IST400/600 Scripting for Games **Final Exam Practice**

Name: ______________________________ **SUID:** ______________________________

Students are allowed to bring a “cheat sheet” to this exam. A “cheat sheet” is a piece of letter-size paper with anything written or printed on it. You can type it or hand-write it. You may photocopy your friend’s cheat sheet (assuming his/her permission) or download from somewhere. You can use the both size of the paper, if you want to, but you can bring only one, and are not allowed to share it with your classmates during the exam.

1. Expressions and statements (2 points x 5 = 10 points)

Following are a list of JavaScript code segments. (Assume `z` is a variable declared elsewhere.)

a.) `var x = new Array();` b.) `30 / 2` c.) `alert(“Hello!”);`

d.) `z == “winter”` e.) `false` f.) `var gBall = { x: 100, y: 125 };`

g.) `var y = [30, 45, 60];` h.) `var z;` i.) `function fun1() {alert(“test”);}`

1.1) Choose all the **string** expression(s). Say “None” if there’re none.

Your answer: ____

1.2) Choose all the **Boolean** expression(s). Say “None” if there’re none.

Your answer: ____

1.3) Choose all the **number** expression(s). Say “None” if there’re none.

Your answer: ____

1.4) Choose all the **function definitions(s)**. Say “None” if there’re none.

Your answer: ____

1.5) Choose all the **array declaration(s).** Say “None” if there’re none.

Your answer: ____

2. Variable Scope (3 points x 5 = 15 points)

Answer the questions based on the code below.

```
<script>
  var gGreeting = “Hello!”;
       var gFairwell = “Adios!”;
       function funA() {
     alert(gGreeting);
     alert(gFairwell);
  }
  function funB() {
     if (gGreeting == “Hello!”) {
       gFairwell = “Bye!”;
     } else if (gGreeting == “Hola!”) {
            gFairWell = “Adios!”;
          }
  }
  function funC() {
     funB();
     funA();
  }
</script>
<body onLoad=”fun2();”>
</body>
```

2.1) Name all the functions <u>defined</u> above.

Your answer: ______________________________

2.2) What will happen, when the code is loaded to a Web browser? (Circle one.)

a) Nothing happens.
b) The browser displays “Hello!” (and only “Hello!”)
c) The browser displays “Hello!” and “Bye!”
d) The browser displays “Hola!” (and only “Hola!”)
e) The browser displays “Hola!” and “Adios!”

2.3) What will happen, when `funC()` is called? (Circle one.)

a) Nothing happens.
b) The browser displays “Hello!” (and only “Hello!”)
c) The browser displays “Hello!” and “Bye!”
d) The browser displays “Hola!” (and only “Hola!”)
e) The browser displays “Hola!” and “Adios!”

2.4) Name all the **local variables** defined above. Say “None”, if there’re none.

Your answer: ______________________________

2.5) Name all the **global variables** defined above. Say “None”, if there’re none.

Your answer: ______________________________

3. Functions (3 points x 5 = 15 points)

3.1) Given the function definition below, what will be displayed, if `fun1(18);` is executed?

```
function fun1(age) {
    if (age < 18)
       alert(You can not vote yet.);
    else
       alert(You can vote now!);
}
```

Your answer:____

3.2) Given the function definition below, what will be displayed, if `fun2(433);` is executed?

```
function fun2 (time) {
    var myHour = Math.floor(time / 60);
    var myMin  = time % 60;
    alert(“It is “ + myHour + “:” + myMin + “ now.”);
}
```

Your answer:____

3.3) Given the code below, what will be displayed, if `fun3(myBirthday);` is executed?

```
var myBirthday = {year: 1973, month: “August”, day: “9th”};
function fun3 (birthday) {
  alert(“Your birthday is “ + month + “ “ + day);
}
```

Your answer:____

3.4) Given the function definition below, what will be displayed, if `fun4();` is executed?

```
function fun4 () {
    var myNumber = 100;
    for (var i = 0; i < 5; i++) {
        myNumber -= i;
    }
    alert(“myNumber is “ + myNumber);
}
```

