ゴールキーパー(設計)

• クラス名 ⇒　**Goalkeeper**

• 主担当者名 ⇒　岡部亜沙美

• 属性（変数）一覧

|  |  |  |
| --- | --- | --- |
| 属性名 | 型 | 説明 |
| x | number | キーパーの現在のX座標の位置。ゴールの横方向の位置を表す。 |
| Y | Number | キーパーの現在のY座標の位置。通常は固定値で中央付近。 |
| width | Number | キーパーの横幅。ブロック判定時に必要になってくる。 |
| height | Number | キーパーの高さ。 |
| movePattern | String | 移動のパターンを表す文字列(例：random,horizontalなど) |
| speed | Number | キーパーの移動速度 |

• メソッド一覧

|  |  |  |  |
| --- | --- | --- | --- |
| メソッド名 | 引数 | 戻り値 | 説明 |
| move() | なし | なし | movePattern  に従って、キーパーの位置を更新する。 |
| isBlocking(x, y) | x :number  y :number | Boolean | ボールの飛来位置（クリック座標）がキーパーの位置内にあるかを判定する。 |
| setPosition(x, y) | x :number  y :number | なし | キーパーの位置を強制的に設定する。 |
| updatePattern(pattern) | pattern :string | なし | 移動パターンを変更する。 |
| draw(context) | context: CanvasRenderingContext2D | なし | キーパーをキャンバス上に描画する。 |

• 各属性，各メソッドの内容に関するメモ

**属性**

x, y：キーパーの表示位置。y座標は常にゴールエリアの一定値でも良いかな？

width, height：ブロック判定時に使う。例えば、ボールの当たり判定と比較するため。

movePattern：動き方を表す文字列。たとえば "horizontal" の場合は左右に往復運動、 "random" はランダム移動。

speed：1フレームごとの移動量を指定。動きが速すぎると難易度が上がる。難易度変更に利用できるかも？

**メソッド**

move()：movePattern に応じたロジックを内部に実装。横移動やランダムジャンプなど。

isBlocking(x, y)：座標がキーパーの矩形領域（x〜x+width, y〜y+height）内にあるかを判定する。

setPosition()：テストやリセット時に使用。ゲーム開始時に中央に配置する用途にも使える？

updatePattern()：ゲーム中に移動パターンを切り替えたい場合に利用。

draw()：Canvas API を使用する場合に、キーパーの可視化を担当。