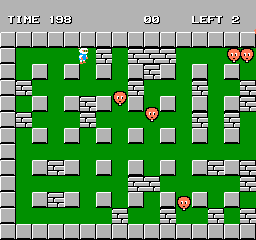
Detailed Game Specification:  
“Bomberman”

Course: COMP 2659, Winter 2019  
Instructor: Marc Schroeder  
  
Author(s): Mack Bautista  
Last Modified: 02/22/2024

# 1. General Game Overview

Bomberman is a 2D maze game, designed with the goal of destroying enemies present in the stage within an allotted time. The player will control Bomberman, a playable character that can be moved multidimensional. Throughout the stage, there will be bricks that will prevent the player from being able to destroy the enemies. To proceed with the stage, the player has the given ability to place bombs. These bombs are essential to opening paths and for completing the stage. Destroying the bricks will give the player power-ups to enhance mobility, the range of bombs, or increase retries as lives. When the player does not complete the stage within the limited time, the game will result in a game over. The default lives will be set to three. The lives, score, and timer are displayed on the top screen.



(from <https://www.mobygames.com/game/10806/bomberman/screenshots/nes/52733/>)

# 2. Game Play Details for Core 1-Player Version

[Second sample screenshot goes somewhere in section 2. Include additional screen shots and diagrams throughout section 2 as necessary.]

## Objectives and Rules

The player will begin at the centre of the screen. Depending on the stage or difficulty, the entire stage will have fewer or more bricks. Within a 3x3 block enclosure, there must be enough space given to be able to dodge any placed bombs to proceed with the stage. The bomb’s radius will be default to a three block space and should only affect the objects that it detects. The detection is dependent on whether or not the object is: player, enemy, bricks, or indestructible.

The objective of the game is to destroy the enemies present in the stage within a given time or game session, i.e each time the player begins the game by clicking 1-player or 2-players. When the life counter reaches less than 0, the session ends and will result in the player being restarted to the main menu. The default time to complete a stage is 240 seconds.

The player may proceed to the next stage once all enemies have been defeated within the given time and have enough remaining lives. It is not necessary to destroy all the bricks within the stage.

# Bomberman Movement and Collisions

Bomberman is the player’s controlled character. Arrow keys will allow the player to move horizontally or vertically if there is no collision detected. The arrows on the arrow keys will depend on the direction the player wants to move. Pressing the left arrow key will move Bomberman to the left, and holding any key such as the left arrow key will continuously make Bomberman move. If the following key is released, it will also release Bomberman’s movement and remain stationary. When the player collides with an oncoming enemy or bomb explosion will it result in a life deduction. When losing a life, the player will be given a two second invulnerability phase and have their position resetted.

# Enemy Movement and Collisions

The enemy movements are randomised. Its path is determined on whether or not an object is within its area. [The method for calculating enemy pathing is still yet to be determined.] As such when the enemy roams around the border of the stage, the directions are only enabled to what is available. Enemy velocity will have a default value. When the enemy comes into contact with a bomb explosion, i.e the bomb object has been updated must the enemy be removed from the stage. Upon colliding with the player, the enemy movements are continuous, only the player’s location must be updated to the starting position.

# Bomb and Brick Collisions

When the player chooses to place a bomb to their current position, there are two phases at which the bomb acts. A solid bomb is first placed in the current position where it can neither be moved, or collided with. After a few ticks, the bomb will detonate in a default three block range. Any object that is detected within its range is either removed or restored.

## Objects

|  |  |  |  |
| --- | --- | --- | --- |
| Object or Object Type Name | Properties | Behaviours | Graphical Image |
| Bomberman | * Position (integer pair: x, y) * Horizontal direction (integer) * Vertical direction (integer) * Size (constant integer pair, height width) * Horizontal velocity (integer) * Vertical velocity (integer) | * Move * Collide * Place bomb |  |
|  |  |  |  |
| Bomb | * Position (integer pair: x, y) * Size (integer pair: height, width) * Internal timer | * Collide * Update |  |
| Enemy | * Position (integer pair: x, y) * Horizontal direction (integer) * Vertical direction (integer) * Size (constant integer pair: height, width) * Horizontal velocity (integer) * Vertical velocity (integer) | * Move * Collide | [Insert image.] |
| Score | * Position (constant integer pair: fixed x, y) * Size (constant integer pair: height, width) * Value (integer) | * Update | [Insert image.] |
| Timer | * Position (constant integer pair: fixed x, y) * Size (constant integer pair: height, width) * Value (integer) | * Update | [Insert image.] |
| Life Counter | * Position (constant integer pair: fixed x, y) * Size (constant integer pair: height, width) * Value (integer) | * Update | [Insert image.] |

To make all bitmaps easier to scale, all sizes have been set to a 32x32 bitmap.