Your answer:____

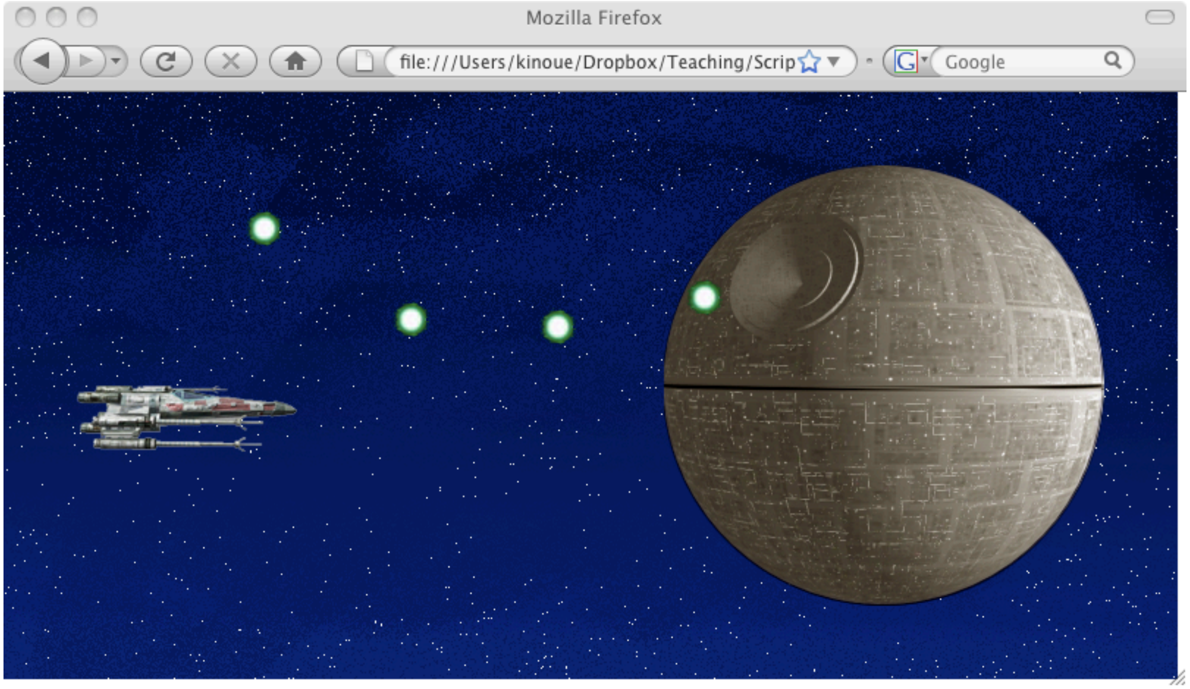
3.5) Following is an incomplete definition of a function that simulates tossing a coin, displaying one of the two different messages: “Head!” or “Tail!”, with 50% of chances. Fill in the blank and complete the definition.

```
function fun5 () {
  var coin = Math.random();

  if (________________________)
    alert(“Head!”);
  else
    alert(“Tail!”);
}
```

