

いつしか部誌を書かなければならない学年と季節がやってきて、過ぎて行きつつある。バクリである。とはいえ、自分は他の先輩方に比べるとPC、及びプログラミングの知識については無に等しく、パソコン部にいて果たしていいのだろうか疑問に思ったりもする。しかしまあ、書かなければいけないのは事実。部員の少なさもあって、このままでは今年の10月頃に部長になってしまうのだ。このままでは最悪。

プログラミングの知識についてはこれから半年かけて伸ばしていくとして、今眼前に立ちはだかる強敵、むしろ無理敵ぐらいの威圧感をかもし出している、「部誌」という名の敵。これを残る4時間で倒さなければならないのだ<sup>1)</sup>。つまり何が言いたいかというと『字数を稼ぐ必要がある』わけで。そうはいっても内容がなければ始まらないのでさっさと始めることにするけれども、これから説明するのは自分の唯一の、あえて挙げると言われたら特技と言ってもいい、「ゲーム」。具体的に言うと「弾幕シューティングにおける弾幕のアルゴリズム」、あとアルゴリズムそのもの。

何を言っているのかわからない人は次を読んでもらえればわかると思う。

### アルゴリズムの話

例を挙げれば、素因数分解。

素因数分解のやり方といえば、分解したい数の二乗根の数値を超えない素数まででひたすらその数を割っていくことだが、これはどの自然数を代入してもやり方そのものは同じだ。つまり、やり方を予め決めておいて、そうすれば数値によって結果を変えられる、ある意味方程式のようなものだと思えばわかりやすいだろう。 $X + Y = 1$ なんてものも方程式だから、もちろん代入する変数はひとつではない。弾幕なら、弾の速さ、角度、大きさ、などなど。