[Note: for some games a particular object may have >1 associated image, depending on its current state (e.g. Mario walking left vs. Mario walking right).]

## Physics

Both player and the enemies will have acceleration and velocity. Acceleration will occur after colliding with a brick (for the enemies) and when they stop (player after letting go of the input keys). Velocity to make calculations easier, will be constant.

Collision detection is a must and is determined when each object have their hitboxes overlapped.

## Asynchronous (Input) Events

[Note: keyboard input is required of all games. Additional mouse input is optional for core game play.]

|  |  |  |
| --- | --- | --- |
| Event Name | Triggering Input Event | Description |
| Move up request | Up arrow key is depressed | Moves the player vertically at y-position > 0. |
| Move down request | Down arrow key is depressed | Moves the player vertically at y-position < 0. |
| Move left request | Left arrow key is depressed | Moves the player horizontally at x-position < 0. |
| Move right request | Right arrow key is depressed | Moves the player horizontally at x-position > 0. |
| Place bomb request | Spacebar is pressed | Places a bomb object in the same position as the player. |
| Quit | Esc key is pressed | Exits out of application. |
|  |  |  |
|  |  |  |

## Synchronous (Timed) Events

[Note: on the Atari ST, one easy option will be to make use of a 70 Hz timer (i.e. 70 ticks per second). So, it may be easiest to base timed events on multiples of 1/70th of a second.]

[Note: the trigger for a synchronous event is typically based on a clock. But some synchronous events are also triggered by the occurrence of other synchronous events. E.g. a ship may move forward 1 pixel every 1/10th of a second, but it might as a result collide with an obstacle – these are two distinct events, one triggered directly by clock ticks and the other conditionally triggered by the original movement.]

|  |  |  |
| --- | --- | --- |
| Event Name | Trigger Timing | Description |
| Move player | [To be determined.] | [To be determined.] |
| Move enemy | [To be determined.] | [To be determined.] |
| Explode bomb | [To be determined.] | [To be determined.] |
| Countdown timer | [To be determined.] | [To be determined.] |
|  |  |  |
|  |  |  |
|  |  |  |

## Condition-Based (Cascaded) Events

[Note: some events may trigger other events, conditionally. This is called event “cascading”. The triggering event(s) may themselves be asynchronous, synchronous, or other condition-based events.]

|  |  |  |
| --- | --- | --- |
| Event Name | Triggering Condition | Description |
| Player - contact collision | When the player encounters any enemy, or the bomb explosion. | Position resets when colliding with an enemy or bomb. Position does not change if the player hits a brick, or indestructible object. |
| Enemy - contact collision | When the enemy encounters the player, or bomb explosion. | Leaves the current stage if it collides with a bomb explosion or continues its pathing when it collides with a player. |
| Bomb explosion - collision | When the bomb encounters the player, enemy, or brick. | Colliding with the hit box of a brick, destroys the brick, and kills an enemy or player. |
| Win condition | If the player eliminates all enemies’ objects without running out of lives, or is within the time limit. | If the player’s number of lives is equal to the amount of game over flag is set to true, and the timer is not equal to 0. |
|  |  |  |
|  |  |  |
|  |  |  |

## Hypothetical Gaming Session

Bomberman will begin at the centre. They have 240 seconds to destroy all enemies that are present within the stage. For the player to access these enemies, they must open the path by blowing up the bricks that are between them and the enemies. The border of the stage will be indestructible. Should the players be unable to complete the objective within the time limit, it will result in a life deduction. A life deduction will also occur if the player collides with an enemy or the bomb explosion. Losing all three lives results in a game over, i.e. the player is returned to the splash screen and must initiate a new game session. Completing the objective proceeds with the next stage where the maze will be different.

# 3. Game Play Details for Core 2-Player Version

The 2-player mode is similar to the 1-player mode except that two controllable players are present on the stage. Both players will begin at the centre except the 3x3 will be expanded into a 4x4 area for manoeuvrability.

[2-player sample screenshot goes here.]

# 4. Sound Effects

|  |  |  |
| --- | --- | --- |
| Sound Effect Name | Brief Description | Event which Triggers Playback |
| Bomb explosion | “Boom” sound | Bomb updates after a few ticks |
| Death sound | “Ping” sound | Player or enemy collides with a bomb explosion/with each other. |
|  |  |  |
|  |  |  |

[Optional: brief description of background music. Music will be required by assignment 3.]

# 5. Additional Features (Time Permitting)

[Add descriptions here.]