4. Game Scripting (5 points x 12 = 60 points)

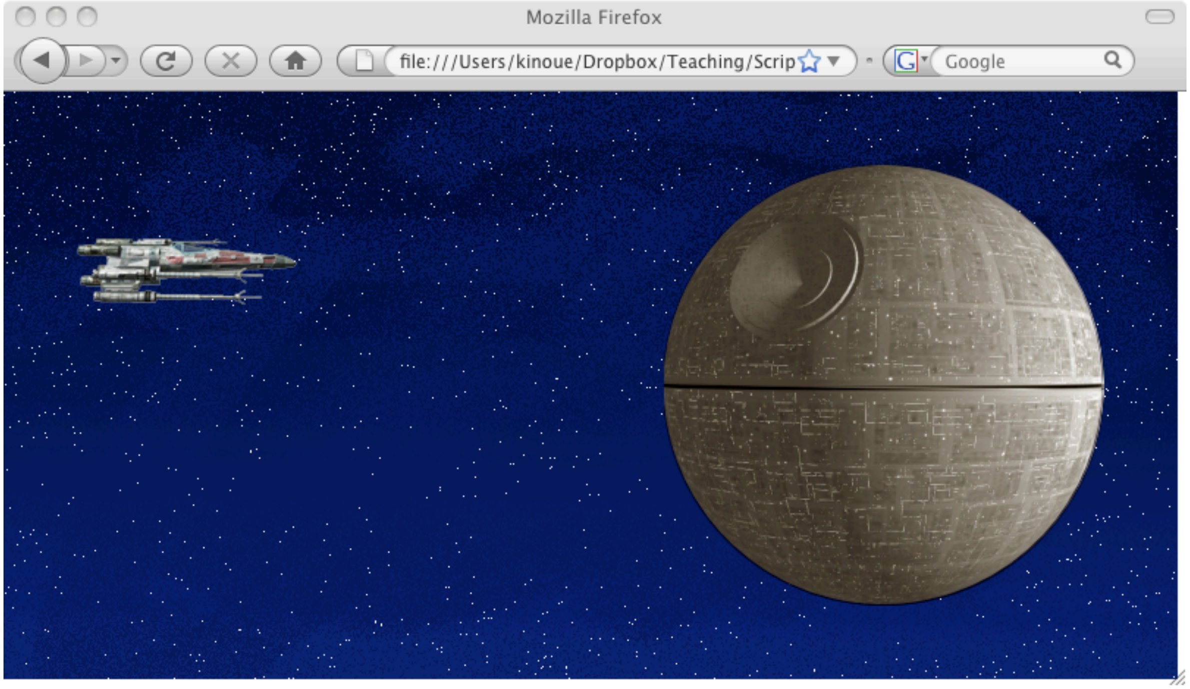
Let's assume you decided to create a game where the player would control the X-wing to fight against the Death Star. In this section, you will implement components of this game so that the Death Star will shoot energy balls towards the X-wing with a certain interval, which should looks like below.



You have four images prepared, each of which has a self-explanatory file name:

- `x_wing.png`
- `background.png`
- `death_star.png`
- `energy_ball.png`

So far, you finished displaying and positioning the elements for the game, which looks like below:



The current code look like the following:

```
<html>
<script>
var gGameState = new Object();
function initializeGame() {

  gGameState.intervalCount = 0;

  // Your answer for 4.1)

  var bg = document.getElementById('bg');
  bg.style.position = "absolute";
  bg.style.top = 0; bg.style.left = 0; bg.style.zIndex = 0;

  var sp = document.getElementById('sp_main');
  sp.style.position = "absolute";
  sp.style.top = 100;  sp.style.left = 50;  sp.style.zIndex = 2;

  var ds = document.getElementById('sp_ds');
  ds.style.position = "absolute";
  ds.style.top = 50;  ds.style.left = 450;  ds.style.zIndex = 2;
}
</script>
<body onLoad="initializeGame();">
  <img id="bg" src = "background.png"/>
  <img onClick="startGame();" id="sp_main" src="x_wing.png"/>
  <img id="sp_ds" src="death_star.png"/>
</body>
</html>
```

4.1) In the space above, insert codes that declare and initialize following properties:

- `gGameState.energyBallInterval`, which represents the interval that the Death Start shoots the energy ball. Initialize it with 25. (**(5points)**

- `gGameState.energyBallList`, which contains an array of objects that represents energy balls. Initialize it as an empty array. **(5points)**

- `gGameState.energyBallSpeed`, which represents the speed of the energy ball in the number of pixels. Initialize it with 10. **(5points)**

4.2) Provide a code segment that define a function `startGame()`, which does the following **(5 points x 2)**:

1) Display a message that says "The game is on!" **(5 points)**
2) Start a timer so that it will call a function `updateGame()`, every 200 milliseconds. Assign the timer to `gGameState.timer`. **(5 points)**

```
function updateGame()
{
  // your answer for 1)


  // your answer for 2)


}
```

4.3) Following is an incomplete code segment for defining `updateGame()`. Fill in the blank by providing codes that calls `shootEnergyBall()` every `gGameState.energyBallInterval` times that `updateGame()` is called. **(5 points)**

