システム開発から見た
ゴースト開発

話す人: フィーネ・ラグサズ

話す人

フィーネ・ラグサズ

ロッシュの限界
カッシーニの空隙
オールトの雲
エンケの空隙
EKBO Phase 2
電子遊泳/プリステラ
ゴースト1とアレゲな仲間たち
創者の憂鬱
濡れ羽色の恋
悪意なき悪意
リビングフィールド

のゴーストマスター兼物書き。FSW「Project Roche Limit」の人
http://blankrune.sakura.ne.jp/

このお話の趣旨

システム開発とゴースト開発には非常に共通項が多い。
そのため、システム開発のノウハウがゴースト開発に役立つ。【要出典】

という観点からかる一くお話したいと思います。

システム開発の流れ

いくつかわけ方や考え方あるので一例として

1. 要件定義……どんなものが欲しいのか聞いて整理すること
- 基本設計……どんな形がいいのか大まかに設計すること
- 詳細設計……基本設計を元にして細かく画面や機能の設計図を書くこと
- 実装……プログラム本体を作ること
- テスト……**バグ探しと修正、一番大変とも言われる**
- リリース……完成したらリリース。この後、契約に応じて修正や機能追加なども

このような段階分けができます。

ゴーストの開発も似たようなことが

だからといって、これの通りにやらなければならない
という話ではありません。
順を追って整理して考えていく方法もあるよ、と思って聞いてやってください。

要件定義をゴースト開発で言うならば

○○なゴーストが欲しい!

どんなゴーストが欲しいのかをおおまかに考える段階です。

- ツンデレなゴーストが欲しい
- ミニゲームを搭載したゴーストが欲しい
- コミュニケートが充実したゴーストが欲しい
- ベースウェアの限界性能を引き出したゴーストが欲しい

などなど。

基本設計・詳細設計

先の要件定義の段階で決めたものを参考に詳しく考える段階です。

- どのような性格なのか?
- どのようなシェル なのか?
- 搭載する機能は?
  - 好感度
  - フェイズ
  - ミニゲーム
- 使用するSHIORIやSAORIは?
- ネットワーク更新は?

などを考えていきます。
思いついたことは片っ端からメモしておきましょう。

実装・テスト・リリース

このあたりは実際、そのままです。
辞書やシェルを作って、動作の確認をし、リリースします。

応用できそうなこと

辞書を書く段階で役に立ちそうなネタを適当に拾ってきました。

- わかりやすい名前をつけよう
- 書き方に規則性を持たせよう
- コメントをつけよう

これらの狙いは
辞書を書くときや後から読むときに楽をすること
です。
次のページから細かくみてみましょう。

わかりやすい名前をつけよう

関数などにそれぞれ名前をつけると思います。
この名前には一目でわかる名前をつけましょう。

一文字や他にも被りそうな名前はわかりにくいです。
例： a / i / hensu1

ある程度、意味のある単語で被らないように工夫しましょう。
例： usrNm / ユーザ名

意味が通る範囲で母音を省略するなどして、短い変数名にするのが通例です。

書き方に決まりをつくろう

改行やカッコの使い方は全体で統一しましょう。
そうすると辞書が読みやすくなります。
関数単位でばらけていたりすると、それだけで読みにくくなります。

ですます調とであるだ調が入り混じっている文章をイメージするとわかりやすいと思います。

コメントをつけよう

だいたいの栞ではコメントを挿入することができます。
実際の処理には関係ないのでなくても問題ありません。

が、一週間ぐらい経ってから読み返してみると意味がわかりにくかったり、理解するのに時間がかかったりします。

どのような役割のものなのか、どのように使うのかなどコメントとして残しておくとわかりやすくなります。

まとめ

- システム開発の流れとゴーストの開発の流れは似ている
  - 段階を追って考えるやり方は流用できる
- 辞書を書く際はルールを決めると読みやすくなる
  - 名前はわかりやすいものにする
  - 書き方に規則を持たせる
  - コメントをつける