

TOUCH BASED INTERFACES

By Dannis Mok

Disable default behaviors

- Most of the touch devices allow zoom and pinch by default. Need to disable these by adding below statement in the header

```
<meta name="viewport" content="width = device-width,  
initial-scale = 1.0, user-scalable = no">
```

Touch Event Handlers

- Touch events are similar to mouse events. Can register the event handlers to the canvas.

```
canvas.addEventListener("touchmove", touchmoveHandler,  
false);
```

```
function touchmoveHandler(event)  
{  
    //Find the touch point's x and y position  
    touchX = event.targetTouches[0].pageX -  
canvas.offsetLeft;  
    touchY = event.targetTouches[0].pageY -  
canvas.offsetTop;  
  
    //Prevent the canvas from being selected  
    event.preventDefault();  
}
```

Touch Arrays

- When a touch event is fired, all the touch points on the screen are copied into an array called “touches”
- 3 fingers touched the screen and the touch points' X and Y positions are stored in the `event.touches[0]`, `event.touches[1]` and `event.touches[2]` respectively.
- First touch points' X position can be accessed by `event.touches[0].pageX` and its Y position can be accessed by `event.touches[0].pageY`

Touch Arrays

- All the touches happened are stored in touches array. If just want to capture the touch events on the target element like canvas, we will use the targetTouches array instead.
- Another array called changedTouches stored the touch points actively involved in triggering the event. It can detect the position of the removed finger for example.

Disable default behaviors

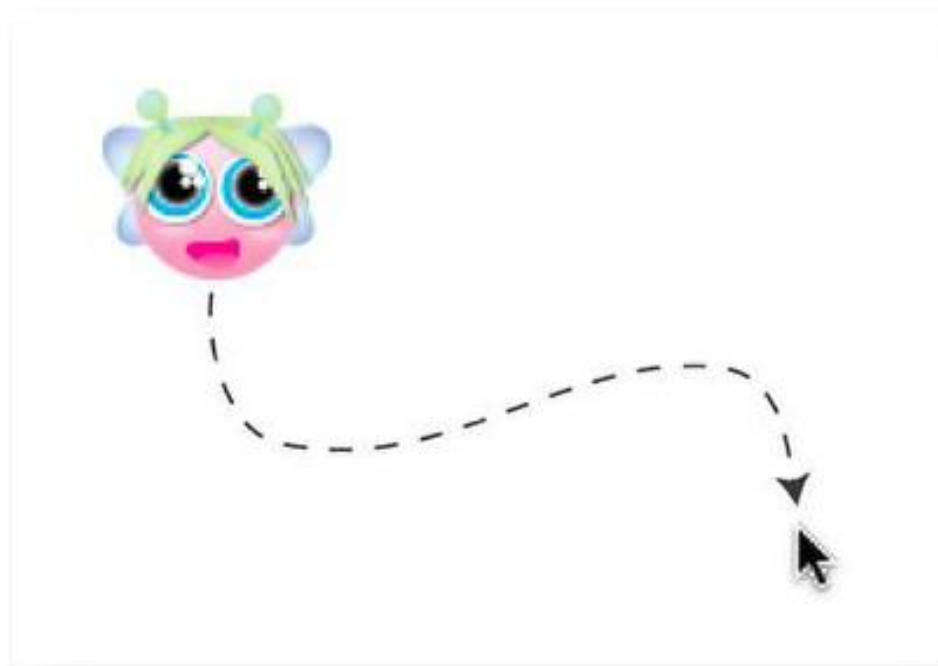
- Default touch events like touchmove will start the scrolling of the browser window. It should be disabled by using

```
event.preventDefault();
```

Touch Events

Touch Events	When it is fired ?	Related Mouse Events
touchmove	When a finger is dragged over the screen	mousemove
touchstart	When a finger touches the screen.	mousedown
touchend	When a finger is removed from the screen	mouseup

