

10月30日分 enPiT 進捗レポート

チーム：スケジュールのプロ メガネ支部
メンバー：島, 川畑, 古田島, 長, 太田, 小林

● 全体の予定と現在の進捗

全体の予定		現在の進捗
11月	<ul style="list-style-type: none">・スケジュール, タスク管理機能の実装・仮データによるタスク処理自動スケジューリング機能の実装・イメージキャラクター作成	<ul style="list-style-type: none">・スケジュール, タスクの追加および編集画面の改良中である・スケジュールの1週間毎の繰り返し入力用のUIを作成した・カレンダー表示APIの学習中である・アプリケーションのソースコードの簡略化を行っている・タスク処理スケジューリングのアルゴリズムを考えた・Live2dで動かす用のキャラクターの立ち絵を制作中である・Live2dで作成したキャラクターをWeb上で動かすAPIについて学習中である
	<ul style="list-style-type: none">・サーバ構築<ul style="list-style-type: none">・クラスタ機能の実装・ユーザアカウント管理機能の実装・データベース構築・サーバ上のデータを用いてタスク処理自動スケジューリング機能の実装・Chrome拡張機能を用いたユーザへのタスク期限通知機能の実装	<ul style="list-style-type: none">・レンタルサーバへのSSH接続を確立した・現状のアプリケーションをサーバ上にアップロードした・DBをサーバ上に設置する方法を学習した・DBにアクセスするための関数群を作成し, ローカル環境にて動作確認を行った・仮データを用いてユーザ情報変更機能を実装した・ユーザ情報変更機能の更なる実装を行った・Chrome拡張機能の実装に関する学習中である
	13日 中間報告	
	<ul style="list-style-type: none">・UI, キャラクターイラスト作成・実サーバでのテスト, デバッグ	<ul style="list-style-type: none">・UI設計の改良を行った・アプリケーションの使い方や機能の説明を知らせるチュートリアルを作成中である
	27日 レポート提出	
12月	11日 最終発表	

- 問題点とその解決策

- 未だに GitHub に使い慣れない。GitHub flow ? なにそれ美味しいの？

- まずは GitHub flow とは何か，ミーティング時に全員で学習を行う。

- <http://b.pyar.bz/blog/2014/01/22/github-flow/>

- その後全員で試し，今後は GitHub flow で各自作業を進める。

- プロダクトの進捗をチーム外の方が見るできない

- サーバ上にソースコードをアップロードしたので，アプリケーションへのリンクを GitHub の README に示す。

- 今後 2 週間の予定

- タスクの自動スケジュールをアルゴリズムに基づいて実装する

- タスク管理機能をより使いやすくするため，UI を改良する

- サーバ上に DB 環境の構築を行い，その動作を確認する

- DB とアプリケーションとの連携を行う

- Chrome 拡張機能による通知表示の実装を行う

- Web アプリケーション上で Live2d を用いたキャラクターを動かす

- タスク処理の自動スケジューリングアルゴリズムについて

各タスクには「重さ」と「締め切り」が存在する。タスクの重さを X ，締め切りまでの日数を D とすると，各タスクの重要度は (X/D) で表される。タスクの重さ 1 につき 1 時間をタスクの処理時間とする。タスクを締め切り順にソートし，スケジュールの空いている時間に X 時間ずつ処理時間として割り当てていく。この時，以下の条件に従って割り当てを行う。

1. ユーザに徹夜をさせないという目的のため，毎日 24 時～翌日 7 時まではタスク処理スケジュールは入れない。
2. 1 日のうち，最大タスク処理時間は 10 時間とする。
3. タスク処理が 3 時間連続したら 1 時間空き時間を挟む。もともと存在するスケジュールとタスク処理の間には空白の時間は挟まなくてよいこととする。

タスク処理のスケジュール期限が来るたびにユーザに通知を行い，タスク処理が完了したかどうかを確認する。タスクが完了していない場合は，ユーザは残りタスクの「重さ」を再入力する。再入力されたタスクとその次のタスクの重要度を比較し，重要度の大きなものから処理を行うよう再スケジューリングを行う。この時，締め切りまでに処理スケジュールを確保できない場合は（自業自得なので）残り時間にタスク処理の時間を詰め込む。

直近 1 週間の空いている時間を T とすると，1 週間のデスマ具合 S は

$$S = \text{sum}(1 \text{ 週間のタスクの } X) / T$$

と表すことができる。この値が 0.3, 0.5, 0.8 を超えるごとにキャラクターの表情が陰しくなっていくことにより，今こなすべきタスクがどれだけ積まれているのかを直感的に把握することができる。さらに，タスク処理のキャパシティを超えてしまい，締め切りまでに処理時間を確保できなくなった場合，キャラクターは鬼の形相となり，ユーザへの反省を促す。