|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 仕様書 | | マップ画面の設計 | | |
| 種類 | 説明・仕様 | | 作成者 | たかなし |
| 日付 | 2014/8/22 | |

**1　内容**

　自分の頭の整理もかねて、マップ画面シーンの設計を行う。

**2　構造**

　マップ画面は、画面上に座標で配置されたポイントと、ポイント同士を結ぶパスからなる。プレイヤーはあるポイントからスタートし、十字キーで進む方向を選択肢、パスを経由して移動する。

　ポイントには戦闘ポイント、アイテムポイントなどいくつかのタイプがある。基本的にマップに一つ配置されたボスポイントで戦闘をクリアするとマップクリアとなる。

**3　各ポイントの機能**

3. 1　エントリーポイント

　マップの入り口。

　特に機能はなく、自由に通過出来る。

3. 2　戦闘ポイント

　そのポイントに初めて到達した時、強制的に戦闘に突入する。

　勝利した場合は戦闘済み状態となり、プレイヤーはそのポイントに移動する。

　戦闘済み状態になった場合、以後自由に通過出来る。

　敗北・逃走した場合は最後に居たポイントに戻される。

3. 3　探索ポイント

　魔力を消費して探索を行うことが出来、ランダムにアイテムを入手する。

　アイテムを入手した場合は探索済み状態となる。

　通常自由に通過出来る。

3. 4　地点ポイント

　マップのレイアウト上配置される、特に機能のないポイント。

　自由に通過出来る。

3. 5　ボスポイント

　戦闘ポイントと同じだが、勝利するとマップクリアとなる。

　通常より強い敵が現れる。

**4　プレイヤーに関する機能**

4. 1　マップからの脱出

　基本的に、どんな状態でもマップから離脱することが出来る。

マップはクリアにならないが、特にペナルティはない。

4. 2　クイックセーブ

　戦闘は運要素が大きく、いつでもセーブできるようにするとリセットの嵐になってしまう恐れがあるので、クイックセーブを導入する。

　クイックセーブは、ゲームを終了すると、そこから始めることが出来るが、始めるとすぐにそのデータが使えなくなる、すなわち中断時に作成されるが複数回使用することは出来ないようなセーブシステム。

　このシステムを使う関係上、マップ上のギミックやどこまで解放されたかというデータは保持しておく必要がある。というか、マップシーンと戦闘シーンを行ったり来たりする関係上、マップの現在の状態をファイルに記録する機能は必須。プレイヤーの位置も、グローバル変数として保持するのではなく、マップシーンのメンバとして保持し、それも含めてセーブする方が良い。

4. 3　アリスの特技

　アリスは魔力を消費して特技を使用することが出来る。

**5　マップの機能**

5.1　各ポイントの表示状態

　ダンジョンの探索っぽくするため、訪れていないポイントは非表示にする。

5.2　クリアデータの保存

　各マップについて、どのポイントをクリアしたか、ボスを倒したか、といったデータを保存する必要がある。

5.3　マップ上のギミックの状態の保存

　跳ね橋などのギミックがある場合、プレイ中はそれを保存できる必要がある

（バトル画面への以降時など）。

　これらを考えると、ギミックもマップデータの一部として保存できるべきである。

**6. マップデータの保存形式**

　マップデータは、以下の二種類の方法で保存する機能をつける必要がある。

A. マップのクリアデータ

　ゲームのセーブデータに含まれるデータ。

　マップの追加にも対応するため、可変長データとして読み込む。

　・各ポイントを通過またはクリアしたことがあるか

　・マップ全体として、クリアしたことがあるか

B. マップの攻略状態を表すデータ

　戦闘画面に移行する時や、クイックセーブの際に記録するデータ。

　一時ファイルとしてゲーム内で一つのみ保存し、特定のセーブデータには付随しない。

　・各ポイントを通過またはクリアしたことがあるか

　・各ポイントの戦闘や探索が終了しているか

　・プレイヤーの位置

　・マップに入ってからの時間

　・各ギミックの状態

**7　設計**

7. 1　マップのデータそのもの

　データベースとしてのマップ。

7. 2　クリア状態を表す

　まず、6.Aマップのクリア状態を表すデータを考える。セーブデータには、マップの攻略状態をマップ毎に全てくっつけたものを保存する必要がある。

　Record\_Mapクラス（Record\_XXXは、データベースに似ているが、内容が更新されるクラス。基本的に内容はセーブファイルに含まれる）を利用してデータを保存する。

簡単のため、各マップについて、このRecord\_Mapのデータ長さは同じになるようにする。

・マップ画面において、マップポイント数の上限を決め、配列としてデータを保持する。

・各マップにおいてポイント数がそのポイントより少ない場合、そのポイントは使用しない（アクセスされない）データとする。

7. 2　ギミックも含めてマップ情報全体を保持する

　クリア状態を表すだけなら