

2D Adventure

Dr. Bradford A. Towle Jr.

Graphics

1/23/2019

1. Objectives

- A. To demonstrate a nominal understanding of 2D graphics

2. Equipment

- A. A laptop, notepad ++ (or some other text editor), a web-browser.

3. Lab Preparation

- A. Make sure you have reviewed the canvas lectures
- B. Make sure you have designed out how you want to go about coding this program

4. Lab Instruction

- A. You will make a simple JavaScript game that will break the canvas up into 64X64 pixel squares (you do not have to visualize this).
- B. You will use a 2D array to dynamically draw a simple maze consisting of bricks and trees.
 - a. The “bricks” and “trees” will be rendered through Context 2D commands. (Not a picture)
- C. There will be one player and at least three different types of monsters.
 - a. The player and monsters will be (64X64 pixel) images placed on the map.
- D. The player will push 1 of four buttons moving their character in that direction one square (You may also use press key callbacks if you wish).
- E. If the player runs into a monster the player will attack the monster (But not move to the square).
- F. After the player has moved all of the monsters will move, if the monster moves on the player it will attack the player (but not move to the square).
- G. Monsters and Players cannot move through trees or walls.
- H. There will be at least 2 potions on the map that will restore the player’s health if they move on to them. (The potion will then be destroyed after it is collected).

- I. If all the monsters have died then print “You Win” on the screen. If the player is killed, then you will print “Game Over”.
- J. Both the monsters and the hero will have the following properties:
 - HP – Health
 - MHP – Max health
 - ATK – Attack bonus
 - DEF – Defense bonus
 - X – (X coordinate on the grid, not the pixel value)
 - Y – (Y coordinate on the grid, not the pixel value)
 - PIC – The variable pointing to the HTML image for the appropriate character.
- K. Both the player and the monster will be rendered by images, they will also have a health bar next to them. (You pick the orientation).
- L. The images will be rendered on the appropriate grid based on the X and Y value.
 - a. Since both the hero and the monster use the same render function you may want to use inheritance.
- M. For combat you will follow the equation: $\text{Damage} = (\text{other.ATK} - \text{this.DEF} \text{ or } 1 \text{ whichever is higher}) * \text{random number } 1-6$.
 - a. Again this could be implemented in a parent class.
- N. You will have three monsters with varying difficulty and one hero. I will let you decide what the attributes should be. The game should be challenging but not impossible.

HINT: Remember to use OOC principles. You should have a main class that will have an update function (called whenever the user enters an input) and have it iterate through the player and all of the monsters.

NAME _____

5. Lab Rubric

	Perfect	Functional, but some logic errors or missing criteria	Attempted, but not functional	Minimal to no effort, did not attempt.
Draw maps and trees with 2d canvas functions	10	7	5	0
Have enemies move after the player moves, but they cannot move over trees	10	7	5	0
Be able to move the hero	10	7	5	0
Hero and monsters die if their HP reaches 0	10	7	5	0
Hero and Monsters can do damage	10	7	5	0
Game over and Game Win criteria work	5	3	1	0
Total	/55			

6. Lab Report Requirements

No report necessary for this lab.

However, there **is** a **PEER** Review required.