

Rough Game Outline

Build Content (graphics / sounds):

- Character + behaviors
- Assets + behaviors
- Screens (each screen is in a different frame)
- Add Assets to each screen

Prepare / Name Assets on stage:

- Design and Layout in each frame (interface, scene, assets, character)
- Name MovieClip Instances controllable on stage in various frames (character, assets, etc.); all instance names must be unique even if on a different frame!

Add Scripts:

- 1 – Add character movements (left, right, up, down, etc.)
- 2 – Add hitTestObject script
- 3 – Move between screens

1 - Add Character Movements

Prevents playback head from leaving frame one automatically:

```
stop();
```

Prevents animation of mole from playing when game starts:

```
mole_mc.stop();
```

Variable that holds a number that can be changed for character movement:

```
var speed = 3;
```

TripSwitch - allows animation inside movieclip to complete by making conditional statement false until animation loop is completed:

```
var tripSwitch = false;
```

Mole's initial position on ground:

```
var groundPosition = mole_mc.y;
```

Here is the full keypress motion for mole:

```
stage.addEventListener(KeyboardEvent.KEY_DOWN, keyIsPressed);

function keyIsPressed(evt:KeyboardEvent):void {
    if (evt.keyCode == Keyboard.LEFT && !tripSwitch) {
        mole_mc.gotoAndStop(2);//walk
        tripSwitch = true;//turns if statement false
        stage.addEventListener(Event.ENTER_FRAME, moveLeft);
        //function will execute even if if statement is false
    } else if (evt.keyCode == Keyboard.RIGHT && !tripSwitch) {
        mole_mc.gotoAndStop(2);//walk
        tripSwitch = true;
        stage.addEventListener(Event.ENTER_FRAME, moveRight);
    } else if (evt.keyCode == Keyboard.UP && !tripSwitch) {
        mole_mc.gotoAndStop(3);//float
        stage.addEventListener(Event.ENTER_FRAME, float);
        tripSwitch = true;
    } else if (evt.keyCode == Keyboard.DOWN && !tripSwitch) {
        mole_mc.gotoAndStop(4);//hide
        tripSwitch = true;
        //keeps mole hidden instead of looping back around
        //through animation (repeat once)
        stage.addEventListener(Event.ENTER_FRAME, hideStatus);
    } else if (evt.keyCode == Keyboard.SPACE && !tripSwitch) {
        mole_mc.gotoAndStop(5);//monster
        tripSwitch = true;
        stage.addEventListener(Event.ENTER_FRAME, monsterStatus);
    }
}
```

Go back to default idle position when key is released:

```
stage.addEventListener(KeyboardEvent.KEY_UP, onKeyReleased);

function onKeyReleased(evt:KeyboardEvent):void {
    tripSwitch = false;//reset switch so you can press button
    //and repeat keypress
    mole_mc.gotoAndStop(1);//go back to blinking
    //just to be safe take out the position changes
    //from pressing left/right keys
    stage.removeEventListener(Event.ENTER_FRAME, moveRight);
    stage.removeEventListener(Event.ENTER_FRAME, moveLeft);
    stage.removeEventListener(Event.ENTER_FRAME, float);
    stage.addEventListener(Event.ENTER_FRAME, fallDown);
}
```

Keeps mole hidden - doesn't repeat through animation again:

```
function hideStatus(evt:Event):void {
    if (mole_mc.hide_mc.currentFrame==7) {
        mole_mc.hide_mc.stop();
    }
}
```

Keeps monster animation from cycling through on repeat:

```
function monsterStatus(evt:Event):void {
    if (mole_mc.monster_mc.currentFrame == 24) {
        mole_mc.monster_mc.stop();
    }
}
```

These two functions (moveRight / moveLeft) allow for mole to continue to move even if tripSwitch is switched. if the x and scaleX properties were directly added to above if conditional statement then all the movement would stop after initial key press:

Moves mole to right:

```
function moveRight(evt:Event):void {
    mole_mc.x+=speed;
    mole_mc.scaleX=1;
}
```