Move the character with touch

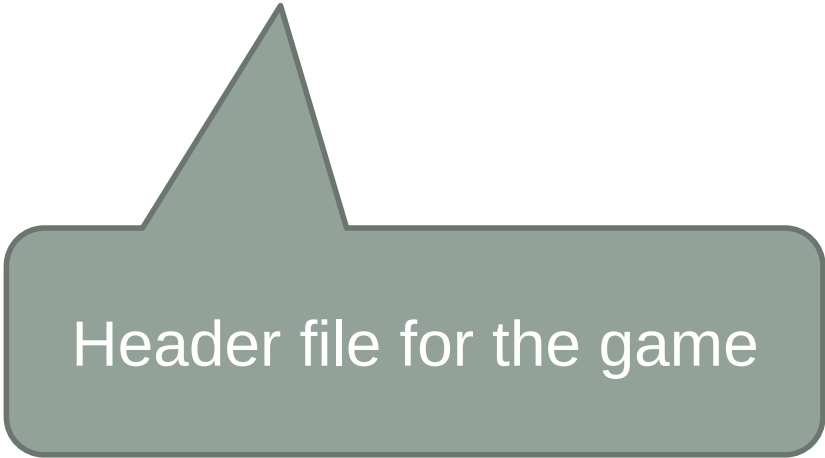


Sample Codes

```
<!doctype html>
<meta charset="utf-8">
<meta name="viewport" content="width = device-width, initial-scale = 1.0, user-scalable = no">
<title>Follow</title>

<canvas width="550" height="400" style="border: 1px dashed black"></canvas>

<script src="requestAnimationFramePolyfill.js"></script>
<script type="text/javascript">
```



Header file for the game

Sprite Template

```
//--- The sprite object
```

```
var spriteObject =
```

```
{
  sourceX: 0,
  sourceY: 0,
  sourceWidth: 64,
  sourceHeight: 64,
  width: 64,
  height: 64,
  x: 0,
  y: 0,
  vx: 0,
  vy: 0,
  visible: true,
```

```
//Physics properties
```

```
accelerationX: 0,
accelerationY: 0,
speedLimit: 5,
friction: 0.96,
```

```
//Getters
```

```
centerX: function()
```

```
{
  return this.x + (this.width / 2);
},
```

```
centerY: function()
```

```
{
  return this.y + (this.height / 2);
},
```

```
halfWidth: function()
```

```
{
  return this.width / 2;
},
```

```
halfHeight: function()
```

```
{
  return this.height / 2;
}
```

```
};
```

Create Sprite Object

```
//--- The main program

//The canvas
var canvas = document.querySelector("canvas");
var drawingSurface = canvas.getContext("2d");

//Object arrays
var sprites = [];
var assetsToLoad = [];
var assetsLoaded = 0;

//The fairy
var fairy = Object.create(spriteObject);
fairy.sourceWidth = 77;
fairy.width = 77;
fairy.x = canvas.width / 2 - fairy.halfWidth();
fairy.y = canvas.height / 2 - fairy.halfHeight();
console.log(fairy.x);
sprites.push(fairy);

//Load the tilesheet image
var image = new Image();
image.addEventListener("load", loadHandler, false);
image.src = "../images/buttonFairy.png";
assetsToLoad.push(image);
```

Touch Event Handler

```
//The touch point (Initialize it to the fairy's center point)
var touchX = fairy.x + fairy.halfWidth();
var touchY = fairy.y + fairy.halfHeight();

//Add a listener
canvas.addEventListener("touchmove", touchmoveHandler, false);

//Game states
var LOADING = 0;
var PLAYING = 1;
var gameState = LOADING;

update();

function touchmoveHandler(event)
{
    //Find the touch point's x and y position.
    //Subtract the canvas's top and left offset
    touchX = event.targetTouches[0].pageX - canvas.offsetLeft;
    touchY = event.targetTouches[0].pageY - canvas.offsetTop;

    //Prevent the canvas from being selected
    event.preventDefault();
}
```

