Lab 2
Fitts’ Law Experiment

Fitt’s Law

Movement time

Speed of the device (Slope)

Amplitude

$$MT = a + b\log_2(\frac{A}{W} + 1)$$

Reaction time (Intercept)

Width

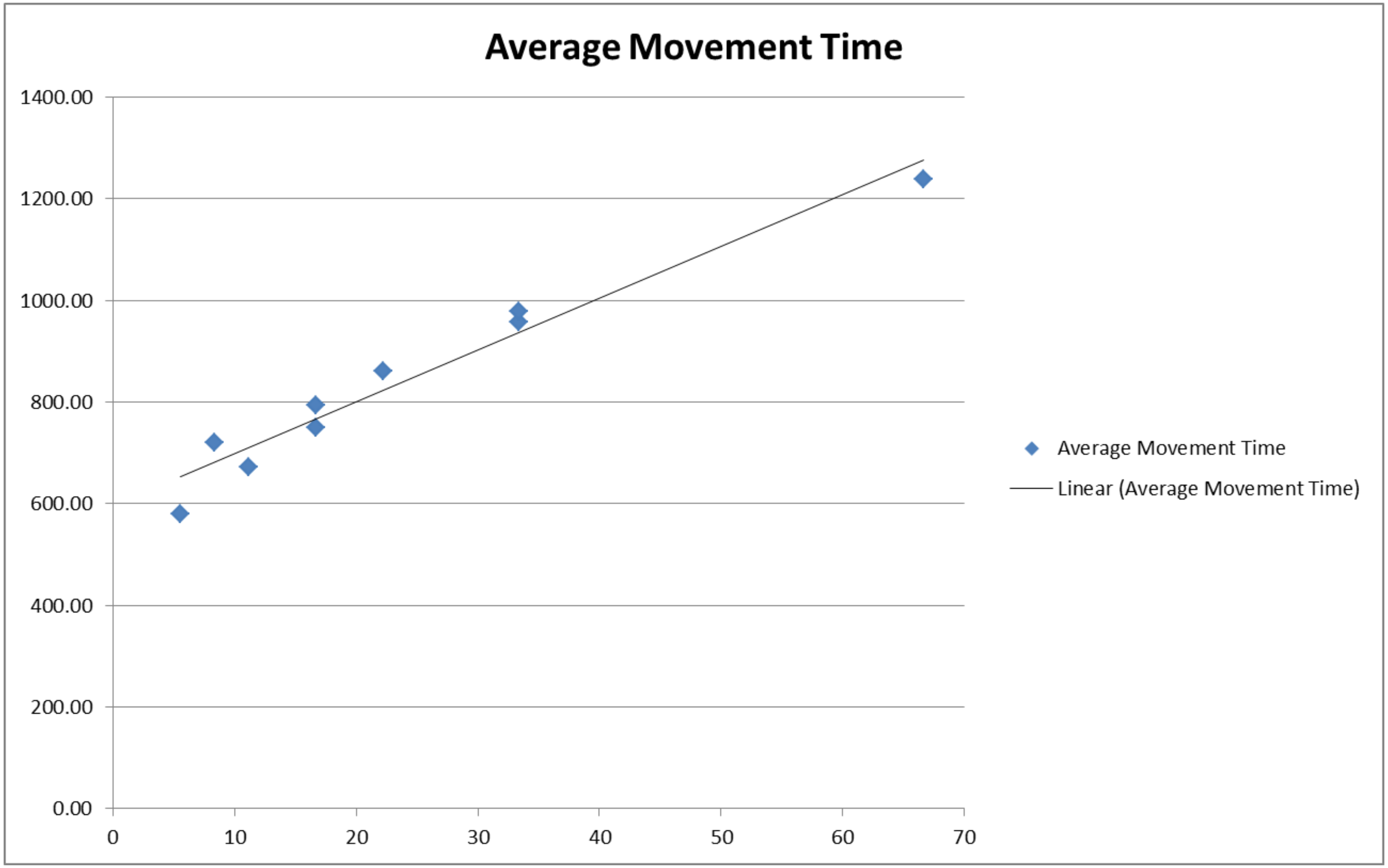
Experiment Design

- Each experimental condition is determined by:
  - Amplitude (A)
  - Width (W)

$$\longrightarrow \frac{A}{W}$$

- Record:
  - Average Movement Time (MT)

Results



Ratio A/W

Fitt’s Law

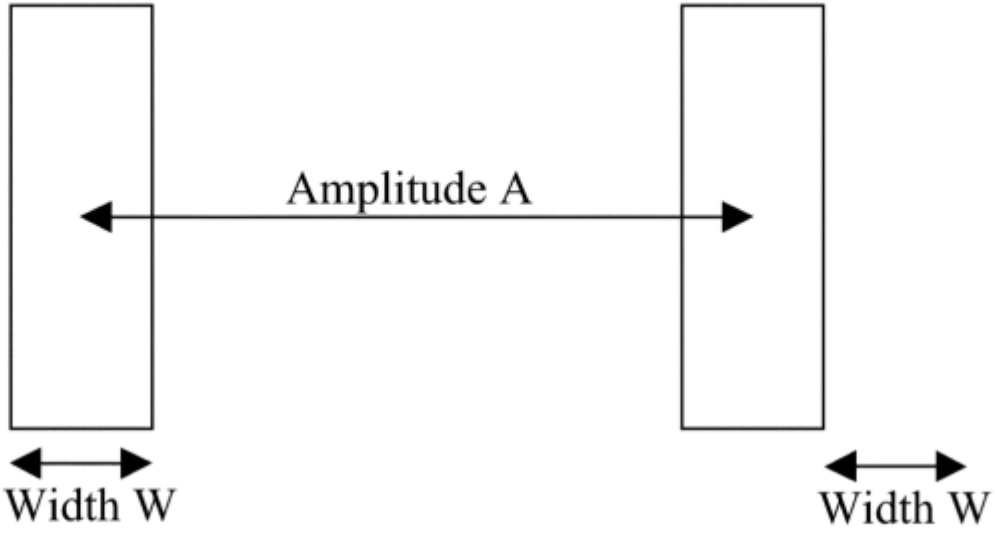


FIGURE 9.6. Fitts’ task paradigm (Fitts, 1954).

