Period 2
Zi Jun Deng, Qinwen Zheng
NotCsNerds
Bomb It 9

Brief description of our game:
We are making a two player game of the game *Bomb It* where players drop bombs, destroy walls, and get power ups to try to defeat each other.

List of functionalities:

- Map
    - The 2D array for map only includes 0, 1, 2 as its elements, which represent an empty block, a breakable wall, and an unbreakable wall, respectively. The property of this game is that after the player uses the bomb to break the "breakable walls", they will turn into either a space block or some power up randomly. We already have the code there set up for the random power up that the breaking up of the walls is going to create.
- Players
    - Players each get assigned to PVector position and array position, so it is easier for us to do coding since we need to track their positions very frequently. Each player has its position assigned, and it is going to move in the map following the rules that are created in the moving method.
- Key pressed for movement of players
    - When keys "w", "a", "s", "d" are pressed, Player1 is going to move in the map as long as its destination is an empty space. We have a similar setup using RIGHT, LEFT, UP, DOWN for the movement of the second player. When these keys are pressed, they are going to check if its destination is an empty block or had been taken by a type of wall. If it is taken by a wall, then the player is not going to move to the block whos the wall is on, it will stay on its current position.
- Walls for boundaries
    - The walls are either breakable walls or permanent (unbreakable) walls. The unbreakable walls are there as a part of the map features to limit the movement of the players in the game, giving more chances in terms of the strategies that the players are able to choose. The breakable walls are the walls which are going to produce Power Ups in the map so the players can pick up these power ups to use them against the enemy.
- Created bomb class
    - We created a bomb class which currently has methods like explode(), which is going to check the nearby blocks which fall into the effective area which the bomb has an effect on.

It has a variable which keeps track of its x and y coordinates positions so we are able to do the checking of its nearby blocks easier.

- Bombing and decrease the lives of players
    - When a bomb is placed in the map, it will become a block which looks like a “breakable wall” but it is going to block the movement of players. It will also start a countdown, after the player who placed that bomb walked for five steps, it is going to apply explode() and explode based on its corresponding power up, which is “bombPower#”, representing the number of nearby blocks that it is going to have an effect on. Once a “breakable wall” is broken, a randomizing process is going to be executed so it will generate one of the power up blocks, or nothing (an empty space) on its original position.
- A tracker of different power ups.
    - There is a sidebar at the right of the main game display, which will show the three main attributes of each player (number of lives, bomb power, number of lives), so the player is able to know the current status and information of the character that they are having control on, and play this game better. Different power ups are shown with their corresponding images so players can see which one is which. So it is easier for them to know what power ups they just picked up.
- The players are able to interact with the Power Ups now
    - When a player uses a bomb to destroy a wall and have a power up left on its original position. The player can walk to the position of where the power up is and “pick up” this power up to make themselves “stronger.” This change will be reflected on the tracker immediately after the player picks it up.
- Display of the bomb exploded area (where the explosion is)
    - When the bomb explodes after 5 movements of the player who places it, the bomb will explode and have a “suddenly” existed black shadow which indicates the blocks that are affected by the bomb.
- A victory status
    - “Live” is one of the attributes of a player in this game. When one of the players has “0 lives” in this game, then that player loses this game and also meaning that their opponent wins this match. The game will be over and have a victory scene displayed. There is a trophy displayed on the screen to show the reward to players.
- Rematch
    - P is the key for a rematch in our game. If you press it, the game will be reset. There is no record for the game score in this game, so each game is a brand new game and you should put your 100% effort into it!