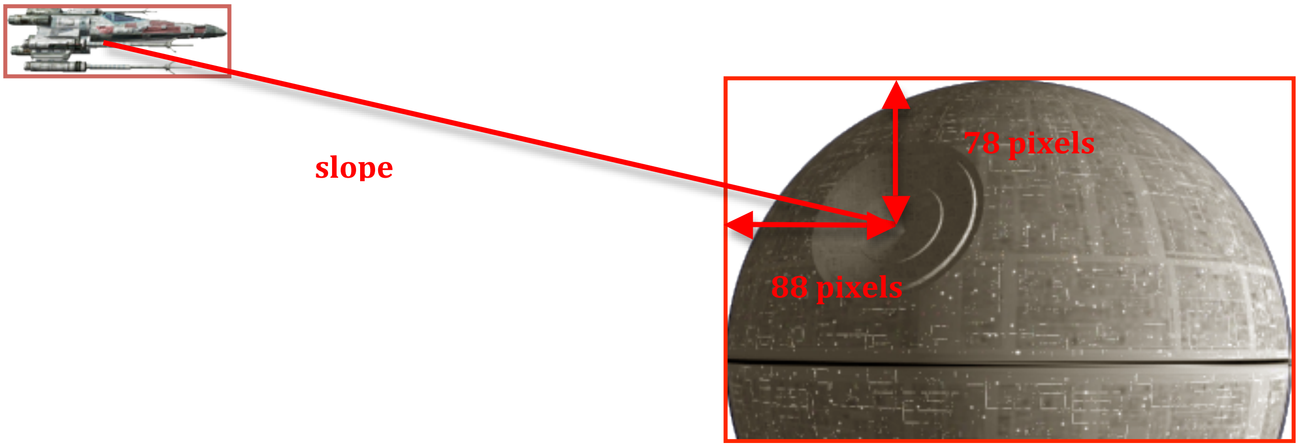
```
function updateGame()
{

  // your answer for 4.3)




   gGameState.intervalCount++;
   updateScoreBoard();
   updateEnergyBalls();
}
```

4.4) Now you are going to define a function, `shootEnergyBall()`, which creates an energy ball that will fly towards the direction of the X-wing. The actual flying part is taken care of by another function, alled `updateEnergyBalls()`, so what you need to do here is to position the energy ball, where the muzzle is located on the graphic, and set the direction of the energy ball, based on the slope of the line that is created by the middle point of the X-wing and the muzzle. The muzzle of the Death Star is located in the 78 pixels from the top image vertically, and 88 pixels from the left of the image horizontally.



Following is an incomplete code segment for defining `shootEnergyBall()`. Complete the definition by following the steps below **(5 points x 4)**:

1) Declare a variable called `mySlope`, and initialize it with the slope of the line between the muzzle and the center of the x-wing image.

2) Declare a variable called `newEnergyBall`, and initialize it as a new object with two properties:
   a. `id`, which should be initialized with the length of the `gGameState.energyBallList`.
   b. `slope`, which should be initialized with `mySlope`, which you just defined.

3) Add `newEnergyBall` to `gGameState.energyBallList`.

4) Position the image element, represented with the variable `eb`, right on the muzzle.

```
function shootEnergyBall()
{
  var sp = document.getElementById('sp_main');
  var ds = document.getElementById('sp_ds');

  // Your answer for 1)









  // Your answer for 2)


  // Your answer for 3)


  var eb = document.createElement('img');

  eb.setAttribute('id', 'eb' + newEnergyBall.id);
  eb.src = 'energy_ball.png';
  eb.style.position = "absolute";
  eb.style.zIndex = 3;

  // Your answer for 4)





  document.body.appendChild(eb);

}
```

4.5) Following is an incomplete code segment for defining `shootEnergyBall()`. Complete the definition by following the steps below **(5 points x 2)**:

1) Fill in the underlined blank to complete the construction of the for-loop, which visits the all the objects in the `gGameState.energyBallList` array.

2) Assign a value to the `ebElement.style.top` property, so that the energy ball will go towards the direction that was calculated earlier using the slope.

```
function updateEnergyBalls()
{

  for (_________________________________________________________) // 1)
  {
    var eb = gGameState.energyBallList[i];
    var ebElement = document.getElementById('eb' + eb.id);

    if (ebElement != null)
    {
     ebElement.style.left =
        parseInt(ebElement.style.left) - gGameState.energyBallSpeed;

     ebElement.style.top =


        ________________________________________________________  // 2)

     }
  }
}
```