Follow the link to try the experiment

http://ergo.human.cornell.edu/FittsLaw/FittsLaw.html

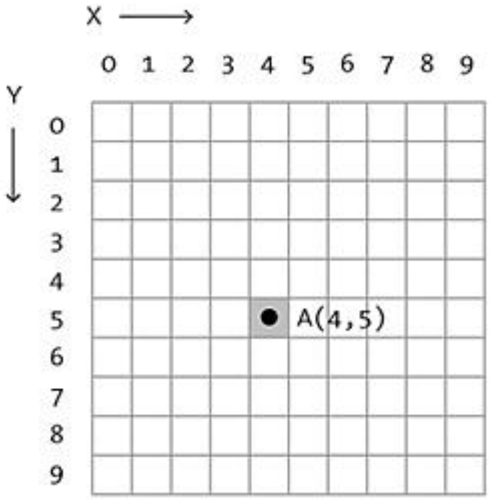
Installing Processing

- Go to processing.org/download
- Download the **Processing 3**
- Unpack the file
- Run processing.exe

Basic structure

```
// runs once during the initialization
void setup(){

    size(800, 600); // window size

}

// 60 calls/second
void draw(){


}
```

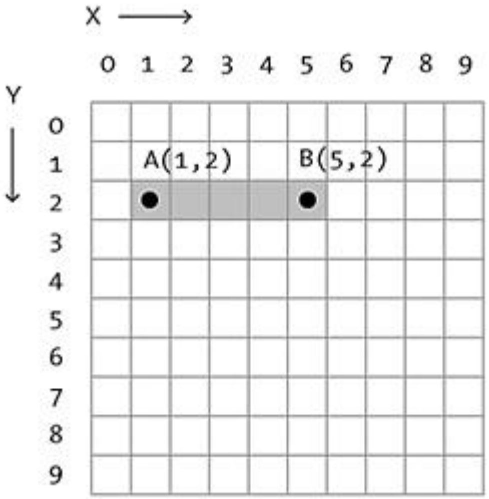
Drawing graphics primitives



```
point(x,y);
```

Example:

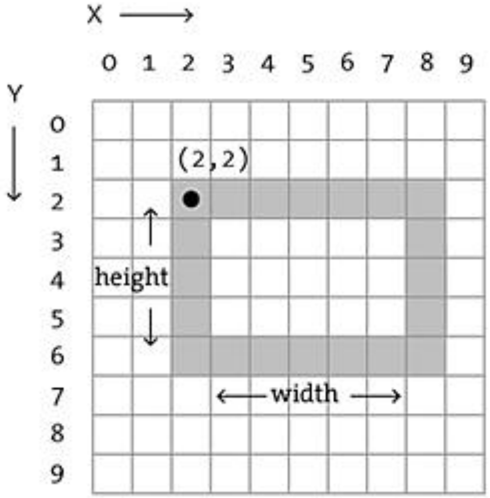
```
A(4,5);
```



$\texttt{line(}\underbrace{\texttt{x1,y1}}_{\text{Point A}}\texttt{,}\underbrace{\texttt{x2,y2}}_{\text{Point B}}\texttt{);}$

Example:

```
line(1,2,5,2);
```



```
rect(x,y,width,height);
```

Example:

```
rect(2,2,7,5);
```

Drawing graphics primitives

```
void draw(){
    ellipse(250, 250, 200, 250);
    rect(400, 200, 150, 50);
    line(100, 500, 500, 500);
    point(250, 250);
    quad(38, 31, 86, 20, 69, 63, 30, 76);
    triangle(300, 75, 320, 20, 340, 75);
}
```

Setting drawing attributes

- Fill Color
  - `fill(150); // gray value`
  - `fill(204, 102, 0); // rgb value`
  - `noFill();`
- Stroke Color
  - `stroke(204, 102, 0); // rgb value`
  - `noStroke();`
- Stroke Weight
  - `strokeWeight(1);  // pixel value`

Setting drawing attributes

```
void draw(){
    fill(204, 102, 0);
    stroke(90);
    strokeWeight(5);
    ellipse(250, 250, 200, 250);
}
```

Mouse Events

- Global Variables
  - `mouseY, mouseX, mousePressed`

- Listeners
  - `mouseDragged()`
  - `mousedPressed()`
  - `mouseReleased()`
  - `mouseClicked()`
  - `mouseWheel()`

Mouse Events

```
int r=0, g=0, b=0;

void draw(){
    fill(r, g, b);
    ellipse(mouseX, mouseY, 80, 80);
}

void mousePressed(){
    r = 141; g = 211; b = 199; // green
}

void mouseReleased(){
    r = 190; g = 186; b = 218; // purple
}
```

Mouse Events

```
int r=0, g=0, b=0;

void draw(){
    background(200);
    fill(r, g, b);
    ellipse(mouseX, mouseY, 80, 80);
}

void mousePressed(){
    r = 141; g = 211; b = 199;
}

void mouseReleased(){
    r = 190; g = 186; b = 218;
}
```

Deliverables

- Processing Code
- Spreadsheet