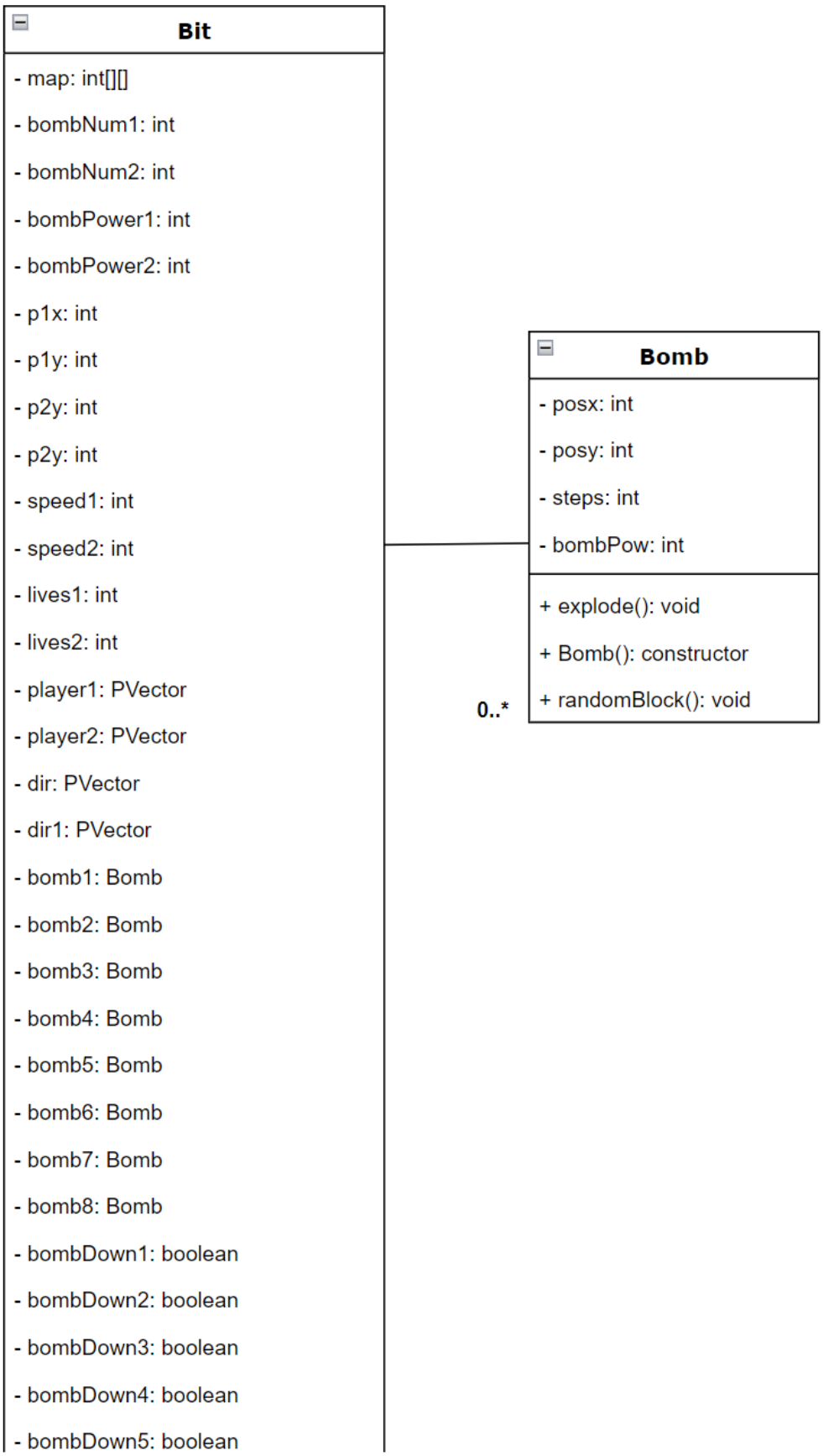
Log of tasks done by each person:

Zi Jun Deng:

1) Created the features of the players
2) Key pressed for movement of players
3) Created bomb class
    a) Constructor
    b) Made multiple bombs
4) Rematch button (“p”)
5) A victory status and visual screen
6) Display of the bomb exploded area (where the explosion is)
7) A tracker of different power ups. (Contributed by both)
    a) Updated trackers of power ups
8) Implemented images of players moving
9) Provided all images
    a) Used photoshop to make images
10) Made UML diagram

Qinwen Zheng:

1) Write up most parts of this document.
2) Create the map
    a) Modified based on a real game map
3) Walls for boundaries
4) Bomb Class
    a) Random Block generator
    b) Explode - bombing and decrease the lives of players
5) A tracker of different power ups. (Contributed by both)
6) Implemented map and powerup images
    a) Resized images to fit game grid
7) Provided steps mechanics to exploding bombs (math!)
8) Created status sidebar
    a) Added colors
    b) Created text


