

ゲームプログラミングの流れ

M1C 某院生

1 はじめに

こんにちは。M1 になってから突然入会して、先日の夏合宿をドタキャンしてしまった某院生です¹

最近暇が無くて全然 68 的活動をしていないんですが、高校時代はちまちまとゲームを作って遊んでいました。そのときの経験をもとに、ゲーム作成の流れを簡単にまとめてみました²。何かの役に立って頂ければ幸いです³。

2 アイディアを固める

まずは作りたいもののアイディアがなければ始まりません。日頃から何か思いついたらすぐにメモをとる癖をつけておくといいと思います。

2.1 紙に書く (描く)

もちろん Wiki や Blog、各種発想支援ソフト⁴などでも構いませんが、いつでもどこでも使えるので紙や手帳がお勧めです⁵。自分に合った方法を見つけておきましょう。

大事なのは、頭の中で描いたものを見える形にすることです。アイディアスケッチをすることで、今まで気づかなかったことが見えてきたり、さらに新しいアイディアが生まれる、といったことがよくあります⁶。スケッチしたものを他の人に見せて意見をもらえるのも大きな利点です。

2.2 想像を膨らませる

製作前にモチベーションを上げるだけ上げておかないと、途中で行き詰まったときにやる気が無くなってしまいます。逆に途中で行き詰まったら、いったん手を止めて頭の中で完成図を描いてみるといいかもしれません。

¹仕方ないじゃん、突然とある研究会で発表を依頼されちゃったんですから。別にたいして凄いものを作ってるわけでもないのに、持ち時間 90 分ですよ？ それ何で死刑宣告ですか!?

²技術的なことは部内向けミニゲーム講座などで学んでください。

³RPG やノベルゲームのような、プログラマーの仕事の割合が少ないゲームを作っている人にとってはあまり参考にならないかも。

⁴「アイディアプロセッサ」で検索するといろいろヒットします。筆者は使ったこと無し。

⁵筆者は B7 サイズの裏紙をクリップで留めて持ち歩いています

⁶紙に書いてから「あ、やっぱりこのゲームつまんなそう」って気付くことも多々…。

2.3 方針

どんなゲームにするのか、どうやって操作するのか、どういうルールにするのか、前もって考えておくと作りやすくなります。ルールはあまり複雑すぎるとわかりにくいゲームになってしまいますし、かといって平凡すぎても面白みがありません。何か他のゲームには無い「ひとひねり」が必要です。

とは言っても、作っている最中に新しいアイディアが生まれたり、途中でだんだん方針が変わってくることもあるので、最初のうちはあまり深く考え込まなくても大丈夫です。

ジャンルにもよりますが、基本方針さえ固まったら細かいルールは作りながら考える、ということもできます⁷。特にパズルゲーム等は、実際に形にしてみないとわからない部分も多いので、さっさと作り始めてしまったほうが手っ取り早かったりします。

3 実装

3.1 言語

基本的に何でも構いません。普段から使い慣れているものを使用すればいいと思います。ただし、HSP などの大きなプログラムを作るのに向いていない言語はやめておいた方が無難です。

3.2 ものの大きさを決める

簡単に思うかもしれませんが、キャラクターやオブジェクトの大きさは後々まで大きく影響します。また、途中で変更しようと思うとバランス調整が大変なことになってしまいます。ここだけは融通が利かないので、慎重に決めてください。

3.3 メイン部分を真っ先に

メニュー等のゲームと直接関係無い部分は後回しにして、いきなりメイン部分から作ってしましましょう。まずはソースコードが汚くなくても構わないので、とにかく頭の中でできているものを形にすることが大事です。

⁷筆者は頭が固いので、この方法だと大抵途中で行き詰まります。

3.4 リファクタリングは気が向いたときに

これに関してはゲームプログラミングに限った話ではありませんが、大きなプログラムを組んでいるとだんだんソースコードが汚くなってきます。それを汚いまま気にせずコーディングを続ける人、気に入らなくなって何度も作り直す人などいろいろな人がいますが、私は「気が向いたとき」や「ちょっと汚くなってきたな」と思ったときにリファクタリングする程度がベストだと思います⁸。ここは人によって意見が分かれそうですね。

