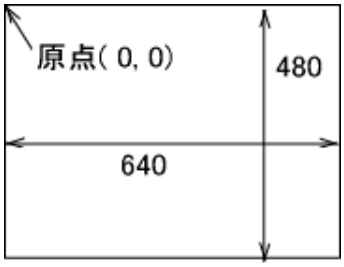
第４回　当たり判定１

当たり判定の前にまずは画面の外にまで行けてしまうので見えない壁を作りたいと思います。



画面の大きさは横 640 ピクセル、縦 480 ピクセルです。画面の左上が座標(0, 0)で画面右下が(639, 479)となります(幅が 640 なので 0 からカウントするので 639 が 640 番目の数字になります)。

さて、見えない壁を作るという話でしたが、自機座標の x 座標が 0～639、y 座標が 0～479 までの範囲に収まっていればいいわけです。なので if 文を使ってそれ以外の座標に出たときは、その座標に戻されるような書き方をすればいいです。

```
if( jiki.x < 0) jiki.x = 0;
if( jiki.x > 640) jiki.x = 640;          //(面倒なので 640 で丸め込み)
```

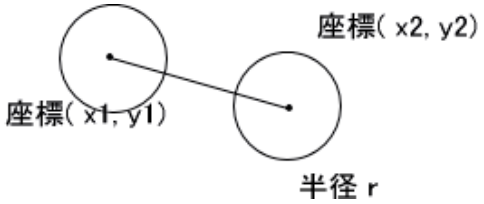
とかこんな感じになります。y 座標に対しても同じように記述してください。

これで円満解決といきたいところですが、これだと中心座標で判断しているので自機の半分は画面がにいっていることになります。細かいことには無頓着な人以外は自機の全体が画面全体にでないようにしてみてください。

いよいよ当たり判定となります。当たり判定はシューティングゲームやブロック崩しなどいろいろなところに使えるので、ぜひマスターしてください。まずは当たり判定がなんたるかを説明します。

当たり判定の原理

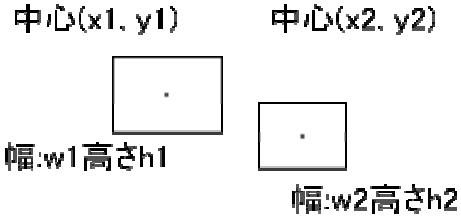
当たり判定というのは二つの物体が接触しているのか、接触していないのかを判定することです。“そんなの一目見ればわかるじゃーん”と思うかもしれませんが、パソコンは座標などをすべて数字として扱うので、数学的に記述しなければなりません。



例えば次のように円が二つあったとします。この二つの円が当っているという条件は

$$\sqrt{(x1-x2)^2+(y1-y2)^2} < (r1+r2)$$

となります。中学３年生以上ならきっと分かると思います。



次に矩形(四角形)の判定です。矩形は円と違って縦と横の概念があるので二回当たり判定をする必要があります。

$$(x1-x2)<(w1+w2) \quad \text{かつ} \quad (x2-x1)<(w1+w2)$$

となります。二物体の中心座標の差が、二物体の幅の和より小さければ当たっているといえます。これだけだと横方向だけなので、縦方向に関しても同じことをすれば矩形が当たっていると言えます。

円と矩形の当たり判定さえ理解できれば、複雑な当たり判定も大抵はこの応用でどうにかなります。

それでは実際にプログラムで書いてみます。今回のアクションゲームは画像などはすべて四角形なので矩形の当たり判定の関数をまず書きます。

```
int g_atari( int x1, int haba1, int x2, int haba2){
	//当たるかどうかを調べる関数(一次元のみ)
	//引数は、座標とその物体の大きさ
	//当たる場合は戻り値が 1。当たらない場合は 0 を返す

	if( (x1 – x2) < ( haba1 + haba2) / 2 && ( x2 – x1) < ( haba1 + haba2) / 2){
		return 1;
	}
	return 0;
}
```

矩形同士の当たり判定の関数です。game.cpp あたりにでも追加しておいてください。当たっていれば 1 、当たっていなければ 0 を返す関数です。return は一つの関数に複数個あっても構わないのでこんな感じになります(但し、必ずどのルートを通っても return があるようにしなければならない)。

```
void g_teki( void){
	//敵に関する関数
	int i;

	for( i = 0; i < TEKI_KAZU; i++){
		if( teki[i].flag == 1){
			//teki[i].flag = 1 で敵がいる

			if( time % 100 < 50){
				//0 ～ 50 フレーム目だと右
				teki[i].sokudo_x = TEKI_SOKUDO;
			}
```

```
                        else{
                                //50 ～ 100 フレーム目だと左
                                teki[i].sokudo_x = -TEKI_SOKUDO;
                        }
                        teki[i].x += teki[i].sokudo_x;
                        //速度を座標に代入する

                        if( g_atari( jiki.x, jiki.haba, teki[i].x - wallx, teki[i].haba)){
                                if( g_atari( jiki.y, jiki.takasa,
                                        teki[i].y - wally, teki[i].takasa)){
                                        //自分が敵に触れたら敵が消滅する
                                        teki[i].flag = 0;
                                }
                        }

                        DrawGraph( (teki[i].x - teki[i].haba / 2) - wallx, (teki[i].y -
teki[i].takasa / 2) - wally, teki[i].handle, true);
                        //敵を描画する
                }
        }
}
```

これは自機と敵が当たった場合、敵が消滅するというプログラムになります。これだとこっちが敵のような気もしますが、これで当ったら何かアクションを起こすというのを確認してください。