつメジャーな泥棒方法である。安全すぎて逆に面白くない。底抜けの壺を見つけたらできる限り取っておきたい。ただし、登ってゆくタイプのダンジョンでは使用できない。

#### 3.2 山彦香の壺+一時しのぎの杖

山彦香の壺(\*7)を焚き、店主に向かって一時しのぎの杖(\*8)を振ることで、階段の上に移動することが可能である。ただし、この場合はかなしばり状態を解除する手段が必要。一発攻撃に耐えられるようにしておくか、空腹状態(\*9)で行うとよい。

### 4. 少し危険 or 高度なテクニック

事故を回避する為に気配察知の腕輪(\*10)があると良い。

#### 4.1 会いたくて震える

店の出口から階段まで一直線上にあり、障害物が間になれば、飛びつきの杖(\*11)で階段上に移動できる。ただし、階段が壁に面していない場合はクッション用に土塊の杖(\*12)で壁を作っておく必要がある。また、必要に応じて店から階段部屋までを掘削できるアイテム(\*13)を使うことになる。

#### 4.2 バトルドーム

身代わりの杖のパチモノである身代わりの杖は、振ると進行方向左から丸太が飛んで来て、10マス吹っ飛ばすという性質を持っている。これを利用して、10マス先の階段にシレンをシュウウツするという方法。山彦香と吹き飛ばしの杖(\*14)でも代用が可能である。どちらの場合も、必要に応じて店から階段部屋までを掘削できるアイテムを使うことになる。

とにかくいかに早く移動するかが重要な泥棒の場面では、これらの道具による移動は大いに役に立つ。直接階段に向かうために使う以外にも、水路上に吹き飛ばすことで他の部屋にワープすることが出来るため、これによる階段部屋への移動にも用いることが可能。

#### 4.3 栄光のビクトリーロード

仕様 2.1-③ を利用し、予め 18 匹モンスターをどこかに隔離、閉鎖しておく。後はウィニングランである。

隔離の方法としては、土塊の杖で部屋を塞ぎ、そこにスポンするのを待つという方法が挙げられる。敵は 30 ターンに一回、シレン

(\*8)魔法弾が当たった者を階段上にワープさせた後、かなしばり状態にする。かなしばり状態とは、一切の行動が出来ない状態である。ダメージを受けると解除される。

(\*9)多くのローグライクでは、HPとは別に空腹度ゲージがあり、これが0になると1歩歩度に1ダメージを受けるようになる。餓死が一番こわい。

(\*10)装備すると、敵の居場所がマップに表示される。

(\*11)魔法弾が当たった場所まで飛びつく。

(\*12)つちくれ。使用すると前方に壁を作る。

(\*13)トンネルの杖、つるはしなど。

(\*14)魔法弾に当たった者を10マス先まで吹き飛ばす。

(\*15)投げたアイテムが壁やモンスターを貫通し彼方へ消える状態。投げたアイテムの効果は当たった全てのモンスターに有効。遠投状態のまま合成の壺を投げてしまう失敗は恐らくほぼ全てのプレイヤーがやらかしていると思う。

(\*16)魔法弾が当たった相手と場所を入れ替える。

のいない部屋でスポーンするので、隔離開始からの経過ターン数/30-隔離部屋以外に湧いたモンスターの数=18 で全ての敵が隔離部屋にスポーンしたことになる。

#### 4.4 肩力S

遠投の腕輪を装備するか、ドラゴン草を飲んで遠投状態(\*15)にし、階段に向かって場所替えの杖(\*16)を投げることによって階段上にいたモンスターと位置を交換する。あらかじめ階段上にモンスターをかなしばりしておくが良い。ただし、店を出て泥棒状態になると金縛りしたモンスターが店主に化けてしまう(=金縛りが解ける)ので、場所替えの杖を投げるのは店の中からでなくてはならない。

### 5. 確率に愛された人のためのテクニック

オススメはしない。

#### 5.1 賭博黙示録

バクチの巻物(\*17)を読む。成功した場合でも5フロア移動してしまうので、「泥棒さえ成功すれば後はもうどうでも良いや」という人にオススメ。

確率に愛された人間なら成功するんじゃないかな。

#### 5.2 ホールインワン

高飛び草(\*19)を飲み、階段に一発着地を狙う。

確率に愛された人間なら大丈夫だと思う。

#### 5.3 アレキサンダー大王

階段までまっすぐ徒歩で向かう。

「無敵を信じた『アレキサンダー大王』は何千と飛んでくる矢の中を平然で歩いたと言うが」って吉良吉廣(\*20)も言ってたし、確率に愛された人間なら大丈夫だと思う。

### 6. 通路でのテクニック

今まで紹介した方法は、ほとんどが「道具が揃っている」「店から階段までの距離が近い」時にしか使えないものであり、多くの場合これらの方法を実践することは難しい(\*21)。実際には、ある程度対策をした上で徒歩で向かうことが多い。その際に一番の問題となるのが、通路での鉢合わせである。

泥棒において、後退することは多くの場合状況を悪化させる。階段からも遠ざかってしまうために、成功率も著しく下がる。よって、鉢合わせの際にいかにしてやり過ごすのが泥棒成功の鍵を握る。ここでは、その方法を紹介する。

(\*17)読むと、10LVアップ・5フロア移動・フロアの魔物をどれか1体根絶やし(\*18)・フロアの魔物が全員二倍速・所持ギタン全消失・手持ちの武器防具が全て1段階成長・自爆のうち、どれか1つ効果を発揮する。

(\*18)その冒険の間に限り、投げて当てた相手と同じ種のモンスターは出てこなくなる。店主種にねだやしを投げた場合、フロア中の店主は消滅するが、ターン経過で再び湧いてくる。

(\*19)同じフロアの別の部屋に移動する。

(\*20)ジョジョの奇妙な冒険 40巻参照

(\*21)序盤ではアイテムが識別(\*22)されていないことも多く、十分な対策が出来ない。終盤では泥棒してる余裕なんてない。

(\*22)入手したアイテムは、見た目や名前だけではどのアイテムか分からない物があり、その状態のことを「未識別」という。未識別アイテムを実際に使用して識別を行うことを「漢識別」、店の売値での判断を「値段識別」と言う。また、識別の巻物などを使用して識別することもでき、これが一番確実である。

## Q. 通路でのすれ違い

図1を見て欲しい。白丸がシレン、四角が店主である。この時、使えるものに以下のアイテムがある。

- ① 混乱の巻物(\*23)
- ② 睡眠草(\*24)
- ③ 飛びつきの杖
- ④ かなしばりの杖(\*25)

あなたならどうするだろうか？

### 解答例

間違っても混乱の巻物なんか読んではいけない。攻撃される危険はできる限り避けなくてはならない。

正解は、「飛びつきの杖による角移動」である。まず図2の位置まで下がり、店主をかなしばりの杖で動けなくする。その後左上に向かって飛びつきの杖を使うことで、比較的安心(\*26)にすれ違うことが可能である。

また、図2の時点でシレンの左上にも店主がいることを確認できた場合は、金縛り後に1歩下がることで後続の敵の索敵から外れ、さらに金縛り状態の店主を敵は壁と認識するため、後続の店主は何ターンかしてからリターンする。その後、慌てずゆっくりと階段に向かえば良い。

## 7. まとめ

今回挙げたテクニックは、どれも(わりと)基本のテクニックである。だが、これだけ覚えていれば後は十分に応用が効くはずである。泥棒のみならず、ローグライクというジャンルは試行回数を重ねることがそのまま上達につながる。何回も繰り返していけば、今置かれている状況に対してどのように対処すべきか、というのは感覚で分かってくるはずである。だからこそ皆さんには是非、習うより慣れるの精神で泥棒ライフを充実していただきたい。

泥棒を練習するならば、原始に続く穴がオススメである。店の出現率が高く、更に10F/25F/50F/75Fで確実に店が出現すること、「もっと不思議のダンジョン(\*27)」であることが理由に挙げられる。

それでは、良い泥棒ライフを。

## 8. 参考

<http://wiki.livedoor.jp/shiren5/> (風来のシレン5 Wiki)



図 1

(\*23)読むと、隣接する敵が混乱状態になる。混乱状態とは、ふらふらと移動し続け、移動先にキャラクターがいた場合に攻撃を行う状態である。

(\*24)飲むと自分が睡眠状態になる。投げ当てると相手が睡眠状態になる。

(\*25)魔法弾が当たった相手を金縛り状態にする。

(\*26)飛びついた先に敵がいれば終わりである。敵がいらないことを祈る他ない。

(\*27)アイテム持ち込み不可/レベルリセット/ほとんどのアイテムが未識別/全99F構成のダンジョン。ストーリーダンジョンとは比較にならない難易度で、プレイヤーの実力が色濃く出る。名前の由来はトルネコの大冒険1と2の「もっと不思議のダンジョン」より。



図 2

# THE ・ 闇カレー

～おいしいカレーを作りたい ( )～

著作 パワーハ

## ◆ はじめに

去年、闇鍋をした俺たち一同は今年も新たな刺激を求めていた。鍋の次に闇にできるものは何があるだろうか……と。悩んだ挙句思いついたのがカレー。

よし！ 闇カレーを作ろう！ ということでこの夏休みに集まったのだ。

## ◆ 買いたしだ———！

まずは食材を買い出しに行った。学校の近くの西友で各々が直感的にカレーに入れる食材を買う(実はこれが一番おもしろい)。もちろん、他の人の材料は知らない。思わぬ食材の組み合わせがより深い闇カレーを生み出すのだ。

しかし、カレーは何にでも合うということで有名だ。会誌のネタがなくなるのではないのかという不安と、今回は大丈夫そうだという安心感が現場にはあった。

だが、まさかあんなことになるとは……。

## ◆ 作るぞ———！！

一人暮らしのメンバーの家を借りてさっそく作り始める。順番で一人ずつ台所に行き、食材を入れていく。後半になればなるほど闇のカレーになっていくだろう。

このタイミングでそれぞれが買った食材の写真を撮った。写真は次のページに載せてある。もちろん、ここではまだお互いの食材は知らない。

なお、3回作りました。

◆ 食材紹介



ネギ味噌、ポテコ、食パン、さつまいも  
豚巻ウィンナー、チーズ

もやし、バナナ、とろろ、タコ、  
七味唐辛子、シナモン、オレガノ



チーズ、福神漬け、かにかま、しらす、  
(きのこの山)



ポップコーン、抹茶ラテ、アクエリア  
ス、ホワイトチョコ、シナモン、ココナツ、  
コーヒブッセ、生ハム、タコウィンナー、  
ベーコン、



ひでえ

◆ 1 回 目 の カレー



どうだ、見た感じおいしそうだろ。うん、見た目だけなんだ……。

味はひどいものだった。カレーのくせに甘い。甘いといっても**変な甘さ**。辛いのに甘い、甘辛いじゃないなくて、辛いかつ**変な甘さ**。具の**バナナ**を食べた時は**きつかったな……**。

そして、人によって**味が違う**！ 食べられる人がいる一方で、まったく食べられない人も。食べ比べてみたら確かに違ったよ……。カレーってこんなにはっきりと味の層が分かれていたものか？

もう、食いたくないよこのカレー……。

※もちろん、スタッフがおいしくいただきました。

### **主な戦犯**

シナモン、バナナ、抹茶ラテ、ホワイトチョコ、コーヒブッセ  
つまり、甘い系

## ◆ 2 回 目 の カレー



1回目のひどさから、さっそく一同は闇カレーにおののいていた。つ、次は食べられるもの作ろうよ！みなは誓った。

……で、できたものがこれ。キ、キーマカレー!?