4 バランス調整

一番重要な部分です。バランス調整の腕前次第で「神ゲー」にも「クソゲー」にもなります。大袈裟に聞こえるかもしれませんが、バランス調整が上手な人は本当に上手です。特に、ルールがシンプルなゲームの場合は、このバランス調整でゲームが全て決まってしまう⁹。

4.1 難易度

難易度は、ユーザーが自由に選択できると取っ付きやすくなりますが、必ずしもその必要はありません。ステージクリア型のゲームならば少しずつ難しくしていったり、CPU と対戦するタイプのものは動作速度を変えるなどして微調節することができます。

個人的な意見ですが、製作者にとって「ちょっと難しいくらい」が丁度良いと思います。ゲームスピードを半分程度にまで落としてテストするのも良い方法です。

4.2 意図しないプレー

作ったゲームが、制作者の意図していなかったスタイルでプレーされることがあります。シューティングゲームがパターン化によって極端に難易度が下がったり、レーシングゲームやパズルゲームが決まった行動をとるだけの作業ゲームになっていたり。格闘ゲームの「待ち」なども同様の問題でしょう。

たいていはバランス調整によってどうにかになりますが、プログラムの不具合が絡んでくるとどうにかできないこともあります。

4.3 プログラムの不具合

ここで言う不具合というのは、ゲームが続行不可能になったりゲームバランスを破壊するような致命的なものではなく、「進行上はあまり問題ないがゲームに影響を

⁸当然ですが、事前に必ずバックアップをとっておきましょう。

⁹ゲームのみの部屋 [3] のストリートファイター2 > キャラ紹介編のページを見れば、製作スタッフがどれ程必死に調整を重ねてきたかがわかるかと思えます。

与えるもの」です。有名な例では「前キャン」「ドリドリ」「悪路無視走行」¹⁰などがあります。

不具合に対しては対処のしかたがいくつかあります。

新しい要素として認める

偶然の産物を怪我の功名として考え、正式にルール上で認めるという対応です¹¹。

ゲームバランスが大きく変わる可能性があるので、ここはできるだけ慎重に決めましょう。

うまくバランス調整する

要素そのものは残しつつ、他の部分の調整することでバランスを保つ、という対応です¹²。

潰す

堅実な対応です。影響が大きいつきなど、どうしようもないときは素直に潰してしまうのが一番です¹³。

5 最後の仕上げ

グラフィックと衝突判定が合っているか、BGM の鳴り始めるタイミングはずれていないか、など、細かいことですが最後によく確認しましょう。

最後までモチベーションが維持できなくて、ゲームメニュー等はいおろそかになってしまいがちですが、ここを頑張ると見た目の印象がとても良くなります。

6 おわりに

偉そうなことをつらつら書いてきましたが、何か参考になる部分はありましたでしょうか？

…私もそろそろ何か作らないとなあ。

参考文献

- [1] 「パネルでポン」開発者インタビュー
<http://www.intsys.co.jp/game/panepon/p09/index.html> .
- [2] クソゲー - Wikipedia
<http://ja.wikipedia.org/wiki/%E3%82%AF%E3%82%BD%E3%82%B2%E3%83%BC> .
- [3] T.Akiba ゲームのみの部屋
<http://games.t-akiba.net/> .

¹⁰詳細は割愛。有名ですのでぐぐればすぐ見つかります。

¹¹パネルでポンの時間差連鎖 [1] などはこちらあたります。

¹²格闘ゲームの「待ち」も重要な戦略のひとつとらえて、ガードクラッシュなど別の部分でバランス調整するなどしています (ZERO3)。開発者の苦勞が伺えます。

¹³例えば「前キャン」は GC 以降のバージョンでは削除されています。