Main Programs

```
function playGame()  
{  
    fairy.x = touchX - (fairy.halfWidth());  
    fairy.y = touchY - (fairy.halfHeight());  
}  
  
function update()  
{  
    //The animation loop  
    requestAnimationFrame(update, canvas);  
  
    //Change what the game is doing based on the game state  
    switch(gameState)  
    {  
        case LOADING:  
            console.log("loading...");  
            break;  
  
        case PLAYING:  
            playGame();  
            break;  
    }  
  
    //Render the game  
    render();  
}
```

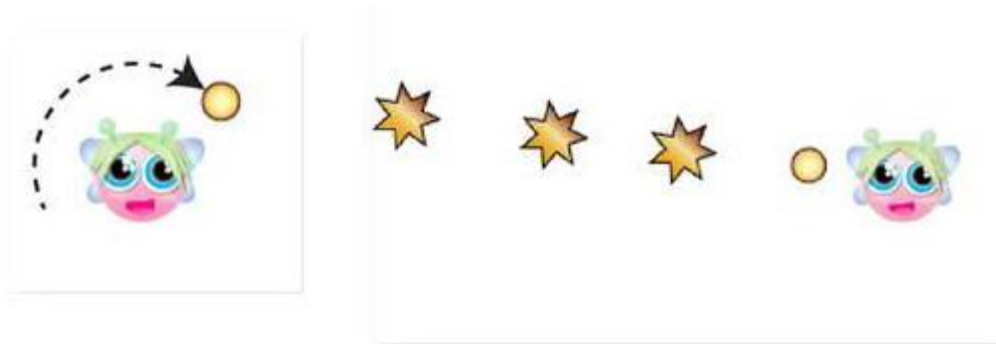
Rendering

```
function loadHandler()  
{  
  assetsLoaded++;  
  if(assetsLoaded === assetsToLoad.length)  
  {  
    gameState = PLAYING;  
  }  
}
```

Finishing the
code by
rendering

```
function render()  
{  
  drawingSurface.clearRect(0, 0, canvas.width, canvas.height);  
  
  //Display the sprites  
  if(sprites.length !== 0)  
  {  
    for(var i = 0; i < sprites.length; i++)  
    {  
      var sprite = sprites[i];  
      if(sprite.visible)  
      {  
        drawingSurface.drawImage  
        (  
          image,  
          sprite.sourceX, sprite.sourceY,  
          sprite.sourceWidth, sprite.sourceHeight,  
          Math.floor(sprite.x), Math.floor(sprite.y),  
          sprite.width, sprite.height  
        );  
      }  
    }  
  }  
}
```

More Touch Example



touchMove to
move the
object



touchStart to
shoot the bullet

TouchMove Handlers

```
canvas.addEventListener("touchmove", touchmoveHandler, false);
```

```
function touchmoveHandler(event)
{
    //Find the touch point's x and y position.
    //Subtract the canvas's top and left offset
    touchX = event.targetTouches[0].pageX - canvas.offsetLeft;
    touchY = event.targetTouches[0].pageY - canvas.offsetTop;
    event.preventDefault();
}
```


TouchStart Handler

```
shootButton.addEventListener("touchstart", touchstartHandler, false);
```

```
function touchstartHandler(event)
{
    event.preventDefault();

    //Create a star sprite
    var star = Object.create(spriteObject);
    star.sourceX = 192;
    star.sourceWidth = 38;
    star.sourceHeight = 38;
    star.width = 38;
    star.height = 38;

    //Center it over the wand
    star.x = wand.centerX() - star.halfWidth();
    star.y = wand.centerY() - star.halfHeight();

    //Set its speed
    star.vx = Math.cos(angle) * 7;
    star.vy = Math.sin(angle) * 7;

    //Push the star into both the sprites and stars arrays
    sprites.push(star);
    stars.push(star);
}
```