Maths space

Codes for the game

Start page

**On the start page, I have coded for the start button to move to frame 2 for players to be directed to the tutorial page.**

**I have also coded for players to be directed to the frame in which they will be choosing which level to play**

 **By adding the two event listeners for a mouse event, the buttons coded should work once clicked.**

stop() ;

startButton.addEventListener (MouseEvent.CLICK, playAnimation);
function playAnimation(evnt: MouseEvent){
 gotoAndPlay (2);
}

stop();

Tutorial.addEventListener(MouseEvent.CLICK, toTutorial);
function toTutorial(e: MouseEvent){
 gotoAndStop(5);
}

Tutorial

**I have added a function here for a mouse event to create a button that will direct players back to the start page (once they’ve finished viewing the tutorial page). Where it says “gotoAndStop(1);” that is the frame of the start page in which the player will be directed.**

stop();

MMenu.addEventListener(MouseEvent.CLICK, movetoMain1);
function movetoMain1(e: MouseEvent){
 gotoAndStop(1);
}

MMenu.addEventListener(MouseEvent.CLICK, movetoMain1);

Level 1

import flash.events.Event;
stop ()

stage.addEventListener(KeyboardEvent.KEY\_DOWN,downHandler);

**The main code in this frame is the keyboard event “stage.addEventListener(KeyboardEvent.KEY\_DOWN,downHandler);”**

**The event listener will call upon the handler and trigger the event, carrying it out. So in this case once the up down left right keys on the keyboard are pressed the “shooter” that has been coded will move in those directions.**

**I have also declared a variable. A variable can be either a number or a letter. The variable is declared and is holding data which is reserved in the RAM, changing every time it comes out, so I have declared a variable for a bullet that I have created (using the name “newBullet” to identify it), and it is holding that bullet for when the player is going to use it.**

**Using arrays is like having a chest of draws. Firstly you would have to declare the array using a variable. What I have named mine is** HScores**. After the colon,** Array **is the data type I wish to use. As mentioned above, using arrays is like having a chest of draws, being able to open and close them when you want. By using .**push **I am requesting for the high scores of 20, 30, 40, 50, and 60 to show as output. With the .sort I can either choose to sort my arrays from descending order, or ascending order, my arrays are already sorted from ascending order, so there isn’t a need to sort it. The output that I will receive is from the for loop, for example, in my codes I am requesting for the highscores to show the first integer to the fifth integer, shown as:** i=0; i<5. **(the i++ means that the integer will be increased by 1). I then use trace to trace the output of the highscores so that it can show on the output screen when playing my game.**

**In the downHandler function I have coded the up, down, left and right arrow keys for the shooter to move.**

var scoreDisplay: Number =10;

//KEY\_UP //downHandler //upHandler

var bullet : Bullet = new Bullet();

var HScores: Array = new Array ();

HScores.push(20);
HScores.push(30);
HScores.push(40);
HScores.push(50);
HScores.push(60);

var i :int;

HScores.sort ();

for(i=0; i<5; i++){
trace(HScores[i]);
HScores[i] = 100;
}

function downHandler(e:KeyboardEvent){

switch (e.keyCode){

 case 37:// left
 shooter.x -=10;
 break;
 case 38: // up
 shooter.y -=10;
 break;
 case 39: // right
 shooter.x +=10;
 break;
 case 40: // down
 shooter.y +=10;
 break;

 case 32: // spacebar
 //shooter.x
 if(bullet != null){
 addChild(bullet);
 bullet.x=shooter.x +30;//5
 bullet.y=shooter.y+20; //100;
 addEventListener(Event.ENTER\_FRAME, flyBullet);
 }else {
 bullet = new Bullet();
 addChild(bullet);
 bullet.x=shooter.x +30;//50;
 bullet.y=shooter.y+20; //100;
 addEventListener(Event.ENTER\_FRAME, flyBullet);
 }

**For the shooter, I have added in my first if and else statement. In the if part of the statement, addChild insterts a bullet if the player has pressed the spacebar.
“bullet.x=shooter.x +30;//50;
bullet.y=shooter.y+20; //100;” This is where the bullet will be coming out of, the shooter.**

**The else statement is exactly like the if statement, if the spacebar is pressed, a bullet will fly out of the shooter.**

function flyBullet (evt: Event){
 bullet.x = bullet.x +10;
 //trace(bullet.x);
 checkHits();
 if(bullet.x > 550){
 if(bullet.parent != null){
 (bullet.parent).removeChild(bullet);
 bullet = null;
 removeEventListener(Event.ENTER\_FRAME, flyBullet);

**“checkHits();” If the function of another if statement below.**

**“removeEventListener(Event.ENTER\_FRAME, flyBullet);” removes the bullet from the frame.**

}

function checkHits(){
 if(bullet.hitTestObject(twenty4)){
 gotoAndStop(19);
 }else if (bullet.hitTestObject(twenty1)){
 gotoAndStop(20);
 }
}

**In this if and else statement, a function has been added for a “hitTestObject”, when another object hits another, and in this case, when the bullet hits the object “twenty4”, what will happen is that the player will be directed to another frame, and in the else part, when the bullet hits “twenty1” it will also be directed to another frame.**

}