一同は恐る恐るカレーを口に運んだ……。……。ん？ 普通だ！ でも味がない。カレーは風味だけ。なんだよこれ、液体のところすらぶつぶつする。すべてが具。味がないが。でも、悪くはない。1回目と比べたら断然食える。味がないけどね。

完食間際でこの味と似たものを思い出す。そうだ、カップヌードルのカレー味の汁を薄めたやつだ！ そう言われればそうである。まあ、味がないほど薄いけど。

※スタッフがおいしくいただきました。

### 主な戦犯

ポテコ、ポップコーン、ココナッツパウダー  
でも、他と比べたらましかな。

◆ 3 回 目 の カレー



時間はすでに深夜。なぜ闇カレーを3回もやってしまったのか……。

最後においしいカレーを作ろうと誓いあい、できたのがこれ。  
ルーを入れる前すでにカレー色になっていたのは気のせいなのか……。

さっそく一口。ん!? 甘い、甘いぞ!

1回目の変な甘さとは違う。もっとちゃんとした甘さ。すっきりしている甘さ。  
というより、カレーの味がしないぞ。

ここでネタばらし! なんとアケリアスを入れやがったのだ。分量間違えただろ  
言われてみればそうだ。この甘さはまさしくアケリアス! 現場では深夜のためか、  
『アケリアス食ってる時にカレーの話するなよ!』と謎ギャグが受けてしまう始末。

## 主な戦犯

# アケリアスの粉

## ◆ まとめ

今回、集まって分かったことは、ロックマン2は難しい。深夜から始めて、終わったのが次の日の昼。もう眠い眠い。よくクリアできたなど思う。特に大変だったのは、クイックマンステージとワイリーステージ1かな。他には……。

……え、闇カレー？ ナニソレマズイノ？ うう、カレー、カレー……うわあああああ

はい、闇カレーの話ですね。もう、記憶から封印したいほどだよ。3食も連続で普通じゃないカレーだったせいか、当分カレーは食べたくなかった。ちょっと恐怖を感じていたのかもしれない。今は大分ましになってきたけど。

でも、この闇カレーで得たものは後ろ向きな成果だけではない。この闇カレーを食べた一同が感じたことがある。いつも食べているカレーはなんておいしいものだったのかと……。カレーだから何でも合うのではない。作っている人が上手いからそのカレーは何でも合っておいしいのだ。また、いつも食べていたおいしいカレーに感謝。今後、学食のカレーを食べた時に涙しながら食べられそうだよ。本当はカレーってこんなにおいしいのかって。しかも値段もリーズナブルだ。なんと、この闇カレー3食セットは3000円以上かかっているのだ。

読者の方も、カレーなどの食べ物を食べる時は、作った人と自然の恵みに感謝して食べよう。いつもの食べているカレーはとてもおいしいものなのだ。

さて、次の闇はどうしようかな。

## ◆ 参加者

パフーハ、ナスビ、クレイジー、3548

# I A M T R E N D

'10 rinatsu

## 君は流行に乗れているか？

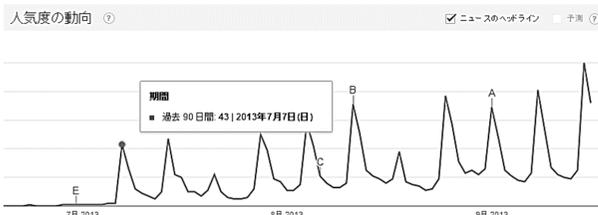
流行。またの名をトレンド。近頃はTwitterの普及により、恐ろしいスピードで物事が流行り、そして廃れていく。半日Twitterを覗いていなければ、タイムラインの流れがつかめないこともザラだ。ぼくもクッキーを焼きながら[1] 記事を執筆しているが、明日は何食わぬ顔で提督業[2] に復帰しているだろう。君は流行、追えているだろうか？

## Google Trends, それは世界を暴く System

流行に乗り遅れてはまずい。そこで、世の中を知るための道具の1つとしてGoogleトレンド[3]を提案しよう。Googleトレンドは、Google検索で検索されているワードを調べることができるサービスだ。その日の頻繁に検索されているワードを知ることができれば、その日にどんな出来事があったのかが分かる。世の中の人々が何に注目していたのか分かるという訳だ。Googleトレンドの優れたところは、急上昇ワードだけでなく、検索ワードを自由に指定しその検索頻度の推移をグラフで可視化できる点だ。さらに、期間・地域などの指定もできるため様々な条件で調べられる。聞くところによると、Googleトレンドを投資戦略に利用しようという試み[4]もあるらしい。

2013年現在で利用できる主な機能は以下の表のようなものだ。

急上昇中	その日の検索ワードの多いものを順番に並べる
ランキング	様々なジャンルにおける人気トピックをピックアップする ※2013年9月時点では米国の傾向しか見れない
調べる	検索ワードを入力し、そのワードの検索状況を調べられる地域、期間などの条件付きでグラフ化が可能。 グラフは相対値なので注意、CSV形式でエクスポートもできる



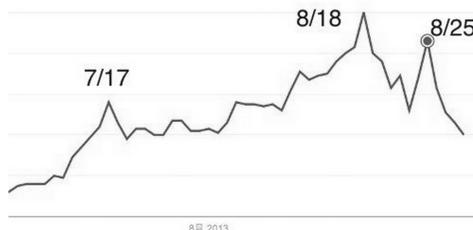
◀ 検索ワード「半沢直樹」について検索数の推移をグラフ化したもの。毎週の放送日にピークが立っており、放送されていない週は検索数が少ないのが分かる。

## 実際に何か調べてみよう

試しに、何か調べてみよう。さて、右のツイートに注目してもらおう。ツイートによると、検索ワード「自由研究」について面白い結果が得られたようだ。



ダ・ヴィンチ・恐山 @d\_v\_osorezan 9月1日  
 「自由研究」の検索数グラフによると、7月17日に第一のピーク（真面目な子）が来て、徐々に検索数を増やしながら8月18日に第二のピーク（真面目と不真面目の境目）を迎え、8月25日に第三のピーク（クズの領域）を迎えることがわかる。 pic.twitter.com/r2E7FL4VL5  
 画像を非表示にする 返信 リツイート お気に入り その他

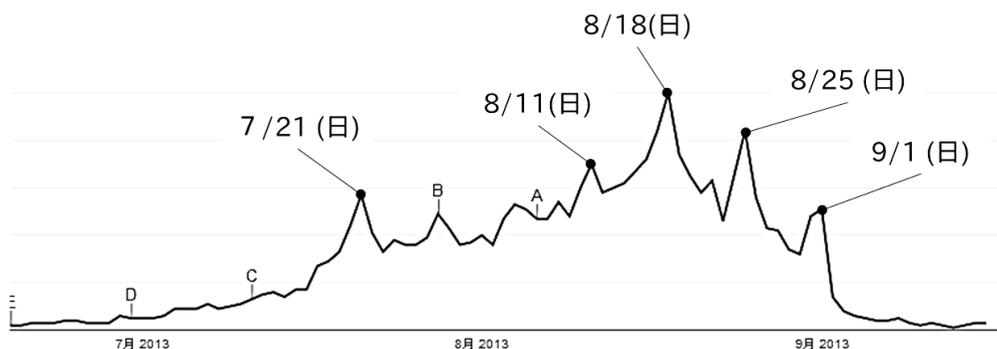


▲偉大なる元ネタ様 [5]

### これは悪乗りするしか無い！

夏休みが終わった今だからこそ、「自由研究」で検索してみようではないか！ どのような結果が得られるだろう。楽しい夏休み、どうしても宿題は後回しになってしまうが……

## カツオはまだ宿題やってないし、余裕っしょwww



検索ワード「自由研究」の検索数の推移を示す。この結果を見るに、真面目に宿題をやっている層は7月の下旬には自由研究のネタを探している、ということだろう。次に8/18（日）に第二のピークがみられる。何も準備していなかった層が焦り始める時期なのだろう。8/25（日）の第三ピーク、ここが夏休み宿題戦線の最終防衛ラインだろう。これ以降の挽回は厳しい（勝手な憶測）。

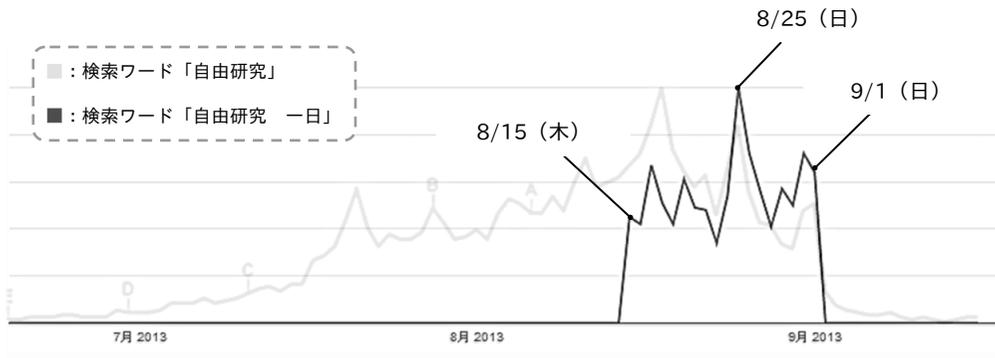
また、ピークが日曜に集中しているのも面白い。夏休みといえど、サザエさん症候群の影響か。はたまた、仕事が休みで家に居る親御さんに急かされているのだろうか。

やべーよ，やべーよ…… 夏休みあと一週間だよ……

グラフからダメな生徒諸君が一定数存在するのは明らかである．そこで，検索ワードに一工夫することで，彼らを炙りだしてみよう．さて，「自由研究 一日」と検索ワードを変更した場合，検索数の推移はどうか？ 一日で忌々しい自由研究を倒したい，いや倒さなければならない．求められるは，一日で終わる「超軽量自由研究」の存在．彼らは一縷の望みを託し「一日で終わる自由研究のネタ」を探し求めるのであった……

多分

では，検索結果を見てみよう．



なんということだ (困惑)

まさか，ここまで綺麗にデータが出るとは想像だにしていなかった．8/14(水)までは検索指数0で推移しているが，翌日の15日になった途端，検索指数は45に跳ね上がる．そして，25日でピーク（検索指数100）を迎え，9/2（月）以降，検索指数0で推移している．この素直なグラフを見るに，夏休み最後の日は9/1（日）か？ 9/2（月）以降，ほとんど検索されていないことから，生徒諸君はなんとか一日で自由研究をでっち上げたのだろう．いや，そうであってほしい．

2013年 8月

日	月	火	水	木	金	土
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

◀今年の8月のカレンダーだ．25日の時点で自由研究が終わっていなければ焦るのは間違いない．

## まとめというか、自由研究は早めにやろう

簡単にGoogleトレンドの機能を紹介し、最後に実際に検索結果をした。無料で利用できる割にかなり有益なサービスなのは間違いないだろう。かなり適当に調べただけだったが、それでも面白い結果が得られたのだ。検索ワードによっては、期間だけでなく地域的な偏りも見られるだろうし、色々と応用することが出来るかもしれない。

最後に、自由研究はもちろん、夏休みの宿題はゆとりを持って取り組もう。夏休みの宿題が終わらないからといって塾をサボることのないようにしましょう、**本当にやめて**。

1日

自由研究

自由研究に役立つ工作や実験がいっぱい！  
カンタンにできるテーマもあるよ！

← 工作 こうさく
← 実験 じっけん
← 調べる しらべる
← 家庭科 かいてい

**★ おすすめ**



キラキラ万華鏡(まんげきょう)