まあ詳しくは、次から見ていくことにするが、その前にひとつ謝っておかなければならないことが。

タイトルはどうみても適当です、本当にありがとうございました。

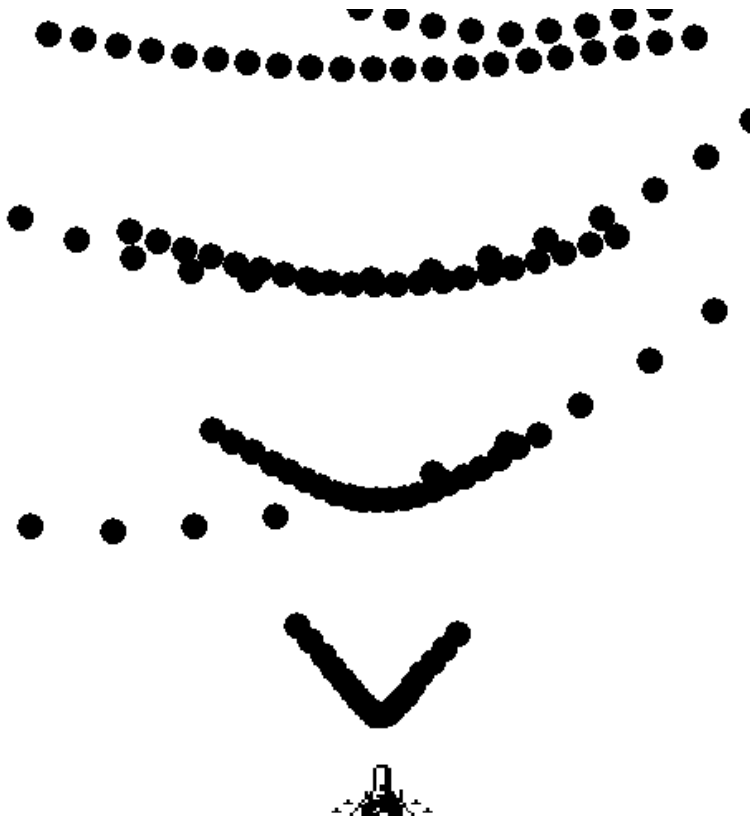
### 弾の種類の話

意味不明で申し訳ない。が、まずはこれを見ていただこう。俗に言う(のかは知らないが)「弾幕」の一種・自機狙いである。先輩による一昨年の文化祭作品から。

---

<sup>1)</sup>4 時間では倒せませんでした

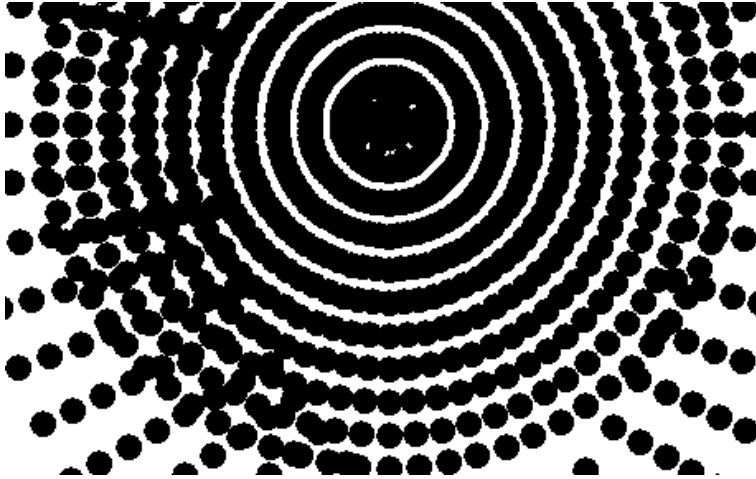
図 3.1 自機狙い弾



自機に向かって色々と飛んできている。その名の通り、自機狙い。速度は同じだが距離が違うので、列になって飛んできている。

自機を狙うのだから、自機の座標の情報が必要だ。弾の発射地点と自機との座標を見比べて、その差をとれば方向ベクトルがわかる。そうなればあとは適当に速度を合わせるように関数に放り込むだけだ。言うほど簡単ではないが、言うほど難しくもない。

図 3.2 72way 弾



中央の敵がやたらめったら弾を出している。全方向 72way 弾というところか。  
全方向に弾を一定の角度を持って出す場合、必要なのは弾の速度と方向だ。「大きさ」と「方向」を持つ数値...というわけで、再びベクトルの出番である。

というのは嘘で、

別にそれでやってもいいんだが芸がないというか他の方法で考えた方がわかりやすいのでここは円らしく角度 (ラジアン) を使うことにしよう。

この場合 72 方向に弾が出ている。一周は  $2\pi$  ラジアンなので、 $\pi/36$  ラジアンごとに弾を出せば万事解決。右方向を 0 度、角度を表す変数を  $r$  (最初は 0) とする。弾を撃つ関数の引数として  $0, \pi/36, 2\pi/36 \dots$  としていくわけだが、この場合すべての弾を同じ速度にしようと思うと、速度 1 ごとに弾が進む  $x$  座標を  $\cos r$ 、 $y$  座標を  $\sin r$  とすればいい。

もう少しだけわかりやすく

$$\sin^2 r + \cos^2 r = 1$$

つまり、どの角度でも中心から同じ速度で離れていくことになる。これに速度をかける。ちなみに角度によって速度を変えて何回か連続で全方向弾を出すと、時間差で飛んでくる弾幕も作れる。途中で速度を変えるようにするのも面白い。<sup>2)</sup> 何フレームかごとに 1 速度を足していくような<sup>3)</sup> ことにすれば、加速弾なんかも作れてお得。

ちなみに全方向弾を作る際、必ずしも 360 の約数でないといけないというわけではない。後で説明もするが、自機の動きを制限するいわゆるワインダーなんかは全方向弾でも自機が入る部分だけ広めに取ってあったりする。その場合は、自機の方向 +5 度ぐらいを 0 として適当にいいところまで足していくのだが。

<sup>2)</sup> フレームで速度を変えればいい

ええ 5 で割ると 2 余る数だとかの条件分岐を使う

<sup>3)</sup> つまり、弾が発射されてからのフレーム数が、例

え

## バカとハサミの使い方の話

アルゴリズム、とは前項で紹介した自機狙いやら全方位弾やらの1つ1つなのだが、しかし弾幕というのはこれらを組み合わせてなるものだ。その組み合わせ方で難易度も変わってくる<sup>4)</sup>。

