* Webゲームループ
  1. 50ミリ秒キー入力待ち
  2. キー入力をもとに、テトリスの関数を呼び出して１回動かす
  3. テトリスのフィールドを表示する
  4. 以上を繰り返す（理論上は、20回繰り返すごとに1秒経過します）
* テトリス
  + ゲーム管理クラス
    - 計算用フィールドデータ  
      縦２３マス、横１２マス、計２７６マスの二次元配列  
      または、要素数276個の一次元配列

|  |  |
| --- | --- |
| データ値 | 内容 |
| -1 | 左右と底の外壁。 |
| 0 | 何もない。 |
| 1 | T型を表す。マゼンタ色 |
| 2 | J型を表す。青色 |
| 3 | I型を表す。水色 |
| ４ | O型を表す。黄色 |
| ５ | L型を表す。橙色 |
| ６ | Z型を表す。赤色 |
| ７ | S型を表す。緑色 |

* + - テトロミノデータ  
      回転軸をxy座標の中心として、4つのブロックを要素数８個の一次元配列で表す  
      T型（**tetroT**）[-1, 0, 0, 0, 1, 0, 0, 1]  
      J型（**tetroJ**）[-1, 0, 0, 0, 1, 0, 1, 1]  
      I型（**tetroI**）[-1, 0, 0, 0, 1, 0, 2, 0]  
      O型（**tetroO**）[ 0, 0, 1, 0, 0, 1, 1, 1]  
      L型（**tetroL**）[-1, 0, 0, 0, 1, 0,-1, 1]  
      Z型（**tetroZ**）[-1,-1, 0,-1, 0, 0, 1, 0]  
      S型（**tetroS**）[ 0, 0, 1, 0, 0, 1,-1, 1]

中心

(0,0)

(0,-1)

(1,0)

(1,-1)

(-1,0)

(-1,-1)

(0,1)

(1,1)

(-1,1)

**tetroT**の例

* + - カレントデータ  
      操作中のテトロミノのxy座表と向きを表す３要素の配列  
      現在の移動不可能回数を数え上げる変数
    - 画面表示用のフィールドデータ  
      計算用のフィールドデータにカレントデータを合体させてウェブ側に渡す。
    - ブロックの定着と生成  
      動けなくなったブロック（カレントデータ）をフィールドデータに書き込んで、新しいテトロミノをカレントデータとして生成する。  
      **def next()**引数 無し  
      戻り値 無し
    - ブロックの移動  
      ブロック（カレントデータ）を指定の方向に指定した距離動かす。  
      **def move(down=0,left=0,right=0)->None**  
      引数

|  |  |
| --- | --- |
| 型と変数名 | 内容 |
| int down | 下方向への移動 |
| int left | 左方向への移動 |
| int right | 右方向への移動 |

戻り値 無し

* + - ブロックの回転  
      ブロック（カレントデータ）を時計回りに回転させる。  
      **def rotate()->None**引数 無し  
      戻り値 無し
    - 当たり判定  
      ブロック（カレントデータ）の動作(**move**関数および**rotate**関数)から呼び出す。ブロックが移動可能か(壁や他のブロックと被らないか)判定する。  
      移動可能であればTrueを返し、移動不可能であればFalseを返す。  
      **def judge(after\_current)->bool**引数

|  |  |
| --- | --- |
| 型と変数名 | 内容 |
| after\_current | テトロミノの移動先のxy座標と方向を表す３要素配列  次のカレントデータになる候補 |

戻り値

|  |  |
| --- | --- |
| 型と変数名 | 内容 |
| can\_move | 移動できるならTrueできないならFalse |

* + - ライン消し  
      ブロック（カレントデータ）が確定して、フィールドデータに書き込まれたら、フィールドデータを行ごとに見て、1行（10マス）が埋まっている行を探索する。その行のフィールドデータを削除して、それより上の行のフィールドデータを下に切り詰める。  
      **def deleteLine()**引数 無し  
      戻り値 無し