ペットボトルけんぴきょう



家族の指紋(しもん)を調べてみよう



カップケーキのピンクッション

**✂️ 工作 こうさく**



牛乳(にゅう)パックのミニじょうろ NEW



カラーモールでミニカゴ作り



組(か)まごのからでエコまじり絵



ビニールたこ

▲ユーザーのニーズに合致したサービスの提供 [6]

### 参考

- [1] Cookie Clicker <http://orteil.dashnet.org/cookieclicker/>
- [2] 艦隊これくしょん～艦これ～ <http://www.dmm.com/netgame/feature/kancolle.html>
- [3] Googleトレンド <http://www.google.co.jp/trends/>
- [4] Google Trendsは投資戦略に使えるか <http://japan.zdnet.com/sp/enterprise-trend/35031563/>
- [5] ダ・ヴィンチ・恐山 (@d\_v\_osorezan) . “ [https://twitter.com/d\\_v\\_osorezan/status/373903094980956161](https://twitter.com/d_v_osorezan/status/373903094980956161) ” 1 Sep. 2013, 5:20 (JST) Tweet.
- [6] 1日のできる自由研究 [http://kids.nifty.com/kenkyu/index\\_one.htm](http://kids.nifty.com/kenkyu/index_one.htm)

# SOUND VOLTEX BOOTH 初心者講座

## ～みーつま攻略指南書☆ミ～

'12I くものひと

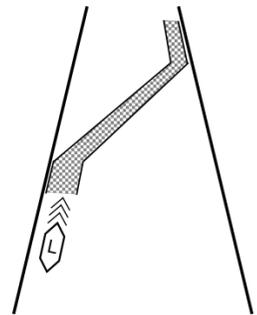
### 1、はじめに

皆様は音楽で競い合ったことはおありでしょうか。現在各地のゲームセンターにて、KONAMI より稼働中の音楽体感型ゲーム SOUND VOLTEX は、新感覚アナログデバイスである「つまみアクション」で、簡単に誰でも競い合うことができる。しかしこのつまみ操作には酷くクセがあり、巧みに操ることは難しいと思われる。そこで今回はそんな奥様方の悩みを解決すべく、解説記事を作成するに至ったのである。是非活用してもらいたい。

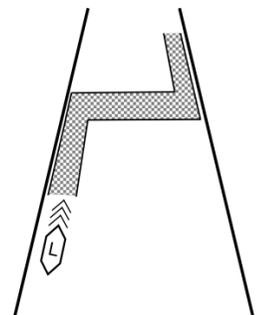
### 2、つまみって何!?

つまみとはコレ⇒(右写真参照)である。これをリズムに合わせてキュインすればよい。そのタイミングは画面が指示してくれるのだが、その表示が(図 1)である。画面下の青や赤のポインターを、それぞれのレールの上を走らせるように、つまみを操作すればよい。この場合、左のつまみを 時計回りに回せばいいんですね、分かります。次に、(図 2)を見てほしい。一見不可能のように思えるが、実はこれも 同様に時計回りに回せばよいのだ。ちなみに、回す速度は速くても遅くてもどちらでもよい。(図 2)の場合でも、ちょこっと回すだけでもちゃんとポインターが働いてくれる。なんとも頼もしい。

しかし、ここで(図 1)において注意すべき点が二つある。一つ目は、つまみを回すだけではなく、回し『続け』なければならない。つまり、先端から終端までつまみをクルクルし続ける必要があるのだ。二つ目は、つまみを過剰に遅い速度で回した際に起こる脱線である。これは初心者にありがちなミスであり、つまみが操作に反応できないために起こる現象だ。これらに気を付ければ、キミもクールなミーツマジョッキーになれるだろう。



(図 1)



(図 2)

### 3、回し方あれこれ

前項にて、つまみとの基礎的な戯れ方を学んでいただけたと思う。しかし実際にプレイしてみると、つまみの回し方に手こずり、ポインターがあっちゃこっちゃに SKY HIGH してしまう。そのようなことがならないよう、様々な回し方を紹介しよう。ぜひとも一つ一つ試してみるとよいだろう。(写真はタイマーサイトにて撮影しました)

#### ① ネジ回し

ネジを手で回す際に、ネジの頭をつまんで回す方が多いと思う。親指,人差し指,中指などの先端でネジをつまみ、手首をひねり、掴み直す動作を繰り返してネジを締める。これと同じ要領でつまみを回す方法がネジ回しなのだが、実はこのやり方は**オススメできない**。この方法を実際に行くと、つまみがいちいち止まってしまい、回転を持続させることができない。このため、ポインターがレールから外れてしまいやすいのだ。これ以外の方法を推奨する。



#### ② 肘回し

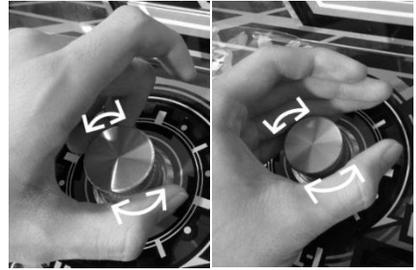
肘回しといっても、つまみを肘に挟むのではない。せやな。つまみに手をかぶせるように親指,人差し指,中指で深く、しっかりと持つ。そして、あまり手首を使わずに腕全体、特に肘を意識して回す方法である。この方法の強みは、安定した速度でつまみを回し続けることができる点だ。また、逆方向に回し直す際にも優秀な面が見られる。デメリットとして、回す角度に限界がある点が挙げられる。常に一方向に回し続けていると、肘の角度に無理が生じてしまう。こういった場合には、再度持ち直すか他の回し方に移行するか、どちらかが必要だ。



#### ③ ペットボトル回し

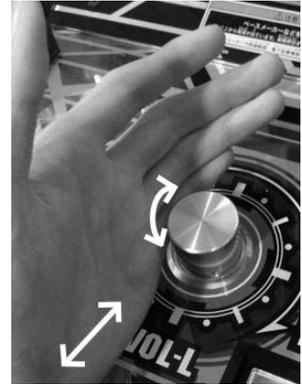
ここにペットボトルがあるとする。あなたならどうフタを開けるだろうか。いやまあ開け方なんて色々あるけども。私はフタの縁を親指でなぞり回し、次に人差し指や中指や薬指などの横の部分で同様になぞり回す。これを繰り返し、フタを開ける。

これを SOUND VOLTEX に応用した回し方が、ペットボトル回しである。分かりづらい。この方法の利点は、一方向に永遠に、永久に回し続けられることだ。欠点は回転に安定感を持たせづらい点である。特に逆方向に回す際に、細心の注意を払う必要がある。



#### ④ ろくろ回し(中級者向け)

ネーミングセンスの無さに泣ける。つまみの縁に小指の付け根辺りを押し当て、そのまま手を押し引きすることによって回す方法である。この方法の利点は、肘回しと同様に回転に安定感を持たせることができ、逆方向に回転させやすいことだ。また、高難易度に挑戦する際に、ボタンから素早くつまみへ手を移動させやすい点もある。欠点はやはり回転量に限界がある点だ。これにより、やはり持ち直しか回し方の移行が必要である。



## 4、まとめ

以上が私の推奨するつまみとの和解指南である。しかし、これらの方法が絶対というわけではない。重要なのは全てをマスターすることではなく、自分に適した方法を模索し、つまみとの有意義な時間を過ごすことである。

上記の回し方を試行してもうまくいかないことが多々あるだろう。そのような時にこそ落ち着き、状況を把握しよう。回し始めるタイミングが早すぎたか？ 回す速度が遅すぎたのか？ 逆のつまみを回していないか？ 回す方向を間違えていないか？ これらを意識して改善していくことにより、つまらないミスは 少なくなるだろう。何事もチャレンジが大事さ。とりあえずまあこのゲーム面白いんで、試しにやってみれば良いと思うよ。

**THANK YOU FOR READING**

**THIS TO THE END!**

# アルファ\*1を目指すためにやったこと

## 方法1

○パクリ bot\*2を作ってフォロワー\*3を増やす



まず面白いネタが好きそうな人をフォロー\*4していきま  
す。次に以下に示すようなパクリ bot からネタをパクって  
つぶやきます。



後は凍結\*5されない範囲で自動フォローしていけ  
ば、RT\*6されていてフォロワーが増えていくハズ。  
フォロワーが増えたところでアカウント名変えれば  
アルファになれるのではないのでしょうか。

## 結果…

\*1 Twitter で影響力のある人です。

\*2 ツールを使って自動でつぶやくアカウント。

\*3 自分の咳きを見ている人です。アルファになるためにはこの人数が多い必要がありますが、多いからといって必ずしも影響力がある訳ではありません。

\*4 あるアカウントの人の咳きを自分のタイムライン上で見られるようにすることです。フォローすると大抵その人に通知が行くので自分の存在を気付かせることができます。

\*5 アカウントが使えなくなること、即ち BAN です。

\*6 リツイート。他人の咳きを自分のフォロワーに見せることです。フォローしてなくてもできます。自分の咳きが RT された数が Twitter における影響力だと考えて大丈夫です。



フォローし過ぎということはないと思うのでスパム報告

\* 7されたのかもしれませんが。

## 方法2

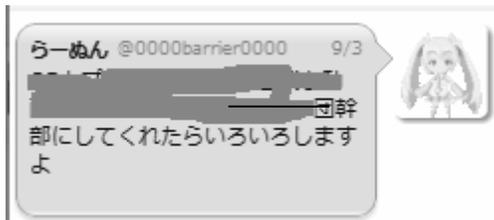
### ○アルファに媚びる



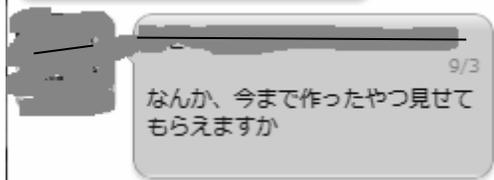
アルファはフォロワーが多いので自分のツイートを RT してくれるとたくさんの人に見られます。

そうすればフォロワーが増えるかもしれません。

～団幹部にしてみよう



アルファの中には「～団」など作り団員を集める人も居ます。幹部になれば団員にフォローして貰えるはずです。



めんどくさい

媚びても RT されないし幹部にもなれなかったので失敗。

\* 7 規約違反をしていたり、スパム行為を行っている悪いアカウントだと運営に報告すること。ブロック（あるアカウントと関わらないようにする機能）されても凍結します。