アルゴリズムの便利なところは、一度それを作ってさえおけば、数値を変えることでいくらでも弾幕なら難易度を变化させることができるから、というのが一番大きいだろう。それに、自機狙いを作る レーザーを作る 全方位弾を作る、ときて、この3つの内いくらでも組み合わせられる。取捨選択が可能なのだ。

例)

1. 自機基準の全方向 72way 弾
2. 弾の発射源を動かす

自機基準の 72way については、`atan2()` 関数を使えばいい。座標の差を入力すると勝手に角度を計算してくれる優れもの。あとはさっきのラジアン<sup>5)</sup>の要領で、発射源を動かすのは、フレームごとに勝手に動くようにしていればいい。

簡単に作れる弾同士の組み合わせでも、面白い弾幕が作れたりする。他のゲームの弾幕を作ることに楽しみを感じる人がいるのは、そのせいかもしれない。

## オリジナリティの話

動きがどうみても単純なものではない弾だってもちろん存在する、というよりいくらでも作れる。加速度をランダムにして  $n$  フレーム後に分裂、とかやるだけでも、単にうざったいだけだろうが、オリジナルの弾幕だ。

ところで、分裂について説明してなかったことを今更思い出した。といっても単純なもので、弾を消すと同時に弾を出せばいいのだが。その際弾の座標が発射源の座標になる。

数学の知識が必要なことは、確か。数学の勉強をしていれば面白い軌道の関数だってお目にかかるかもしれないし、美しい幾何学模様、というものも知れる。そんな「作ってみたい模様」を関数で表記し弾の軌道にして際限無く弾を出し続ければ、とても綺麗な幾何学模様が不思議と弾幕によって浮かんでくるわけだ。

関数同士を組み合わせることによって、幾何学を完成させるアルゴリズム。やっぱり、プログラミングやらが面白いと思う瞬間でもある。

まだ体験してないけど。

## 言い訳の書き方

終わってしまった。

いうまでも無い、ネタ切れだ。そもそもなぜレーザーも作れないのにこんなものを書こうなどと思ったのか、今でもさっぱりである。

ちなみに世間には誘導レーザーなんて奇妙なものもあったりして、作れるならばその紹介で1ページは消費していたであろう。いかにもややこしそうだし。

<sup>4)</sup> たまに完全なオリジナルの軌道を持つ弾幕があったりするが

<sup>5)</sup> さっきの自機狙いもそうすればよかったんじゃないですか。いかって? 処理重いじゃないですか。

弾とレーザーの一番の違いといえば、「くぐれるかくぐれないか」というのは間違いないと思う。上で述べたワインダーなんかも、人間技ではないが抜ける余地はある。たまにない。

なぜ後書きでワインダーの説明なんかをしようとしているのかというと、単純に忘れてたから。つくづくダメ人間だ、最近はやりのNEETにはならないよう最大限の努力をするつもりではあるが、果たして。

そういえばひとつ勘違いをしていたのは、NEETというその言葉自体には本来は、「働きたくない人」という意味はないそうで。「教育を受けていない、労働をしていない、職業訓練も受けていない」というわけで。どうでもいいや。

あ、ワインダーは「自機の動きを制限するような、でも完全には制限できてないものすごく速い弾」だと思ってくれればいいです。作り方も自機±5度に打つとかでオーケー。

なぜ後書きになってこうまで文体が崩れているのか説明すると、疲れたから。あと時間がないから。

とまあ、あらかた思いつく言い訳と理由を書いてきたけれども、果たしてこれが後書きになるのかそれはわからない。ただ単に書きたいこと、書ける最大限のことを書くように努めただけだ。

最後はうまく締まったと思うが、どうだろう？

来年は、これの2倍の分量は書きたい。と思う。自信はない。

ゲームが完成していることを祈りながら、先輩方にバトンタッチ。