

「diep.io」に関する物理法則

Ponyomediatly

1. はじめに

diep.ioとはブラジルの開発者マテウス・ベルナド・パラダレス氏によって開発された大規模多人数同時参加型オンライン2Dアクションゲームである。私は元々このゲームを遊ぶ側の人間として起居していたが、ある日を境にゲーム内の処理についての解析を行う者へと変貌していった。意外にも、世界を見れば私のような者たちは沢山いる。私たちはSNSサービスを利用し、コミュニティを作り、集まって情報を収集および共有、交換、調査や検証等を行っている。

そもそも、このゲームにおける物理的情報をまとめたドキュメントは既に存在しているが、それでもこのようなものを作成することに至った経緯としては、私が独自にこのゲームを再現するプロジェクトに役立つための情報源をまとめたいと思ったからである。言い換えれば自分用のものをただ作りたかっただけである。

酔生夢死よりかは許されるが、人生の約5%をこのゲームに費やしたと考えると哀愁が漂うので、それに関することの記載はやめておく。

2. エンティティ

ここで言われるエンティティとは、サーバーの処理対象な概念が存在するものを指す。エンティティには幾つかの種類が確認されている。通信で利用されるものを列挙すると砲台(C)、衝突可能(CA)、破壊可能(DA)、マップ情報(MI)、名づけ可能(NA)、プレイヤー情報(PI)、描画可能(RA)、機体(T)、チーム情報(TI)である。これらは組み合わせて一つのエンティティにすることが可能であり、例えば、我々プレイヤーのエンティティ構成(P')は

$P' = CA + DA + NA + PI + RA + T (+TI)$
である。

2.1 砲台エンティティ

砲台エンティティ(C)は、弾エンティティを生成することが可能なエンティティである。なお、通信で利用する際のエンティティ構成(C')は

$C' = C + CA + RA (+TI)$
である。

弾エンティティを生成するタイミングは間隔、遅

延によって決定され、ステータス「reload」のポイント量 S_6 が大きいほど、発射タイミング t は小さくなる。これを考慮し、通信内容で使用されているデータを参考に関係式を考えると、

$$t = \frac{\frac{1000}{T} \cdot 0.6 \cdot r}{1.075^{S_7}}$$

のような式になると推測した。なお、定数 T はチェックレート（1秒間に何回サーバーの更新頻度を行っているかを示す値）であり、その値はこのゲームにおいては 40。 r は機体「Tank」を 1.0 とした時の基礎リロード時間である。

2.2 衝突可能エンティティ

衝突可能エンティティ(CA)は、エンティティ同士が互いに部分的に重なった際に物理的挙動に関する処理が実行されるエンティティである。なお、その一連の処理をゲーム内における「衝突」という。

衝突しているかどうかの検出方法は、クライアントが描画する図形によって異なる。具体的には、描画予定の図形が四角形の場合は「四角形」による判定アルゴリズムで行い、正多角形または円の場合は「円」による判定アルゴリズムで行う。正多角形でも円による判定で行う理由について、これはあくまでも推測に過ぎないが、正多角形の頂点数が多いほど検出しているかどうかの処理判定に時間を要してしまう。よって、円による判定で処理を単純かつ軽量にするためこのようなアルゴリズム方式になってしまっていると考えている。また、全体の当たり判定処理には四分木空間分割が用いられている。なお、四分木に関するアルゴリズムはここでは割愛する。

さて、まず衝突検出するための重要な要素である「エンティティの大きさ」について調べていく。大きさを中心からの距離 r とした時、プレイヤーエンティティについてはレベルによって変動する。これを考慮すると、エンティティのレベル l として

$$r = 50 \cdot 1.01^{(l-1)}$$

のような関係式を得られた。他のエンティティについては、値が固定なのでまとめると次のような表になる。

表1. 特定のエンティティについての r の値

エンティティ名	r
Crasher (Small)	$17.50\sqrt{2}$
Square, Triangle, Crasher (Large)	$27.50\sqrt{2}$
Pentagon	$37.50\sqrt{2}$
The Guardian of the Pentagons, Summoner, Defender	$67.50\sqrt{2}$
Alpha Pentagons, Fallen Booster, Fallen Overlord	$100\sqrt{2}$
Dominator	150
Mothership(?), Arena Closer	200

次に、衝突時のダメージ量について考える。衝突相手に付与可能な最大ダメージ量を D'_a とするとき、この値は互いのエンティティの種類の組み合わせによって異なる。それを考慮した関係式を考えると

$$D'_a = D_a \cdot D_m$$

となる。 D_a は衝突相手に対する本来のダメージ適用量、 D_m は互いのエンティティの種類の組み合わせによって変動するダメージ適用率である。 D_a の値の詳細について通信内容解析によると、プレイヤー（機体）エンティティに対しては、 S_2 をステータス「Body Damage」のポイント量とすると

$$D_a = 6 \cdot S_2 + 20$$

である。弾エンティティに対しては、 S_5 をステータス「Bullet Damage」のポイント量とすると

$$D_a = d(7 + 3 \cdot S_5)$$

である。 d は機体「Tank」を 1.0 とした時の弾の基礎ダメージ量である。その他のエンティティについての値のデフォルト値は 8.0 で、それ以外の値に関しては次のとおりである。

表2. 特定のエンティティについての D_a の値

エンティティ名	D_a
Pentagon	12
Alpha Pentagon	20
Crasher (Large)	4.0
The Guardian of the Pentagons, Summoner, Defender, Fallen Booster, Fallen Overlord	?
Dominator, Mothership	?
Arena Closer	?

D_m の値に関して、デフォルト値は 1.0 でそれ以外の値に関しても次のとおりである。

表3. エンティティの組み合わせによる D_m の値

組み合わせ	D_m
互いに機体	1.50
相手が弾	0.250
互いに弾 (?)	0.50 (?)