方法3

○自分でアカウント作成、フォローしまくる



アカウントを連番でたくさん作りすぎると凍結します。気をつけましょう。勿論失敗です。

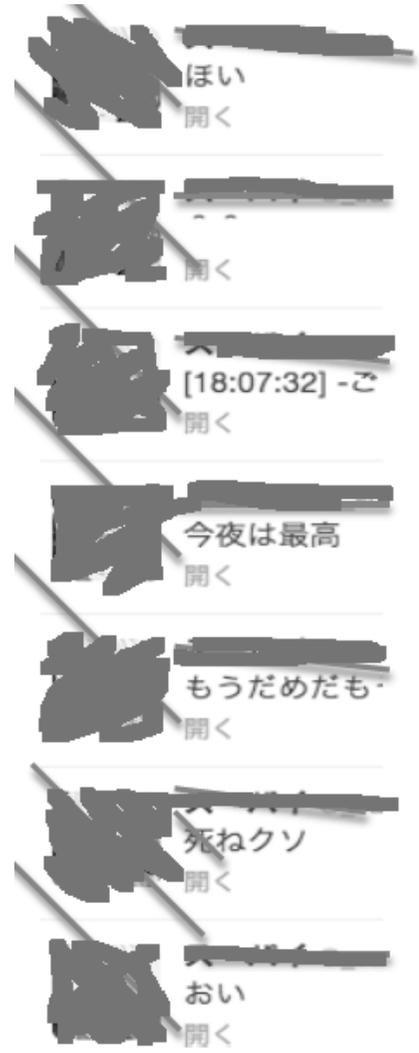
方法3

○他の人と協力してフォローを盛る



クソアルファになるのもチョロいです。このやり方を広めて電通大総アルファ化計画を始めようかと思われました。

とりあえずフォロワーが多いというだけのアルファは多いです。



※画像は無関係のものです。

実際のアルファはいつでも面白いことを言っているわけではありません。よってフォロワーが多いという理由だけで更にフォロワーが増える可能性を検証します。



# あの

なんと下がっている。その通り、所詮は見せかけのフォロワーだったのです。

## \*まとめ

よく考えたら本当のアルファにたった2週間でなれるはずがありませんでした。

アルファはクソアルファなりに昔からコツコツと売名してフォロワーを蓄積して今があるわけですし、面白いアルファだって一朝一夕に今の地位を築けた訳ではありません。

手っ取り早く名実共にアルファになってフォロワーを稼ぐには、やはり何かの製作者になるというのが一番ではないかと思います。

つまり既に他のコンテンツで売名し終わっている人たちが有利ということです。ちなみにツイッターでネタクラスタとしてアルファになっている人はフォロワー1.5~2万程度です。

恋愛ツイートなど一般受けが良さそうなツイートをしてアルファになっている人はフォロワー5万程度 of アルファをよく見ます。

一般にもヲタクにもウケたある人は1週間でフォロワー10万を越しています。

フォロワーを無駄に数万も盛っている人も多いこの世の中なので一概には言えませんが、一般人にもウケるツイートができる人のほうがRTもお気に入りもフォロワーも稼げるのではないかと思います。

# 遊☆戯☆王 ろっぱ式No.杯

～No.ハンターは1人！この俺だ！！～

## 1. はじめに

この記事は、当サークルの夏合宿で行った特殊ルールによる、遊☆戯☆王OCGのトーナメント形式での大会「遊☆戯☆王 ろっぱ式No.杯」を、後日その場にいた面子でレポートしたものである。

## 2. 特殊ルール

- ・エクシーズ召喚の際、相手と共有の、OCG化されている全No.で構築されたNo.エクストラからランダムにモンスターを選びエクシーズ召喚する。

- ・RUM等でCNo.を召喚するときには、同様に構築されたCNo.エクストラから、ランダムにモンスターを選び、エクシーズ召喚する。

- ・お互いのフィールド上の「No.」と名のついたモンスターは、以下の効果を得る。『「No.」と名のついたモンスター以外の相手のカードの効果の対象にならず、「No.」と名のついたモンスター以外との戦闘では破壊されない。』

- ・《ガーベージ・ロード》などのエクシーズ素材とする場合の制約は無視できる。

- ・《No.88 ギミック・パペットーデステニー・レオ》と《No.91 サンダー・スパーク・ドラゴン》のエクシーズ召喚時、コントローラーは手札を任意の枚数、エクシーズ素材に追加できる。

- ・《No.53 偽骸神 Heart-ear tH》の3つ目の効果を発動する場合、No.エクストラから《No.92 偽骸神龍 Heart-ear tH Dragon》をエクシーズ召喚できる。

- ・融合モンスターとシンクロモンスターは、全ての効果が無効になり攻撃することができず、召喚したターンの終了時にゲームから除外される。

## 3. 参加者紹介

**満足**：本大会の主催者。使用デッキは【ギミック・パペット】。

**BTB**：徐々に米版統一に近づく【聖騎士】を愛用する。

**semmel**：使用デッキは【ガエルランサー】。ポセイドラが強く見える。

**くりボー**：クレインやエクソを軸とした【ランク4】を使用。

**A**：全てのNo.を提供してくれた御仁。【ガガガ】を使うリアルNo.ハンター。

**キノコリ**：エクシーズも多用する【リチュア】を使う。

**yasainoyama**：遊戯王初心者だが、えげつない【暗黒界】を使いこなす。

**D・Y**：バックに容赦の無い【ガジェット】で参戦。

**purin**：特別枠。エクシーズ？なにそれ美味しいの？

4. 試合レポート

**満足**：本題に入るまでが長すぎやしないかって？ ルール説明が重要なんだからしょうがないじゃない！ 正直ルールの調整が一番大変だったんだぜ。No.の活躍を主体としつつ、それぞれのデッキの個性を損なわないように……。

一同：早く進めろよ。

**満足**：(´・ω・`)

～1回戦～

A【ガガガ】

VS

D・Y【ガジェット】

**満足**：幕開けの第一試合！ この場にD・Y氏がいないからA氏に語ってもらおうか！ どうだった、この試合？

A：No.がたくさん出たから良かったよね。みんなすぐ死んだけど……。

くりボー：折角出てきたNo.がことごとくミラフォやら奈落やらブラホで散ったからな(笑)。

yasainoyama：対象とらない除去は強いですねー……。

**満足**：ルール決めた時点でこうなるのは想定してたし……(震え声)。

A：あと見どころは、俺が《RUM-バリエーションズ・フォース》打ったところでしょ！

**満足**：《No.12 機甲忍者クリムゾン・シャドー》が《CNo.32 海咬龍シャーク・ドレイク・バイス》にカオスエクシーズチェンジ！ その後すぐミラフォで退場して行ったけど(笑)。

**semmel**：あとは、途中で《No.56 ゴー

ルドラット》が《トラゴエディア》のアドバンス召喚のリリースに使われたりしたな。

A：決着をつけたのはD・Y氏が引き当てた《No.61 ヴォルカザウルス》だったし、No.もピンキリだなあ……。

**満足**：と言うわけで勝者はD・Y【ガジェット】であった。

～2回戦～

**semmel**【ガエルランサー】

VS

**満足**【ギミック・パペット】

**満足**：私と**semmel**氏の対決だったが、とりあえず初手に《No.6 先史遺産アトランタル》引いてがっかりした。

A：【ギミック・パペット】と言いつつ、実際は《レベル・ステイラー》でエクシーズ乱発してたけどな。俺の【ガガガ】もだけど。

**semmel**：その後、僕が《No.53 偽骸神 Heart-earth》出して即《シー・ランサー》のリリースにしたな。

A：それが後にああなるとは……。

**semmel**：墓地へ送っておいたHeart-earthをリビデで釣り上げて自壊させ、現れよ！ 《No.92 偽骸神龍 Heart-earth Dragon》！！

一同：おおおおお！？

**満足**：一応加えておいた6つ目のルールが使われるとは……。

**semmel**：実はずっと狙っていたのさ！

**満足**：こちらは《No.54 反骨の闘士ライオンハート》と《No.9 天蓋星ダイソン・スフィア》で対抗。途中で出てきた《N

o.42 スターシップ・ギャラクシー・トマホーク》とか言うお荷物は《強制転移》でプレゼントしつつ、お互い戦闘を行えず試合は膠着状態に。

**semmel**：その均衡を破ったのは《No.19 フリーザードン》。素材が無かった偽骸神龍の効果を乱発できるようになった！

**A**：そして、試合は最終局面へ……。

**semmel**：偽骸神龍とライオンハートの効果で、お互いのライフも残り僅かという場面で引き当ててしまったのは《No.40 ギミック・パペット・ヘブンズ・ストリングス》。

**満足**：ここでその時の盤面をご覧ください。こう。(下図)



上：**semmel** 残り LP1400

下：**満足** 残り LP1100

**semmel**：偽骸神龍の攻撃が通れば勝てるのだが、ダイソンへの攻撃は止められてしまう。

**満足**：かと言ってターンを回せば、ヘブンズ・ストリングスがダイソンの効果発動条件を満たしてしまっているため、直接攻撃で私が勝つ。

**A**：つまり導き出される結論は……。

**semmel**：行け！ ヘブンズ・ストリングス！ 『ヘブンズ・ブレード』！！

**満足**：迎え撃て！ ライオンハート！ 『バ

ーニング クロスカウンター』！！

**BTB**：ライオンハートの効果。このカードの戦闘によって自分が戦闘ダメージを受けた時、受けた戦闘ダメージと同じ数値のダメージを相手ライフに与える……。

**semmel&満足**：うわああああ！（LP→0）

一同：えええええええ！？

**semmel**：まさかの引き分けに終わったという。

**満足**：これは想定していなかった……。とりあえず、厳正なる抽選の結果 **semmel**【ガエルランサー】が勝者となった。

～3回戦～

**yasainoyama**【暗黒界】

VS

**キノコリ**【リチュア】

**満足**：おい！ この大会の主旨無視するなよ！ ガストクラークで初手ハンドスとかやってんじゃねーよ！

**キノコリ**：手札に揃ってたんだからしょうがない。その後エクシーズして《No.42 スターシップ・ギャラクシー・トマホーク》出したし。

**yasainoyama**：まあ、トマホークはブラホで破壊して、そのまま押し切っちゃいましたけど。

**キノコリ**：トラゴにヴェーラー打たれたのがアカンかった。

**満足**：と言うわけで勝者は **yasainoyama**【暗黒界】。あっさり終わりすぎだ……。

～4回戦～

**BTB**【聖騎士】

VS

くりボー【ランク4】

BTB：いやあ、《H・C エクストラ・ソード》は強敵でしたねえ。

くりボー：とか言って《No.15 ギミック・パペット-ジャイアントキラー》で全部吹き飛ばしてくれたじゃねーか！

満足：このルールでのジャイアントキラーは最強クラスの性能だなあ。

くりボー：それでも序盤はこちらが押してたな。《No.69 紋章神コート・オブ・アームズ》が制圧してくれた。

BTB：その横に無駄にエクストラソード素材にした《No.49 秘鳥フォーチュンチュン》居たろ。忘れんなよ（笑）。

キノコリ：エクソチュンチュン（ATK1400）（笑）。

満足：コート・オブ・アームズをミラフォで処理し、返しに現れたのはもう一体の最強クラスのNo.。

BTB：そう！ 《No.11 ビッグ・アイ》降臨！ が、フォーチュンチュン効果の対象にとれないじゃん！！

A：ここに来てまさかの活躍（笑）。

くりボー：時間稼ぎにしかならなかったけどなー……。

満足：その後引いたNo.もビッグ・アイを倒せるものではなく、最後に《No.42 スターシップ・ギャラクシー・トマホーク》を奪われて決着。BTB【聖騎士】の勝利が決まった。

