BIRZEIT UNIVERSITY

Computer Science Department

Mobile Game Programming (COMP2351)
Second Semester 2017/2018

Instructor: Dr. Yousef A. Hassouneh Office Masri 322

Assignment 2: due date 12/05/2018

The Battle for Ram Aras

With device adaptive and touch event support

Overview

We will develop a small game titled The Battle for Ram Aras using JavaScript. This is a 2D space shooter game where the goal is to save the planet Ram Aras from invaders. This assignment, we will implement a laser-shooting battlecruiser.

Instructions

You will implement the game as JavaScript modules: the battlecruiser (`Battlecruiser.js`), the laser (`Laser.js`), and the enemy (`Enemy.js`). Also you MUST follow the book structure in implementing the game.

I have provided few sprites, for the rest you must search the web for them, for instance sprites for your enemies, you need at least three sprites for each type { normal, super, and subprime} .

Specifications and Controls

- ✓ The `Battlecruiser` sprite shall use the asset `battlecruiser.gif`
- ✓ The `Laser` sprite shall use the asset `laser.gif`
- ✓ Both sprite classes shall have a methods named `update, Initialize, draw, handleInput`
- ✓ You are free to use additional variables for each of the sprite classes.
- ✓ The controls for the battlecruiser:

    - LEFT ARROW: Moves the battlecruiser to the left of the screen (in the x-direction)
    - RIGHT ARROW: Moves the battlecruiser to the right of the screen (in the x-direction)
    - UP ARROW: Moves the battlecruiser up the screen (in the y-direction)
    - DOWN ARROW: Moves the battlecruiser down the screen (in the y-direction)
    - SPACE BAR: Fires a laser with sound. Important: the battlecruiser can fire multiple lasers.
- Define a type called `Enemy` that:
    - Load the asset
    - You need to create more than one instance based on Enemy class.
    - Have and maintain the following member variables {sprites, lives, power factor {it is 1 for normal, 1.5 super ,2 subprime}, speed, movement [an enemy can move either horizontal, vertical, diagonal or circle]
    - The velocity vector is calculated based on the current position and movement type.
    - Define a power as propriety which is calculated based on the number of lives and power factors.
    - Have a method named `update`
    - Method named `draw`
    - Use the asset `mutalisk.gif` for battlecruiser lives
    - Use the asset (change image to) `laser_explosion.gif` if an enemy is collided by a laser; the enemy is destroyed and disappear from the battlefield if its number of lives become ZERO, otherwise the number of lives decreased by 1.
- your game should implement collision detection
- If a laser is fired, the sound asset `laser.wav` shall be played.
- The points are calculated as follow each successful hit get 10 points * power factor
- Each the enemy live is taken if successfully hit by a laser
- If the battlecruiser collides with an enemy the game is over and the sound asset `death_explode.wav` is played.
- the battlecruiser and the Enemy should be stay inside the game area / battle filed.
- The game score shall be rendered near the upper-left corner of the screen.
- If the game is over, please give feedback to the player that the game is over.
- Upon game start, you should provide user instructions on the game area canvas
- Make your game adaptive to various devices, smart phone, tablet, and laptops with different screen.
- Make your game response to touch event, so the battlecruiser will be moved, so it will follow the movement of the finger. Also, the laser is fired when the finger touch the enemy.