Moves mole to left:

```
function moveLeft(evt:Event):void {
    mole_mc.x-=speed;
    mole_mc.scaleX=-1;
}
```

Allows mole to leave ground:

```
function float(evt:Event):void {
    if (mole_mc.y>=groundPosition - 50) {
        mole_mc.y-=speed;
    }
}
```

Allows mole to fall back down:

```
function fallDown(evt:Event):void {
    if (mole_mc.y<groundPosition) {
        mole_mc.y+=speed;
    } else if (mole_mc.y>=groundPosition) {
        stage.removeEventListener(Event.ENTER_FRAME, fallDown);
    }
}
```

2 – Add hitTestObject script

Moves assets off stage and out of sight when character collides with them:

```
var getOffStage = -100;
```

Add this script inside function keyIsPressed:

```
if (currentFrame==1) {
    stage.addEventListener(Event.ENTER_FRAME,
checkInventoryFrame1);
} else if (currentFrame == 2) {
    stage.addEventListener(Event.ENTER_FRAME, checkInventoryFrame2);
    stage.removeEventListener(Event.ENTER_FRAME, checkInventoryFrame1);
}
```

checks hitTestObject on byte inventory in frame 1:

```
function checkInventoryFrame1(evt:Event):void {
    if (mole_mc.hitTestObject(byte_1)) {
        byte_1.x =getOffStage;
    } else if (mole_mc.hitTestObject(byte_2)) {
        byte_2.x =getOffStage;
    } else if (mole_mc.hitTestObject(byte_3)) {
        byte_3.x =getOffStage;
    } else if (mole_mc.hitTestObject(byte_4)) {
        byte_4.x =getOffStage;
    }
}
```

checks hitTestObject on byte inventory in frame 2:

```
function checkInventoryFrame2(evt:Event):void {
    if (mole_mc.hitTestObject(byte_5)) {
        byte_5.x =getOffStage;
    } else if (mole_mc.hitTestObject(byte_6)) {
        byte_6.x =getOffStage;
    } else if (mole_mc.hitTestObject(byte_7)) {
        byte_7.x =getOffStage;
    } else if (mole_mc.hitTestObject(byte_8)) {
        byte_8.x =-100;
    } else if (mole_mc.hitTestObject(byte_9)) {
        byte_9.x =getOffStage;
    }
}
```

3 – Move between screens

These scripts initialize certain starting parameters; add to top of script window

Extreme left and right side of stage - used for moving to next frame:

```
var stageRight = 560;  
var stageLeft = -10;
```

Add script within conditional statement for keypress LEFT:

```
stage.addEventListener(Event.ENTER_FRAME, endOfStage);
```

And RIGHT:

```
stage.addEventListener(Event.ENTER_FRAME, endOfStage);
```

Determines whether mole goes into next screen or not:

```
function endOfStage(evt:Event):void {  
    if (mole_mc.x>stageRight) {  
        nextFrame();  
        mole_mc.x=0;  
    } else if (mole_mc.x < stageLeft) {  
        //prevFrame();//use if you want to go back to  
        //previous frames...  
        mole_mc.x=0;  
    }  
}
```

