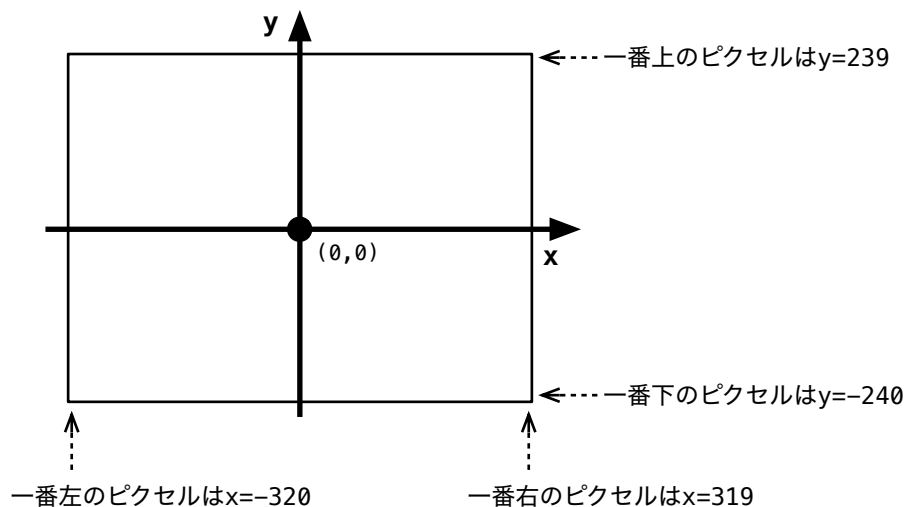


▶ 画面の座標系

画面サイズは640×480ピクセル固定です。画面中央が原点(0, 0)です。



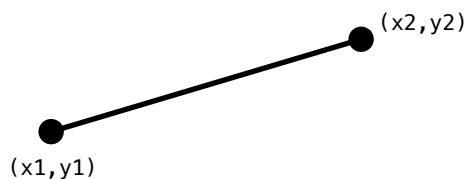
▶ DrawPoint(x, y, color)

点(x, y)の位置に1ピクセルの点を描画します。



▶ DrawLine(x1, y1, x2, y2, color)

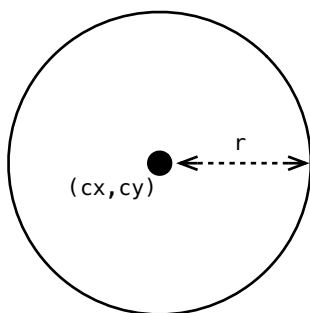
点(x1, y1)と点(x2, y2)を結ぶ線分を描画します。



▶ DrawCircle(cx, cy, r, color)

FillCircle(cx, cy, r, color)

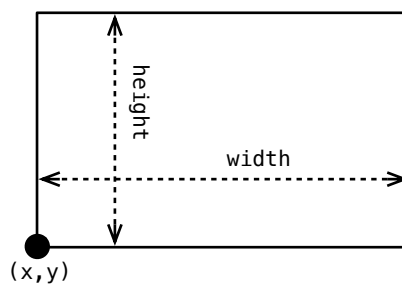
中心点の座標(cx, cy)と半径rを指定して、円を描画します。



▶ DrawRect(x, y, width, height, color)

FillRect(x, y, width, height, color)

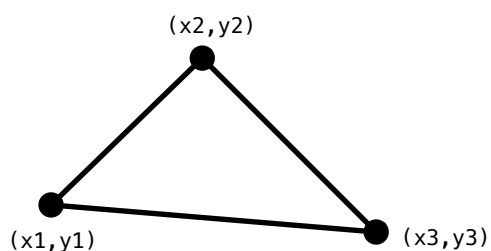
左下の始点(x, y)と縦横のサイズ(width, height)を指定して、矩形を描画します。



▶ DrawTriangle(x1, y1, x2, y2, x3, y3, color)

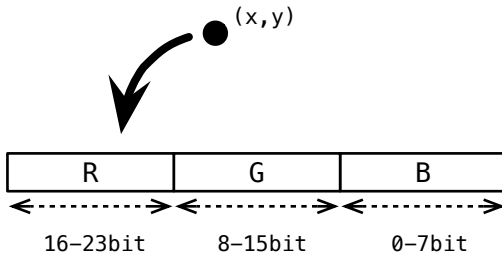
FillTriangle(x1, y1, x2, y2, x3, y3, color)

3点(x1, y1), (x2, y2), (x3, y3)を結ぶ三角形を描画します。



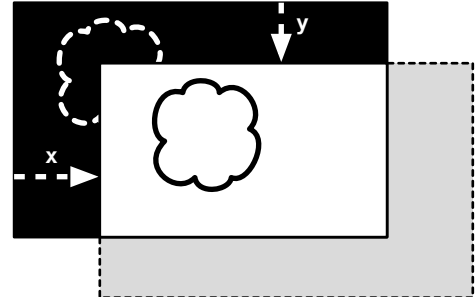
▶ `int GetColor(x, y)`

座標(x, y)の位置の色情報を取得します。
取得したint型の色情報は、下位ビットから順に、
16-23ビット目が赤、8-15ビット目が緑、
0-7ビット目が青の要素で、0~255の値です。