くりボー：ビッグ・アイ強すぎんだろ……。

BTB：同じランク7でも格が違うな！

～準決勝1回戦～

semmel【ガエルランサー】

VS

D・Y【ガジェット】

満足：波乱の1回戦を勝ち残った両者の対決！ どんな感じでした？

semmel：No. 同士が殴りあって、最終的にブラホに全て飲み込まれた回だったな。あと地砕きが強かった。

満足：また汎用除去かよガジェット！ そういうデッキだからしょうがないけどな！

A：でも、とどめ刺したのは《No.15 ギミック・パペット-ジャイアントキラー》だったからね。やっぱ強いなこいつ。

満足：全くだ。ジャイアントキラーの効果で状況を一瞬でひっくり返し、semmel【ガエルランサー】が勝利を収めた。またあっさり終わってしまった……。

～準決勝2回戦～

yasainoyama【暗黒界】

VS

BTB【聖騎士】

満足：この試合は最もNo.の引きの強さが勝敗に影響した回だったな。

yasainoyama：《No.25 重装光学撮影機 フォーカス・フォース》一体倒すために6回エクシーズしましたから……。

BTB：《No.42 スターシップ・ギャラクシー・トマホーク》と《No.63 おしゃもじソルジャー》と、二回連続で攻撃力0のNo. 引いたのには笑った。

yasainoyama：その後引いた《No.54 反骨の闘士ライオンハート》も神警で止めら

れて、次の《No.15 ギミック・パペット  
トージャイアントキラー》でようやく処理  
した頃には、僕のデッキ残り5枚だったん  
ですけど。

BTB：正直デッキ切れ待っても勝てたけど、  
折角《No.91 サンダー・スパーク・ド  
ラゴン》が出たから派手にとどめ刺させて  
もらったよ。

満足：終始《No.25 重装光学撮影機フ  
ォーカス・フォース》が何故か場を制圧し  
て、BTB【聖騎士】が長期戦の末、勝利を  
もぎ取った。

～決勝戦～

semmel【ガエルランサー】

VS

BTB【聖騎士】

満足：決勝戦ですけど……。

BTB：聖剣しか引けなかった（^ p ^）。

semmel：何か相手がモンスター出してこ  
ないから適当に殴ってたら勝てた。

満足：……BTB氏。ちょっと表出てくださ  
い。

BTB：俺自身に聖剣が装備できれば……！

semmel：と言うわけで、僕が優勝したけど  
……、決勝がこれでいいのか……。

～決着がつきましたが、ここで《No.42  
スターシップ・ギャラクシー・トマホーク》  
の華麗な戦績を振り返ってみましょう～

・2回戦

強制転移後、両者引き分け。

・3回戦

ブラホで破壊された後、敗北。

・4回戦

最後の最後に出てきて、敗北。

・準決勝2回戦

時間稼ぎするだけして、敗北。

・まとめ

No.中最多登場4回において一勝も無し。

・結論

《No.42 スターシップ・ギャラクシ  
ー・トマホーク》をコントロールしたプレ  
イヤーは勝てない。

なんだこの呪いのNo.。

～EX マッチ～

semmel【ガエルランサー】

VS

purin【ニードル・デストラクション  
針虫の狂宴】

満足：決勝があつたのでEXステ  
ージ突入だ！ ろっぱ式No.杯を制し、全  
てのNo.を手中に収めた semmel氏は、果  
たしてエクシーズを狩る者 purin氏に勝つ  
ことができるのか！？

semmel：この大会、そんな設定だったんだ  
……。

purin：そんな異名持った覚えないんだけど  
……。

満足：いいからデュエルしてください！

purin氏はエクストラ要らないでしょう。

デッキ名からして。

purin：まあね。

semmel：試合の方は、《No.54 反骨の闘  
士ライオンハート》が《ニードルワーム》  
を倒せなくて15枚くらいデッキ削られた  
な。

くりボー：そして引き当ててしまった絶望

のNo.《No.42 スターシップ・ギャラクシー・トマホーク》……。

**満足**：もうだめだ……おしまいだ……。

**A**：しかし、そこで現れた新たなNo.!

**semmel**：絶望を吹き飛ばせ！ その幸運で我が手に勝利を！ 出でよ！ 《No.7 ラッキー・ストライプ》！！

**キノコリ**：これなら勝つる！

**semmel**：サイコロで7は出なかったが、打点としては十二分！

**満足**：《No.48 シャドー・リッチ》、《No.40 ギミック・パペット-ヘブンズ・ストリングス》を加えての一斉攻撃で、

**semmel**【ガエルランサー】が辛くも勝利を手にした。

### 5. 振りかえり

**満足**：《No.42 スターシップ・ギャラクシー・トマホーク》出過ぎだろ……。8試合中5回だぞ。ホントに呪われてんじゃねえのこいつ。

**A**：ちなみに、次点が《No.40 ギミック・パペット-ヘブンズ・ストリングス》の4回だな。こっちは結構活躍してる。

**BTB**：まあ3000打点ってだけで十分使える。

**キノコリ**：それにしても、《No.39 希望皇ホープ》と《No.32 海咬龍シャーク・ドレイク》が1回も出てないのは惜しかったよね。CNo.も《CNo.32 海咬龍シャーク・ドレイク・バイス》しか出てないし。

**満足**：俺も一応《RUM-ヌメロン・フォース》入れてたんだぜ。引けなかったけど。

**semmel**：あと、このルールだと《No.8 紋

章王ゲノム・ヘリター》も最強クラスだったけど誰も引いてないな。

**yasainoyama**：登場回数にはバラつきがありましたね。できればもう少しやりたかったところです。

**くりボー**：確かに、まだ出てないNo.が10体くらいいるし、No.が増えればまた改めてやる価値はあるな。

**満足**：まだ見せられてないギミックとかもありますしね。《ナンバーズ・ウォール》引けなかった……。

**A**：《ガガガガール》の効果使えなかった……。

**くりボー**：《H・C エクストラ・ソード》2体でオーバーレイできなかった……。

**BTB**：はいはい。

**満足**：と言うわけで、アニメのスフィアフィールドのルールを意識してやってみた、「遊☆戯☆王 ろっぱ式No.杯」でした。意外と程良いゲームバランスで進行できたので、機会と全No.があれば是非やってみてください。

では、次はどんな大会が開かれるのか。今後のX680x0同好会決闘班の活動にご期待ください！



2013年9月1日大会当時の使用可能No.カード数。No.:60枚。CNo.:9枚。

# アカウントが凍結された時の話

by 暮

## ●はじめに

世は情報化社会、色々な SNS が普及していますね。その中でも特に電通大生が大好きなものがあります。わかりますね。そうです、「Twitter」です。そんな「Twitter」で私自身がした、特別な体験について述べさせていただきます。

## ●事は期末テスト前の休日

7月下旬のテスト前、しかも休日。そんな時は本来なら長時間勉強に励むのが理想ですね。ですが、私のような学生にはそのようなことは理想だけで終わってしまうのです。私はテスト期間突入前に「とあるブラウザゲーム」に嵌ってしまっていました。当然、私はゲームという誘惑に敗北し勉強することを忘れてました。

その「とあるブラウザゲーム」をプレイしていくうちに、ある難所をクリアしました（ポケモン金銀で言うジョウトのバッチ全部集めたよー、ぐらいのもの。一つの区切りです）。今思えばそこまで難所でも無かったのですが、嬉しかった私は「やったあああああ 2-4 クリアしたよおおおお!!!!1111」と Twitter で呟こうとしたのです (via web)。そしたらですね、見慣れない通知が。

「あなたのアカウント (@\*\*\*\*\*) は現在、凍結されています。詳しくは……」

へ？

## ●そもそも凍結って？

(Twitter においての)凍結とは、つぶやき、タイムライン(リスト含む)の閲覧が出来なくなる状態のことを言います。本来凍結は、3段階のステップがあります(以下 web 版の話)。

まず最初に警告の通知がホームに表示されます。凍結される危険のあるアカウントに通知が来るそうです。そして第2ステップで、アカウントの凍結が行われます。凍結は、運営に申請を送ることで数日から、長くて数週間で解除することが出来ます。遂には最終段階でアカウント停止。新しくアカウントを作らない限り Twitter に復帰することは出来なくなります。

で、今回私がされた凍結は1つ段階を飛ばした第2ステップの「凍結」だったのです。

## ●何したら凍結されるの？

普段、凍結される一番の原因はスパム報告を多く受けているアカウントです。スパム報告を受ける人については深くは語りませんが、有害そうなアカウントはスパム報告を受けてる印象が強いです。

他には「人間離れした行動」を取るアカウントも対象になっているようです。bot や特殊なアプリケーションを使用して、普通では出来ないようなことをしているアカウントのことです。

## ●何故、私は凍結されたのか

まずスパム報告を受けるようなアカウントなのかを判断する必要があります。しかし、私のアカウント名をここで晒すのもアレなので読者の方には今から書いてあることを信じてもらえないです。自分で言うのも変ですが、割と一般的な Twitter の使い方をしていると思っています。基本は趣味や日常のことを呟いているだけです。ツイート数も平均 17 ツイート/1 日と少ない方です(電通生ツイッター使用者の中では、全ユーザーの中だと 10 ツイート/1 日を超えると多い方らしいですが……)。

もう1つ凍結の原因と成り得る「人間離れした行動」ですが、いくつかアプリケーションを使ってはいましたが、特別派手なことをするものは使っていませんでした。

### ●ちょうどこの時期に……

当時 Twitter をやっている人で経験した方もいると思いますが、突然フォロー・フォロワー数が何人か減るとい現象が起きていました。その原因こそが、「アカウント凍結騒動」です。全く身に覚えのない人が突然アカウントを凍結されるという事態になっていました。運営がスパム業者のアカウントを潰すための対策なのではないか、と考えられていましたが未だにその詳細な原因は不明となっています(2013年9月時点)。(詳細:9月14日閲覧 NAVERまとめ【注意】Twitter運営がスパム業者対策を強化しアカウントが凍結される騒ぎが発生中 <http://matome.naver.jp/odai/2137482186717062301>)

(以下一部憶測です。)

### ●本当に身に覚えがない人が凍結されてる?

もし、本当にスパム業者を潰すためにこの騒動が起きていたのなら、全く身に覚えがない人が凍結されるなんてことは想像し難いことですね。無作為に凍結しても何も意味がありません。何かしら原因があるから凍結されていると考えた方が自然です。なので今回の騒動は「凍結する基準が下がった」と考えていいでしょう。

### ●騒動を踏まえて、じゃあどうして凍結されたのか?

凍結される原因として、「人間離れした行動」を取るアカウントは凍結の対象になると、先に述べました。私はフォローを整理できるアプリケーションを以前使っていたことがあり、それでフォロー数を増やしたり減らしていたことがありました。それでもフォローの増減は目立つほどでもなかったのですが今回の騒動で凍結のハードルが下がり、「人間離れした行動をとるアカウント」と判断されて凍結されてしまったのだと思います。

そして、凍結解除申請を運営に送って3日ほどしたら凍結が解除されました。私以外の騒動の被害に遭った人も丁度その日を境に凍結が解除されたようです。現在、騒動は完全に収まっています。

### ●凍結されて感じたこと

豆知識なのですが、凍結されてすぐにタイムラインが見れなくなるわけでは無かったです(リストは最初から見れなかったです)。ある程度時間が経つとタイムラインが流れるのが止まって、見ても意味が無いという状態になりました。

やたら無闇にアプリケーションは使うものじゃないなと思いました。些細なことでも凍結対象になることがあるのでアプリケーションの導入、使用には気を付けた方が良いでしょう。人に迷惑をかけなければ凍結されないという訳でもないというのがよくわかる出来事でした。

それと薄々とは感じていましたが凍結されると不便ですね。そんなに使ってないと思っていましたが、通学中や休憩中の暇潰しが無くなってしまうのは困ります。幸いにも土日を含んでいたので少しマシでしたが。

で、凍結されてた訳だしテスト勉強捗ったよね! ね! ……お察ください。

Thank you for reading! (´ω`)

# つり野郎

いーはちえる

## 釣りがしたい

2013年、X680x0 同好会に空前の釣りブームが訪れた。ある者は大学から程近い多摩川へ、またある者は海へと、次々に釣りをしに出かけた。

「この流れに乗るしか無い」

そう思った私達は、ついに行動を起こすことにした。

## いざ阿佐ヶ谷

8月某日、私達は阿佐ヶ谷に集まった。



最近こういう写真を見ると、唾液が出るようになってきた。

阿佐ヶ谷は JR 中央線の三鷹～新宿間のほぼ中間に位置し、都会といえば都会といった雰囲気を出しているところだ（失礼）。少なくとも、調布よりは都会だろうか（再び失礼）。ここに来た理由はただひとつ。釣りをするためである。