さて、次は実際に衝突した際にダメージ処理を考えたいのだが、コンピュータ上、ここである問題を引き起こす。2つのエンティティ A , B を用意する。衝突時に A が B を瞬時に破壊可能な場合に並列処理の関係上、以下のような問題が発生する。

並列処理

- ① A が B に対してのダメージ処理 ここで B を破壊
- ② B が A に対してのダメージ処理 ← B は破壊済みなので、ここの処理が実行不可

B は A に対して多少のダメージを与えているので、この問題を解決するための計算を考慮しなければならない。まず、衝突相手に対して付与可能なダメージ量 D_r は、 D'_a とその相手のエンティティの現在の体力 H'_c を用いると

$$D_r = \begin{cases} H'_c & H'_c - D'_a < 0 \\ D'_a & \text{otherwise} \end{cases}$$

となる。あくまでも付与可能なダメージなので、破壊済みであるにも関わらずダメージを与えるケースを無視しなければならないため、条件式になっている。そして、その衝突相手に対してのダメージ適用率は D_r と D'_a を用いて

$$D_p = 1 - \frac{D_r}{D'_a}$$

と表せる。なお、この値の変域は必ず

$$0 \leq D_p \leq 1$$

である。このダメージ適用率を応用することで、余ったダメージ量を無駄にすることなく双方にダメージを適用することが実現できる。実際にエンティティ A とエンティティ B が衝突し、互いに付与するダメージ量 D は

$$\begin{aligned} D_{AB} &= D'_{Aa} - D_{Br} \cdot D_{Ap} \\ D_{BA} &= D'_{Ba} - D_{Ar} \cdot D_{Bp} \end{aligned}$$

となる。

最後に、ノックバック（反動）について考える。

衝突相手に反映される反動量 K'_a について、付与する反動量 K_a と相手の反動適用率 K_m を用いると

$$K'_a = K_a \cdot K_m$$

となる。そして K_a の値について、デフォルト値は 8.0 で、それ以外の値に関しては次のとおりである。

表4. 特定のエンティティについての K_a の値

エンティティ名	K_a
壁	2.0
ドローン系	4.0
Pentagon, Alpha Pentagon	11
Crasher (Small), Crasher (Large)	12

となる。 K_m の値について、弾エンティティにおいては

$$K_m = k$$

である。なお、 k は機体「Tank」を 1.0 とした時の弾の基礎反動量である。その他エンティティにおいては、デフォルト値は 1.0 で、それ以外の値に関しては次のとおりである。

表5. 特定のエンティティについての K_m の値

エンティティ名	K_m
Crasher (Small)	2.000
Pentagon	0.500
Crasher (Large)	0.100
Alpha Pentagon	0.050
The Guardian of the Pentagons, Summoner, Defender, Fallen Booster, Fallen Overlord, Mothership	0.010
Dominator, Arena Closer	0

2.3 破壊可能エンティティ

破壊可能エンティティ (DA) は、体力概念があるエンティティのことを指す。最大の体力を H_m とし、現在の体力を H_c とする。ただし、変域は必ず

$$0 \leq H_c \leq H_m$$

である。サーバー側の処理では $H_c = 0$ のとき、エンティティの削除処理が行われる。なお、 H_m の値について、プレイヤーエンティティに関しては、エンティティのレベル l , S_1 をステータス「Max Health」として

$$H_m = 50 + 2l + 20 \cdot S_1$$

のような関係式を得られた。また、弾エンティティに

関しては、 S_4 をステータス「Bullet Penetration」のポイント量とし、 h を機体「Tank」を 1.0 とした時の弾の基礎体力とすると

$$H_m = (2 + 1.5h) \cdot S_4$$

のような関係式を得られた。他の種類のエンティティについては、次のとおりである。

表6. 特定のエンティティについての H_m の値

エンティティ名	H_m
Square, Crasher (Small)	10
Triangle, Crasher (Large)	30
Pentagon	10^2
Alpha Pentagon, The Guardian of the Pentagons, Summoner, Defender, Fallen Booster, Fallen Overlord	3.0×10^3
Dominator	6.0×10^3
Mothership	7.0×10^3
Arena Closer	1.0×10^4

驚くことに「Arena Closer」はダメージ適用が無効であるにも関わらず最大体力が定義されている。これらは通信内容解析により得られた確定値であるので、この情報は事実である。

さて、衝突可能エンティティと破壊可能エンティティの情報から耐久力が求められる。耐久力は高ければ高いほどそのエンティティを破壊しづらくなる性質を持つ。定義として、耐久力 E は

$$E = H_m \cdot D_a$$

のような関係で得られるものとする。 D_a は衝突相手に対する本来のダメージ適用量とする。これは体力も衝突時に相手に付与できるダメージ量も大きいほど値が大きくなる。

例えばプレイヤーエンティティにおいて、ステータス「Max Health」と「Body Damage」のポイントによる耐久力 E は

$$H_m = 50 + 2l + 20 \cdot S_1$$

$$D_a = 6 \cdot S_2 + 20$$

より

$$\begin{aligned} E &= H_m \cdot D_a \\ &= (50 + 2l + 20 \cdot S_1)(6 \cdot S_2 + 20) \\ &= 4(25 + l + 10 \cdot S_1)(3 \cdot S_2 + 10) \end{aligned}$$

となる。この式より、レベルが上がる度に耐久値が 4 ずつ上昇することが分かる。

3. 参考文献

- . <http://spade-squad.com/physics.html> 【SAPDE-SQUAD.COM】より
- . <https://github.com/ABCxFF/diepindept/blob/main/physics/README.txt> 【diepindept】より