Bit
- map: int[][]
- bombNum1: int
- bombNum2: int
- bombPower1: int
- bombPower2: int
- p1x: int
- p1y: int
- p2y: int
- p2y: int
- speed1: int
- speed2: int
- lives1: int
- lives2: int
- player1: PVector
- player2: PVector
- dir: PVector
- dir1: PVector
- bomb1: Bomb
- bomb2: Bomb
- bomb3: Bomb
- bomb4: Bomb
- bomb5: Bomb
- bomb6: Bomb
- bomb7: Bomb
- bomb8: Bomb
- bombDown1: boolean
- bombDown2: boolean
- bombDown3: boolean
- bombDown4: boolean
- bombDown5: boolean
Bomb
- posx: int
- posy: int
- steps: int
- bombPow: int
+ explode(): void
+ Bomb(): constructor
+ randomBlock(): void
0..*


- bombDown6: boolean
- bombDown7: boolean
- bombDown8: boolean
- breakableWallORgiftimg: PImage
- wallimg: PImage
- powerUPimg: PImage
- morebombimg: PImage
- liveAddimg: PImage
- bombimg: PImage
- player1Up: PImage
- player1Down: PImage
- player1Left: PImage
- player1Right: PImage
- player2Up: PImage
- player2Down: PImage
- player2Left: PImage
- player2Right: PImage
- trophy: PImage
- player1img: PImage
- player2img: PImage
+ setup(): void
+ draw(): void
+ grid(): void
+ drawPlayer(): void
+ keyPressed(): void

To play this game, you need to understand the three important attributes: the number of bombs, the power of bombs (damage), and the number of lives. These attributes are displayed on the right part of the screen with descriptions.

The main goal of the game is to defeat your opponent by reducing their number of lives to zero using bombs. Each player has a certain limit on the number of bombs they can have on the map. This limit can be increased by collecting the power-up for "number of bombs." However, there is a cap of four bombs per player in the game at any given time.

All bombs will explode after you move for five steps. The explosion area of a bomb covers the nearby horizontal and vertical blocks. The extent of this damage is determined by the "power of bombs" attribute. For example, if the power of bombs is 2, the bomb will damage blocks in a 2-block radius in the vertical and horizontal directions.

The number of lives represents how many chances you have for your character. When you get hit by a bomb, you lose one life. If you step directly on top of a bomb, you will lose four lives at once. If your number of lives reaches zero, your opponent wins the game.

For Player 1, the moving keys are A, S, D, and W. The "put a bomb" key is E.

For Player 2, the moving keys are UP, DOWN, LEFT, and RIGHT. The "put a bomb" key is M.

In the game, there are gift boxes scattered around the map, acting as temporary walls. Players cannot pass through these gift boxes[1], so you need to use bombs to destroy them and access the power-ups hidden inside. When a gift box explodes, there is a chance for a power-up to appear, although sometimes the gift box may contain nothing. Power-ups give you advantages in the game, and both players have an equal chance of obtaining them. Be quick to collect power-ups to strengthen yourself and prevent your opponent from getting them.

Use strategic thinking to place bombs effectively and pick up power-ups to gain an advantage over your opponent. The game ends when one player's number of lives reaches zero or below. The winning player will be displayed with a big trophy on the screen[2]. You can restart the game by pressing the "P" key.

---

[1] You obviously cannot because they are just here.

[2] So you can be happy, or sad.

Image Citations

- Trophy image - https://www.freepik.com/free-photos-vectors/trophy-pixel-art
- More bomb - https://www.vectorstock.com/royalty-free-vector/black-bomb-pixel-art-colorful-vector-21005227
- Plus image - https://www.pngwing.com/en/free-png-nxxpt
- Photoshop tool to make more bomb image - photopea.com
- Gift - https://www.freepik.com/free-photos-vectors/gift-pixel
- Wall - https://www.freepik.com/premium-vector/tile-stone-brick-porcelain-stoneware-marble-roof-tiles-vector-retro-8bit-pixel-art-game-surface-pattern-background-8-bit-pixel-cubic-brickwork-rock-gravel-concrete-wall-blocks_25539045.htm
- Bomb - https://stock.adobe.com/images/pixel-bomb-with-burning-wick-icon-black-dynamite-weapon-ready-to-explode-symbol-of-explosive-discounts-and-gaming-8bit-graphics-powerful-vintage-vector-grenade/462307030
- Heart/lives - https://opengameart.org/content/heart-pixel-art
- Explosion/Bomb power up - https://www.istockphoto.com/photos/pixel-explosion