今回この企画に参加してくれたのは、X680x0 同好会の同期 8 人。筆者含めて計 9 人で 釣り堀へ行った。

## 駅から近いぞ！

一体どこで釣りをするのかと思うだろうが、実は、古くからあるという釣り堀が阿佐ヶ谷にはあるのだ。それが、今回お邪魔する釣り堀、「寿々木園」さんだ。

駅の南口から、歩くこと 2 分。いや、もっと短かったかも知れない。駅前の広場を抜け、コンビニの前の道を右折し、そこから真っ直ぐ歩いて行くとさっそく看板を発見出来る。近い。すごく近い。ここで釣りをするのだ。テンション上がってきた。



目の前には「釣り堀」と書かれた看板。ここに間違いはない！

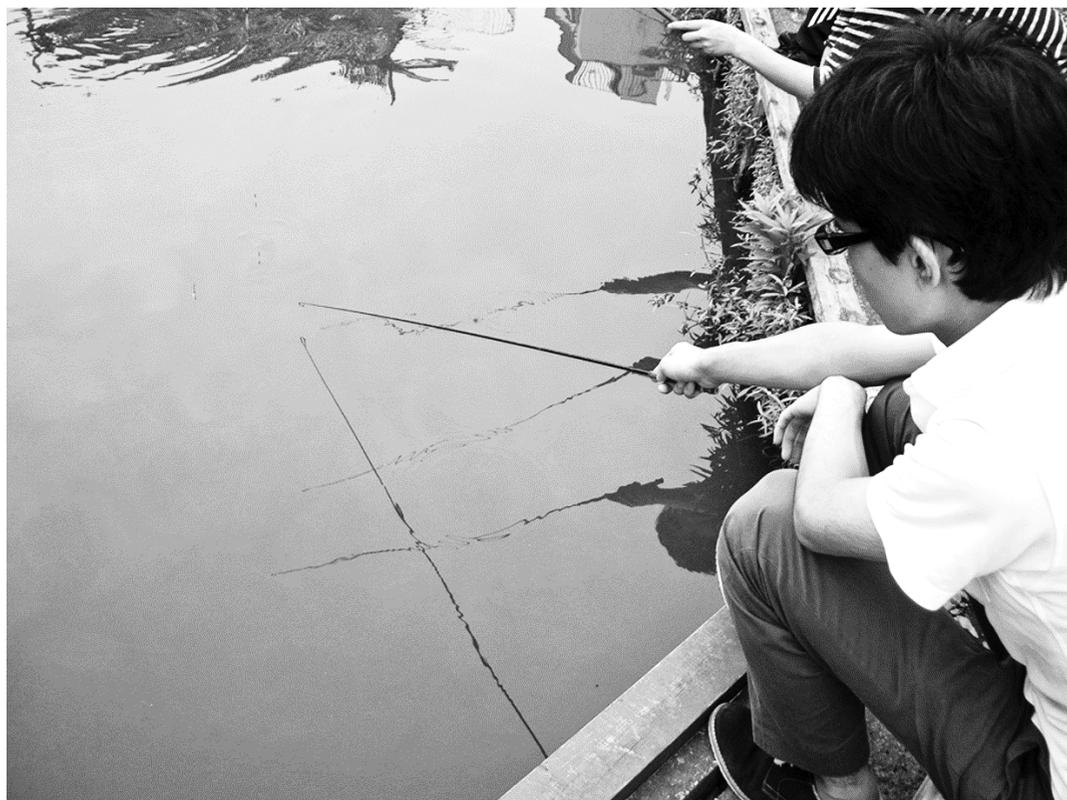
釣りに飢えた私達は、何とも言えない高揚感を抱きながら、入っていった。

## 金魚を釣るぞ

寿々木園さんは、周りをビルに囲まれた、趣のある釣り堀である。市ヶ谷の、電車から見える釣り堀とはまた違った良さを感じる。釣り池は3つあり、入り口側の1つが金魚の、奥2つが鯉の釣り堀である。今回私達は金魚に挑戦することにした。貸竿、エサ付きで最初の1時間550円で釣りが楽しめる。とてもリーズナブルだ。

竿はいわゆるウキ釣り用の1mくらいの竿で、棒ウキがついている。店先に出ていた釣り竿の数が、明らかに参加人数と比べて少なかったので一瞬焦ったが、奥からいっぱい出して頂いたので、全員一緒に釣りをすることが出来た。エサは一般的な練り餌だ。缶に入れた状態でもらえるのが嬉しい。片手で練り餌を持ち続けて臭くなるのを防ぐことが出来るからだ。

お金を払って、竿を借りてエサを貰い、空いている好きな場所に行き糸を垂らす。さあ、釣りの始まりだ。



エサをつけて池に投げたら、アタリが来るまでジッと待つ

果たして、どれだけ釣れるのだろうか。期待をともに、みな眼前のウキに集中した。

## 手段は選ばない

ウキが沈んだら、合わせるだけ。そうすれば、金魚が釣れる。単純明快なルールのはずなのだが、何故か誰も金魚を釣れなかった。どうやら、まだ体の黒い小さな金魚が群がって突くため、すぐにエサが取られてしまうことが原因のようだった。

そこで、筆者を含めた何人かは、何とんでも金魚を釣るべく、違う釣り方を始めた。それは、最初はやらないようにと、心のうちに決めていたのだが、なりふり構ってはいられなかった。サイトフィッシングともいう、「卑怯釣り」を始めた……。



よく分かる卑怯釣りの図

冬とは違い、水温が温かいので、水面付近に目視出来る金魚がいた。そこで、その金魚の目の前に釣り針を落とし、食べたのを確認したら引き上げる。これぞ卑怯釣り。実際にこういう釣り方があるので卑怯でも何でもないが、負けた感じが拭えない。それでも、この釣り方のおかげで、全員ボウズという悲しい結末を回避出来た。

## 努力はやがて報われる

---

筆者を筆頭に、周りは次々と卑怯釣りで金魚を釣り上げていくなか、己の信念を貫き正攻法を続ける侍が1人いた。誇り高き竜の名にふさわしい、飛竜氏である。目先の金魚にとらわれた筆者とは対照的に、ただただ、自らのウキが動くのを待っていた。真剣勝負にふさわしい、大物がかかるのを信じて。

その時は突然訪れた。男と男（推定）の意地のぶつかり合いだ。勝負は一瞬だった。勝ったのは、飛竜氏だった。



勝負には負けたが、全力を尽くした戦いに満足した顔をしている

これは、今回の釣りで最も大きな金魚であった。

## 祭りは終わる

---

それぞれの釣り方で金魚を釣り、それぞれ楽しんだ時間も終わりを迎えた。太陽も沈んでしまいそうな時刻だった。当初 1 時間の予定であったが、結局、30 分延長した。

今回、大体の参加者が釣果を上げることが出来た。筆者は 3 匹釣ることが出来たが、最も多くの釣果を上げたのは魯鈍ジョンさんだった。その数なんと 7 匹。彼以外の平均が 3、4 匹の中、ぶっちぎりの結果だった。

短い時間ではあったが、釣りの楽しさを感じることが出来た。きっと、他の人も同じだろう。楽しんでくれたのであれば、企画者として、これ以上に嬉しいことはない。



今回の釣果（一部）

## 謝辞

---

この企画に参加してくれた友人達、素晴らしい体験をさせてくださった寿々木園さんに、深く感謝致します。また、この記事最後まで読んでくださった読者の皆様にも感謝申し上げます。本当にありがとうございました。

## 参加者

---

いーはちえる えむひよ 臙豆腐 きよーか 凝塊 K f 高橋りなつ 飛竜  
魯鈍ジョン（敬称略）

# 実践！ ボゴソート

ゆーてー

協力者

酉腋恭介、てんぷれ、ナナタブ、  
紫電マスター、けんぼー

## 1. 目的

ソートアルゴリズムの一種であるボゴソートを手動で行うことでコンピューターの偉大さ、ありがたさを実感しよう。

## 2. ソートアルゴリズムとは

そもそもソートアルゴリズム is 何、ということでもまずはその説明から。ソートアルゴリズムとはバラバラに並んでいるデータを昇順または降順に並べ替えるときの手順のことです。例えば隣り合う二つの要素の大小関係を比べて入れ替えたり、データの中の適当な値を取ってきてそれより大きいものと小さいものを入れ替えたりする、などなど様々な手順があります。うーん、なんだかよくわからないし手間がかかりそうで大変そうですね。もっとわかりやすいソートアルゴリズムは無いのか、ということが出てくるのがボゴソートです。

## 3. ボゴソート

ボゴソートのアルゴリズムは単純です。例えばトランプの 1 スートをソートする場合（写真協力：酉腋恭介）、



図 1. 13 枚のトランプ

1. トランプ 13 枚を放り投げてバラバラにする。



図 2. 宙に舞うトランプ

2. 無作為に集める。



図 3. 地に堕ちたトランプ

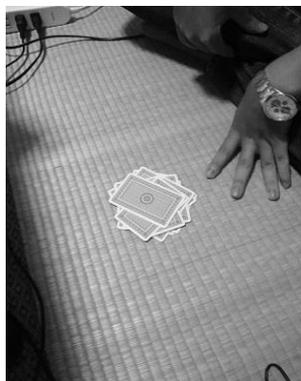


図 4. 集まったトランプ

3. ソートされているかチェックする。ソートされていないならば 1 に戻る。



図5. ダメだったトランプ



図6. 苦悶の表情を浮かべる酉腋くん



図7. 成功したトランプ



図8. 完全勝利した酉腋くん

たったこれだけです。かんたん！

ちなみに、このソートの平均計算量は $O(n \times n!)$ 、つまりこのアルゴリズムで  $n$  枚のトランプを並べ替えるにはだいたい  $n \times n!$  回かかるみたいですね。  $n=2$  なら 4、  $n=3$  なら 18 回、  $n=4$  だと 96 回、トランプ 1 スートを揃えるなら 80951270400 回も放り投げ続けなければならないのです。しかし所詮は確率の世界、うまくいけば 1 回で揃うこともあります。まさにロマン。最悪死ぬ。

## 4. 用意したもの

実践するにあたって以下のものを用意しました。

- ・トランプ
- ・人材(西脇恭介、てんぶれ、ナナタブ、紫電マスター、けんぼー)



図9. トランプはもちろん新品のものを用意しました。

## 5. 実践

このボゴソートを、トランプ 2 枚から始めて何枚までできるか、気力がなくなるまでやってみました。以下に、 $n$  枚のときにソートされるまでの試行回数と、当時の思い出を書いていきます。その日の天気は雨でした。

- ・  $n=2$  のとき

平均 4 回ですが、**1 回**でソート完了。幸先がいいですね！

- ・  $n=3$  のとき

平均は 18 回ですが **2 回**で揃いました。ここもかなり好調でした。

- ・  $n=4$  のとき

**6 回**。めっちゃすんなりいきました！ 平均 96 回とはなんだったのか。

- ・  $n=5$  のとき

**平均 600 回です!!! 34 回**で揃いました!!! 運が良すぎる……。

ちなみにここでやっと放り投げる行為に無駄が多いことに気づき、トランプをシャッフルして確認する、という方式にシフトしました。

- ・  $n=6$  のとき

**平均 4320 回**……なんだろう増えてきてる確実に、着実に。覚悟していこ

う、1000 回でも 2000 回でもやってやろうじゃん……………結果 32 回。

**減った。**もしかしてボゴソートってたいしたことないんじゃないか？ ボゴソートって実はいいやつなんじゃないか？ ボゴソートへの信頼度が高まる中、我々は次のステージへと進んでいきました。

・ n=7 のとき

実は n=6 までは 2 人で実践していたのですが、ここで増援が入り 4 人に増えました。クアッドコア万歳、これでこのステージもすぐに超えられる。今までの感じから言っても 100 回くらいで終わるんじゃないか？ ボゴソート（笑）。そんな感じで意気揚々と臨んでいきました。