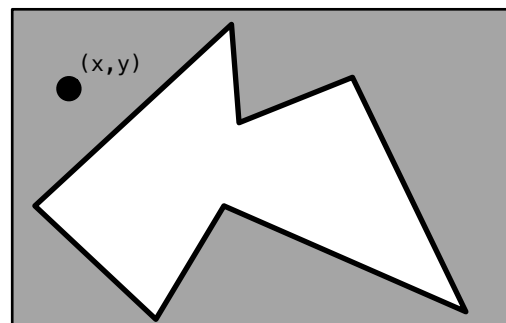
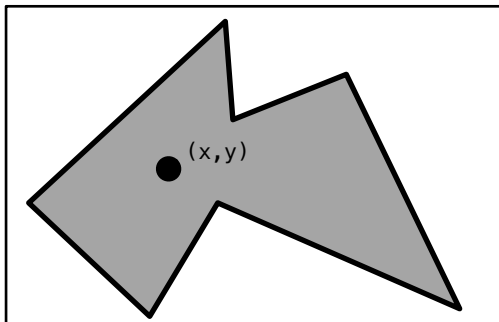
▶ `Scroll(x, y)`

X方向にxピクセル、Y方向にyピクセルだけ画像
バッファをスクロールさせます。スクロールした
結果、何もない箇所はRGB=(0, 0, 0)の黒で埋め
られます。



▶ `Paint(x, y, paintColor, borderColor)`

座標(x, y)を中心として、borderColorの色で囲まれた領域をpaintColorの色で塗りつぶします。
閉領域の内部の座標を指定した場合にはその領域の中を塗りつぶし、外部の座標を指定した場合には
その周囲の領域を塗りつぶします。画面端も塗りつぶしの境界となります。



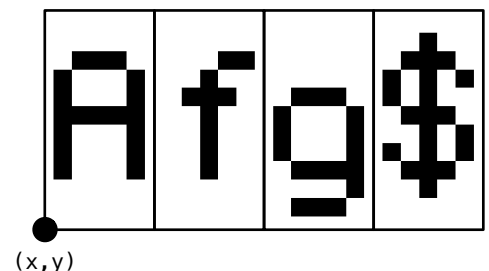
▶ `DrawCharacter(c, x, y, color)`

左下の座標(x, y)を指定して、12x20ピクセルの文字cを描画します。
文字cには、ASCII文字コード内の文字だけが指定できます。



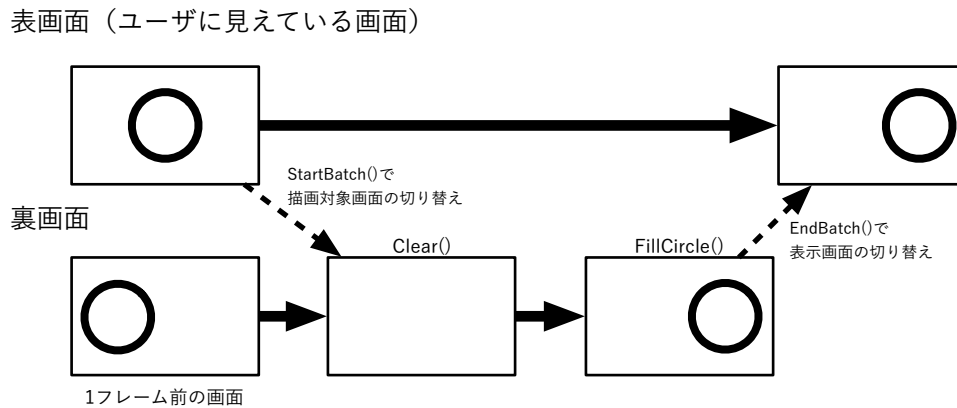
▶ `DrawText(str, x, y, color)`

左下の座標(x, y)を指定して、12x20ピクセルの文字を並べて
文字列を描画します。ASCII文字コード内の文字だけが指定できます。



▶ StartBatch() EndBatch()

StartBatch()を実行すると、描画命令はすべて裏画面に対して実行されます。
EndBatch()を実行すると、裏画面と表画面が切り替わります。
アニメーション処理のちらつき防止のために、この機能を使用してください。



▶ int GetMouseX() int GetMouseY()

マウスカーソルのX座標とY座標を取得できます。

▶ bool CheckMouse()

マウスの左ボタンが押されているかどうかをチェックします。

▶ bool CheckKey(unsigned int key)

引数に指定したキーが押されているかどうかをチェックします。
以下の定数を指定できます。

```
kKeyA, kKeyS, kKeyD, kKeyW    // ASDWキー
kKeyUpArrow, kKeyDownArrow    // 上下の矢印キー
kKeyLeftArrow, kKeyRightArrow // 左右の矢印キー
kKeyReturn, kKeySpace         // returnキー, spaceキー
```

▶ Sleep(float seconds)

秒単位で指定した時間だけ、処理を一時的に休止します。
0.1や2.5のように、小数点付きの値を指定できます。
60fpsで処理を行う場合は、0.0167を指定してください。