Script In Full:

```
stop();  
mole_mc.stop();  
var speed = 3;  
var groundPosition = mole_mc.y;  
var tripSwitch = false;  
var getOffStage = -100;  
var stageRight = 560;  
var stageLeft = -10;
```

```

stage.addEventListener(KeyboardEvent.KEY_DOWN, keyIsPressed);
function keyIsPressed(evt:KeyboardEvent):void {
    if (currentFrame==1) {
        stage.addEventListener(Event.ENTER_FRAME, checkInventoryFrame1);
    } else if (currentFrame == 2) {
        stage.addEventListener(Event.ENTER_FRAME, checkInventoryFrame2);
        stage.removeEventListener(Event.ENTER_FRAME, checkInventoryFrame1);
    }
    if (evt.keyCode == Keyboard.LEFT && !tripSwitch) {
        mole_mc.gotoAndStop(2);
        tripSwitch = true;
        stage.addEventListener(Event.ENTER_FRAME, endOfStage);
        stage.addEventListener(Event.ENTER_FRAME, moveLeft);
    } else if (evt.keyCode == Keyboard.RIGHT && !tripSwitch) {
        mole_mc.gotoAndStop(2);
        tripSwitch = true;
        stage.addEventListener(Event.ENTER_FRAME, endOfStage);
        stage.addEventListener(Event.ENTER_FRAME, moveRight);
    } else if (evt.keyCode == Keyboard.UP && !tripSwitch) {
        mole_mc.gotoAndStop(3);
        stage.addEventListener(Event.ENTER_FRAME, float);
        tripSwitch = true;
    } else if (evt.keyCode == Keyboard.DOWN && !tripSwitch) {
        mole_mc.gotoAndStop(4);
        tripSwitch = true;
        stage.addEventListener(Event.ENTER_FRAME, hideStatus);
    } else if (evt.keyCode == Keyboard.SPACE && !tripSwitch) {
        mole_mc.gotoAndStop(5);
        tripSwitch = true;
        stage.addEventListener(Event.ENTER_FRAME, monsterStatus);
    }
}

stage.addEventListener(KeyboardEvent.KEY_UP, onKeyReleased);
function onKeyReleased(evt:KeyboardEvent):void {
    tripSwitch = false;
    mole_mc.gotoAndStop(1);
    stage.removeEventListener(Event.ENTER_FRAME, moveRight);
    stage.removeEventListener(Event.ENTER_FRAME, moveLeft);
    stage.removeEventListener(Event.ENTER_FRAME, float);
    stage.addEventListener(Event.ENTER_FRAME, fallDown);
}
function hideStatus(evt:Event):void {
    if (mole_mc.hide_mc.currentFrame==7) {
        mole_mc.hide_mc.stop();
    }
}
function monsterStatus(evt:Event):void {
    if (mole_mc.monster_mc.currentFrame == 24) {
        mole_mc.monster_mc.stop();
    }
}

function moveRight(evt:Event):void {
    mole_mc.x+=speed;
    mole_mc.scaleX=1;
}

```

```

function moveLeft(evt:Event):void {
    mole_mc.x-=speed;
    mole_mc.scaleX=-1;
}
function float(evt:Event):void {
    if (mole_mc.y>=groundPosition - 50) {
        mole_mc.y-=speed;
    }
}
function fallDown(evt:Event):void {
    if (mole_mc.y<groundPosition) {
        mole_mc.y+=speed;
    } else if (mole_mc.y>=groundPosition) {
        stage.removeEventListener(Event.ENTER_FRAME, fallDown);
    }
}
function endOfStage(evt:Event):void {
    if (mole_mc.x>stageRight) {
        nextFrame();
        mole_mc.x=0;
    } else if (mole_mc.x < stageLeft) {
        mole_mc.x=0;
    }
}
function checkInventoryFrame1(evt:Event):void {
    if (mole_mc.hitTestObject(byte_1)) {
        byte_1.x =getOffStage;
    } else if (mole_mc.hitTestObject(byte_2)) {
        byte_2.x =getOffStage;
    } else if (mole_mc.hitTestObject(byte_3)) {
        byte_3.x =getOffStage;
    } else if (mole_mc.hitTestObject(byte_4)) {
        byte_4.x =getOffStage;
    }
}
function checkInventoryFrame2(evt:Event):void {
    if (mole_mc.hitTestObject(byte_5)) {
        byte_5.x =getOffStage;
    } else if (mole_mc.hitTestObject(byte_6)) {
        byte_6.x =getOffStage;
    } else if (mole_mc.hitTestObject(byte_7)) {
        byte_7.x =getOffStage;
    } else if (mole_mc.hitTestObject(byte_8)) {
        byte_8.x =-100;
    } else if (mole_mc.hitTestObject(byte_9)) {
        byte_9.x =getOffStage;
    }
}

```