そして 100 回、**終わらない**。まあ今までがちょっと調子良すぎたというもあるし多少回数が多くても仕方ないですね。

200 回が過ぎた。当然のごとく終わらない。つらい。メンバーが入れ替わったりして 300 回を超えた。そろそろ終わってください。

400 回、まだ揃わない。1234576 や 2134567 などの惜しい並び(武寺か何か?)もたまたま現れ、メンバーも「ああ、惜しかった」などと言っていたが、そもそも調整もできないし、毎回違う並びが現れる一期一会のこのソートに「惜しい」などという概念は存在しません。

500 回を超え、565 回の試行を終えたところでやる気と時間が無くなったのでその場は解散となりました。

## 6. 結果

この記事の締め切りに追われながら筆者は一人でボゴソートを行っていましたが結局 n=7 のソートは終わりませんでした。**終わりませんでした。1000 回やりました。**たぶん 2000 回やってもだめだと思います。



# 数独、やろう

’ 10 アロエ

## ◆ 数独とは

数独とは、 $9 \times 9$ マスの枠に1～9の数字を埋めるペンシルパズルのことである。広くはナンバープレースの呼称で親しまれている。数字を埋める条件として、同じ列・同じ行・同じ正方形には1～9の数字が重複してはならない。1列は縦に9マス（以下「列」と記す）、1行は横に9マス（以下「行」と記す）、1つの正方形は $3 \times 3 = 9$ マス（以下「正方形」と記す）である。マス内にはあらかじめ数字がいくつか埋められてあり、その数字を頼りにして残りの数字を推理する。自分が解こうと思った問題を撃破できたときは、相応の達成感を味わえる。以下に数独の問題例を載せる。

	8	6				7	9	
3			6		4			8
5		7		1		3		6
	1		3	6	8		5	
		5	7		2	4		
	7		4	5	1		8	
7		2		4		8		5
4			9		6			3
	3	1				9	4	

1	
2	
3	
4	
5	
6	
7	
8	
9	

ちゃんと解ける問題である。気が向いたら解いてみてほしい。

数独は誤りがあれば列・行・正方形のどこかで必ず数字の重複が起こる。別解がない限り、完成すれば正解していることになる。クロスワードパズルのような言語力を問うパズルでは、このような正当性を判断することは難しいだろう。

次のページから、数独の解き方を解説する。

◆ 解き方（基本編）

・ すぐに埋められるマスを探す

あらかじめ埋められた数字によって、すぐに埋められる箇所を埋めるところから始めよう。ここで埋めた数字のおかげで、他の数字を埋めることも可能になる。数字を見つける際は、既に埋めてある数字が多い方が残りの候補も埋めやすい。その反面、どの数字をまだ調べていないのか探す手間が残る。筆者は忘れやすい人間なので、数字の1から順番に埋められる箇所を探すことにしている。

紙面の都合上、マス目の局所的な部分しか載せない。そのため、行方向の例が多くなってしまう。そのことを念頭に置いて、列方向にも同様の手法を適用してほしい。

5	4	3	2	a	6	7	8	9

1 行には 1～9 の数字が 1 つずつ入る。⇒ a には 1 が入る。

			2	3	4			
			5	a	6			
			7	8	9			

1 つの正方形には 1～9 の数字が 1 つずつ入る。⇒ a には 1 が入る。

1								
			2	3	4			
			5	6	a			

中央の正方形は空いている 4 マス中 3 マスに 1 を入れられない。  
⇒ a には 1 が入る。

1								
							1	
			2	a	3			

中央の正方形は空いている 7 マス中 6 マスに 1 を入れられない。  
⇒ a には 1 が入る。

### ・ 9つ書き終えた数字をチェックする

すべての列・行・正方形に同じ数字が1つずつ入っていることを確認できたら、欄外にその数字を見つけたことを記しておこう。これは地味で億劫な作業である。しかし、自分の忘れっぽさを補って時間を節約するには有効だ。この記事に載せた問題には、チェックしやすいようにチェック欄を設けた。

1	2	3	4	5	6	7	8	9

解いたかな？ 備忘録に チェック欄。

### ◆ 解き方（テクニック編）

#### ・ 小さい数字を書く

1つのマスの中に候補となる数字を書く。ここで書き入れる小さい数字は、1マスを  $3 \times 3 = 9$  分割したときの1区分に収める。以下の例を見てもらえばわかるだろう。

3	1 <sub>4</sub> <sup>2</sup>	8	6	5	7	4 <sub>4</sub> <sup>2</sup>	9
7	8	6	9	4 <sub>2</sub> <sup>3</sup>	1	5 <sub>2</sub> <sup>3</sup>	
9	4 <sub>4</sub> <sup>2</sup>	5 <sub>1</sub> <sup>2</sup> <sub>3</sub>	7 <sub>1</sub> <sup>2</sup> <sub>3</sub>	7	6	8	4 <sub>4</sub> <sup>2</sup> <sub>3</sub>

1マスを  $3 \times 3 = 9$  分割して、決まった位置に小さい数字を入れる。

8	6	5
9	4	<sub>2</sub> <sup>3</sup>
<sub>1</sub> <sup>2</sup> <sub>3</sub>	7	<sub>1</sub> <sup>2</sup> <sub>3</sub>

8	6	5						
9	4	<table border="1"><tr><td><sub>2</sub><sup>3</sup></td></tr><tr><td></td></tr><tr><td></td></tr></table>	<sub>2</sub> <sup>3</sup>					
<sub>2</sub> <sup>3</sup>								
<table border="1"><tr><td><sub>1</sub><sup>2</sup><sub>3</sub></td></tr><tr><td></td></tr><tr><td></td></tr></table>	<sub>1</sub> <sup>2</sup> <sub>3</sub>			7	<table border="1"><tr><td><sub>1</sub><sup>2</sup><sub>3</sub></td></tr><tr><td></td></tr><tr><td></td></tr></table>	<sub>1</sub> <sup>2</sup> <sub>3</sub>		
<sub>1</sub> <sup>2</sup> <sub>3</sub>								
<sub>1</sub> <sup>2</sup> <sub>3</sub>								

正方形の拡大図。左側は実用するときの図。右側は9分割を見やすくした図。

1マスを9分割して決まった位置に小さい数字を入れることには、理由がある。推理が進むに連れて、小さい数字を書いたり消したりすることになる。この操作によって曖昧に書き残された小さい数字が現れる。この数字から推理をしようとする場合、何の数字が書いてあったのかわからないと時間の無駄になる。しかし、位置を決めておけば、消えかけた数字の位置から数字を復元できるのだ。

小さい数字を書くと言っても、書き入れる基準はまちまちである。次のページでは一定のルールに基づいた2つのケースを取り上げる。

1. 候補となるすべての数字を書く。

空いているすべてのマスに候補となる数字を書き入れる手法。メリットは、唯一に定まる数字を見つけることが容易になること。デメリットは、作業量が多くなること。難易度が高くなると、この唯一に定まる数字が1箇所にしかなく、それを見つけるまで場が硬直状態になることもある。

3	1	$4^2$	8	6	5	7	$4^2$	9
7	8	6	9	4	$2^3$	1	5	$2^3$
9	$4^2$	5	$1^2^3$	7	$1^2^3$	6	8	$4^2^3$

空いているすべてのマスに候補となる小さい数字を書き入れる。

2. 候補となる数字が2つだけ書けるときに書く。

列・行・正方形において1つの数字を候補として持つマスが2つだけあるとき、その2つのマスに小さい数字を書き入れる手法。メリットは、片方の小さい数字が消えると、もう一方には必ず数字を埋められること。デメリットは、列方向に見て書いた数字を、行方向に見て決定できると誤認すること。逆もまた然り。

3	1	$4^2$	8	6	5	7	$4^2$	9
7	8	6	9	4		1	5	$3$
9	$4^2$	5	$1$	7	$1$	6	8	$4^3$

列・行・正方形の中で2つだけ候補があるところに  
小さい数字を書き入れる。

ちなみに、筆者は基本的に2の基準を進めていき、どうしても行き詰まった際に1の基準を採用する。その理由は、マス目の変動が少ない序盤を乗り越えられることと、後への保険をかけられることにある。難しい問題ほど、マス目の変動は滞りやすい。そんなとき、小さい数字が変動するだけでも飽きずに続けられる。また、後ですべての候補を書くことになっても、手間が軽減される。

小さい数字を埋めるときも1～9までを順番に1周見ていくとやりやすい。3周もすれば見落とした箇所はほとんどなくなるため、3周を終えてもマス目に変化が見られないようであれば、すべてのマスに小さい数字を書き入れる覚悟が必要だろう。

次のページでは、各マスに書き入れた小さい数字の活用法を紹介する。

・ 小さい数字から推理する

ある列／ある行／ある正方形の中で同じ小さい数字を書き入れられるマスが1箇所しかない場合、そのマスにその数字を埋められる。このとき、確定した数字と同じ列・同じ行・同じ正方形の中にある小さい数字はすべて消すこと。

			6			
			2			
9		5	1	3		7
			4			
			8			

小さい数字が1つだけの場合、その数字を埋めてよい。

また、数字は埋まらなくとも候補の位置を限定できる場合がある。

		1	2	1		
					1	
1						

数字を埋められなくても、候補（小さい数字）の位置は決められる。

1						
		2	3	4	1	1
		1	1	1		

中央の正方形は3行目にしか候補を入れられない。

⇒ 右の正方形は2行目にしか候補を入れられない。

小さい数字を書くことが真価を発揮するのは、ある列・ある行の中で同じ小さい数字の組み合わせを持つマスが、そのマス中の数字の個数分だけ存在するときである。言葉だけではわかりにくいであろう。百聞は一見に如かず。次のページで紹介する。

		1 2				1 2		

ある行の中に、小さい数字の1と2を持つマスが2つだけある  
⇒ 同じ行の他のマスに1と2は入らない。

			1 2					
				1 2				

ある正方形の中に、小さい数字の1と2を持つマスが2つだけある  
⇒ 同じ正方形の他のマスに1と2は入らない。

			1					
			2					
	1						2	
2	2	2	3	4	1	1	1	
			1 2		1 2			

より実戦に近い例。下段中央の正方形で1と2が候補となるのは3行目。  
⇒ 左右の正方形の2行目の候補が限定される。

筆者がお勧めしたいのは、ある数字が確定したことによって他の列・他の行・他の正方形でも同じ数字が確定しないかを確認することである。この細かい確認作業が全体としての作業時間を軽減してくれる。

・ 注意を怠らない

小さい数字を見て入る数字が決定しそうだというマスを見つけたときも、すぐ飛びついて数字を埋めるのはよくない。落ち着いて周りのマスを見てみよう。既に同じ数

字が埋められていた場合、過去にその小さい数字を消し忘れた可能性がある。また、集中が途切れて普段なら発生しないはずの推理ミスをしている場合もある。よほど明確なときを除いて、2, 3度推理を見直そう。

