**Project PF1 – Bomberman**

**Information**

This is the team project for programming fundamentals.

Title: Bomberman

Team members: Chengyu Yu, Elena Hu

**Functionality**

This project aims to develop a 2D battle game based on the gameplay of Bomberman for MacOS keyboard. Players will place bombs in a map with randomly generated obstacle elements to destroy obstacles and try to eliminate opponents.

Map: The map is a 15x11 rectangle with 165 cells. The maps will be generated randomly. It consists of a random combination of three elements, one unit of one element occupies one cell: ‘fb’ fix position and indestructible blocks; ‘rb’ random position and destructible blocks; ‘ra’ random position aisle; ‘fa’ fix position starting areas.

- ‘fb’ fix position and indestructible blocks: players can neither pass through these blocks nor place bombs on them. Players cannot destroy them. The position is fixed and will remain unchanged, even after random map generation.

- ‘rb’ random position and destructible blocks: players cannot pass through these blocks nor place bombs on them. Players can destroy them. After they are destroyed, they become aisles. The position will change after the random map generation.

- ‘ra’ random position aisles: players can walk through the aisle. Players can place the bombs here. The position will change after the random map generation.

- ‘fa’ fix starting areas: there are two ‘fa’, one on the top-left corner, one on the bottom-right corner. Each of them occupies 4 cells. When the game starts, the players will appear on different ‘fa’. They are like aisles, but the position won’t change after random map generation.

Movement: player 1 moves with ‘up’, ‘down’, ‘left’ and ‘right’ keys and place the bombs with ‘return’ key; player 2 moves with ‘w’, ‘s’, ‘a’ and ‘d’ keys and place the bombs with ‘return’ key.

Non-exploded Bombs: Bomb will occupy one ‘ra’ cell. Players cannot pass through bombs or place another bomb on bombs. The bombs will explode 3 seconds after the player places it. Each player can have a maximum of ‘n’ bombs on the map, where ‘n’ represents the number of minutes that have passed since the start of the game.

Exploded Bombs: Exploded bombs will occupy a cross-shape area of 9 cells. Players can walk in this area. It lasts 1 second. All destructible blocks and players in this area during this period will be destroyed/defeated.

Game end conditions: 1) if a player is defeated, game over. The player defeated loses. 2) if after 5 minutes, no player is defeated, the game ends in a draw.

**Resources**

(require 2htdp/image)

(require 2htdp/universe)

**Data structures**

(define-struct game-state [map bomb player1 player2])

;map is one of the following:

; -- map1

; -- map2

; -- map3 (if we can)

; -- #false

; map1,map2,map3 are (vector(vector(char)))

; map1 = random of 11\*15 axiom (finished)

; map2 = easting egg of creator name shape (we can do)

; map3 = the biggest map that created by l-system arithametic (if time permit)

;before choosing a certain map, map = #false, render the starting picture/choosing picture

;player1,player2 is a sturcture

(define-struct player1 [position dead?])

(define-struct player2 [position dead?])

;dead is a Boolean

;means the state of the players

;bomb is one of the following:

(define-struct bomb [maximum list<position>])

;-- #false

;maximum is a Number which can changed

;represents the amounts of bomb that players could use

;position reprents the poistion of the elements

;position is not Posn, its a new defined structure

(define-struct position [row column])

;(define-struct position [Number Number])

;represents the position in the map

;examples

;(make-position 1 2)

;represents the position in the 1 row and 2 column of the map

;warning

;it can not be used directly, we need design a function help the structure position

;access the map

;#false means no bomb in the game now