数独は、自らの注意力との勝負である。序盤から中盤にかけては常に推理を続けることになる。推理から導いた数字で埋まったマスが一定数を超えると、残りは作業のように埋められる。

他にも高度なテクニックはある。しかし、基本的には上記のテクニックだけで太刀打ちできる。筆者はここに書いたことしか知らない。なお、上記のテクニックだけだと時間が掛かる。そのため、もっと早く解きたいという方は、インターネットで検索して各自で調べてほしい。

## ◆ こんなときは

### ・ 数字に重複が出てしまった

まずは落ち着いて重複が出た数字の周りを見てみよう。他にも重複を発見できるのであれば、そこで誤った可能性が高い。ここで誤りの箇所を発見できない場合、または終盤まで進みすぎて間違いの箇所を容易に修正できない場合、数字を入れる際に誤った推理を行った可能性がある。諦めて最初からやり直そう。最初からやり直すときに大切なのは、ミスをしたときと同じ推理だと思ったところで、改めてその考え方を見直すことである。勝手な思い込みによる推理のミスは極力排除しよう。

### ・ 行き詰まってしまった

落ち着いて全体を見渡してみよう。どこかに手がかりはないだろうか。小さい数字が少しだけ書いてあるマスがないか、探してみよう。ある列・ある行・ある正方形で、既に多くのマスが埋まっているのであれば、その周りのマスを調べてみよう。

### ・ 頭が回らなくなってきた

たかがパズルと侮るなかれ。数独は集中力を要するパズルなので、長時間取り組めば疲れを感じる。適度に休憩を挟んで、本領を発揮できる状態で臨もう。

## ◆ 腕試し

次のページから筆者が自作した数独の問題を載せておく。小手調べに取り組んでみてはいかがだろうか。会誌を汚したくない方は、ページをコピーして取り組むことをお勧めする。なお、答えはこの記事の最後に載せる。後で遊んでみたい方は、誤って最後のページの数字列を覚えてしまわないように、お気をつけて。

・ 初級

2				6				7
	7		8	1	4		5	
		5	2		7	6		
	6	1		2		8	7	
5	3		7	9	1		6	2
	2	7		8		5	3	
		4	6		2	3		
	5		1	4	9		8	
7				5				4

1	
2	
3	
4	
5	
6	
7	
8	
9	

・ 中級

		7				4		9
	5		7				6	
4		9		3	8			7
	4		1		6	3		
		6		7		8		
		1	3		2		5	
2			9	1		7		6
	9				3		1	
1		4				9		

1	
2	
3	
4	
5	
6	
7	
8	
9	

・ 上級

	9	6				3	1	
				3				
		4	7		5	6		
		2	5		8	1		
	4						5	
8				4				7
4								1
				9				
5	3		8		6		2	4

1	
2	
3	
4	
5	
6	
7	
8	
9	

◆ 数独の変わり種

同じルールのパズルばかりに取り組んでいると、飽きを感じることもある。そうならないためには、時々違うルールの数独に取り組むと良い箸休めになるだろう。なお、筆者に例題を作る余力がなかったため、以下は文面のみの紹介となる。

・ 複数枚の数独が重なったもの

通常の  $9 \times 9$  の数独の一部のマスが他の  $9 \times 9$  の数独の一部と重なっている。重なっているマスの条件が自ずと厳しくなるので、重なる部分のマスが埋めやすい。

・ 列・行・正方形のマスの数が多いもの

$9 \times 9 = 81$  マスではなく、 $4 \times 4 = 16$  マス、 $16 \times 16 = 256$  マス、 $25 \times 25 = 625$  マスの数独もある。

◆ おわりに

もしこの記事を読んだ方が数独を面白そうだと感じたならば、書店で軽く探してほしい。世の中にはこれほど数独を扱う雑誌があったのかと目を見張ることであろう。個人的には、レベルが偏っておらず異色の数独も含む雑誌を購入すると末永く楽しめる。懸賞もあると、解いた後のご褒美がある気がして満足感が増えるかもしれない。

◆ 解答集

ここには本編で出題した問題とトップに載せた問題の解答を載せる。

2	8	3	9	6	5	1	4	7
6	7	9	8	1	4	2	5	3
1	4	5	2	3	7	6	9	8
4	6	1	5	2	3	8	7	9
5	3	8	7	9	1	4	6	2
9	2	7	4	8	6	5	3	1
8	9	4	6	7	2	3	1	5
3	5	2	1	4	9	7	8	6
7	1	6	3	5	8	9	2	4

初級の解答

6	3	7	2	5	1	4	8	9
8	5	2	7	4	9	1	6	3
4	1	9	6	3	8	5	2	7
5	4	8	1	9	6	3	7	2
3	2	6	5	7	4	8	9	1
9	7	1	3	8	2	6	5	4
2	8	3	9	1	5	7	4	6
7	9	5	4	6	3	2	1	8
1	6	4	8	2	7	9	3	5

中級の解答

7	9	6	2	8	4	3	1	5
1	5	8	6	3	9	4	7	2
3	2	4	7	1	5	6	9	8
9	7	2	5	6	8	1	4	3
6	4	3	1	2	7	8	5	9
8	1	5	9	4	3	2	6	7
4	6	9	3	5	2	7	8	1
2	8	7	4	9	1	5	3	6
5	3	1	8	7	6	9	2	4

上級の解答

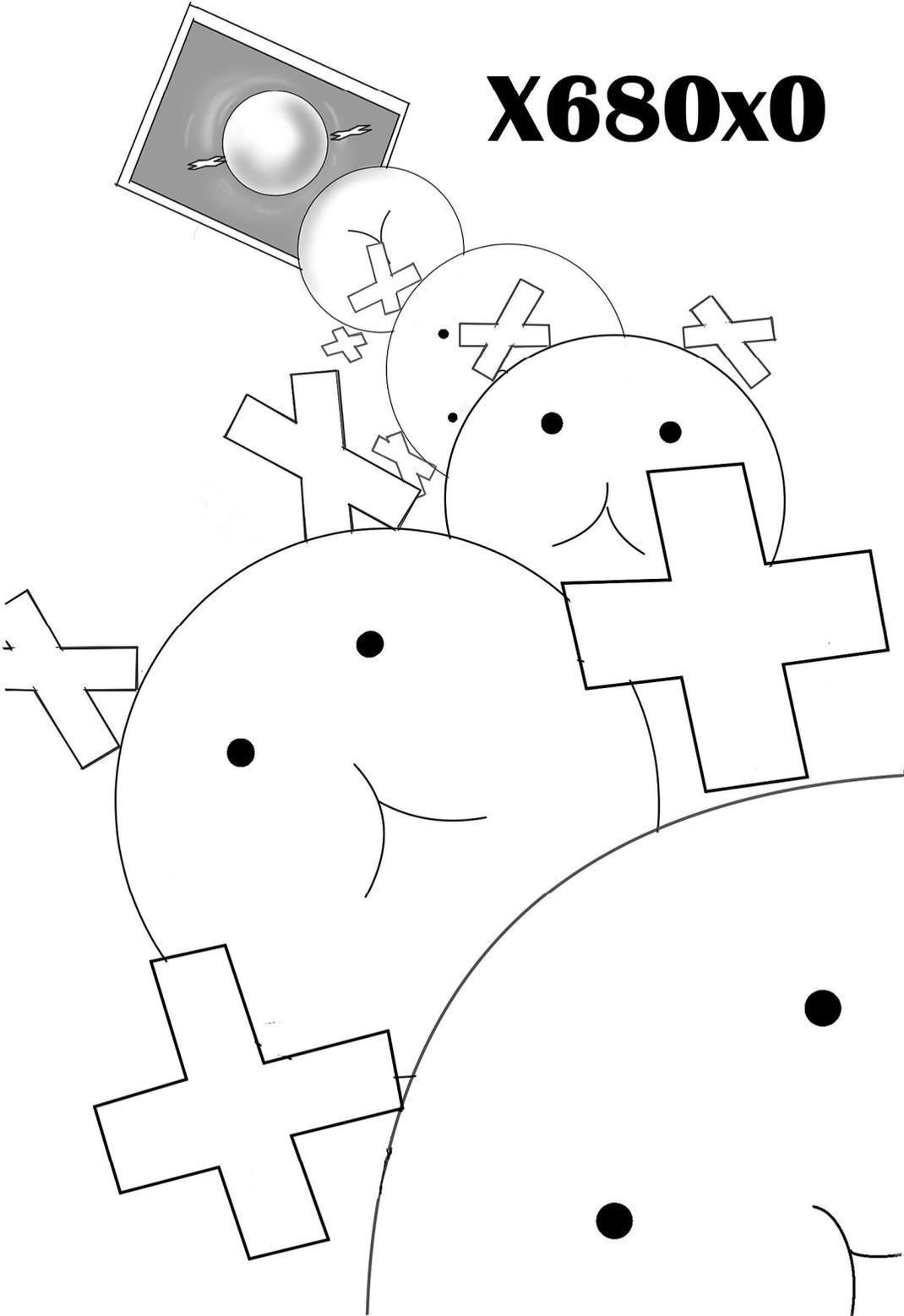
1	8	6	2	3	5	7	9	4
3	2	9	6	7	4	5	1	8
5	4	7	8	1	9	3	2	6
9	1	4	3	6	8	2	5	7
8	6	5	7	9	2	4	3	1
2	7	3	4	5	1	6	8	9
7	9	2	1	4	3	8	6	5
4	5	8	9	2	6	1	7	3
6	3	1	5	8	7	9	4	2

トップの解答

◆ 余談

スペースが余ったので執筆を終えての感想を記すことにする。この記事を書くにあたって、問題の作成に一番時間が掛かった。数独の解答をランダムに生成するプログラムを作成するだけで半月を費やしてしまった。そこから問題を作っては解き作っては解き、解が1つに定まることを確認する途方も無い作業があった。ペンシルパズルの制作者の苦勞を知り、頭が上がらない思いでいっぱいである。筆者はこれからも一人数独ファンとして楽しませて頂く立場に徹したいと思う。問題作成は懲り懲りだ。

**X680x0**



## あとがき

『68 通信 vol.19』、楽しんでいただけたでしょうか。

今年も多くの部員たちの協力により、素晴らしい記事がたくさん集まりました。結果として 200 ページを超す大ボリュームとなり、一大学のゲーム制作サークルが頒布する活動報告誌のなかでは日本最大レベル(!)のものとなっています。

私達が製作しているゲームのメイキング記事、部員たちの持つ様々な知識や技術の紹介、はたまた一発ネタなど、なかなか読み応えのある仕上がりになっているのではないかと自負しております。

この会誌を手に取りここまで読んでくださった皆様、そして会誌の制作にご協力いただいた多くの方々、本当にありがとうございました。

編集長 奥野舜

### 『68 通信 vol.19』

発行	2013 年 11 月 22 日
発行元	電気通信大学 X680x0 同好会
表紙	及川雄介
裏表紙	梅津和朗
目次	香住昂明
扉絵	坂井大輝 小林隼人 鈴木航平 山口拓哉
挿絵	橋渡翼 高木一成 寺坂元 渡邊秀哉
編集	奥野舜 早瀬圭佑

本書における表現や発言、その他全ての事柄、またはその引用や転載によって如何なる被害を受けようとも、当方は一切の責任を負いかねます。ご了承ください。

本書の転載に関しましては、以下のアドレスのホームページ上の掲示板にてその旨をお伝え頂き、許可を得てくださるようお願い申し上げます。

<http://www.x68uec.org/>



**X680x0同好会** <http://www.x68uec.org>