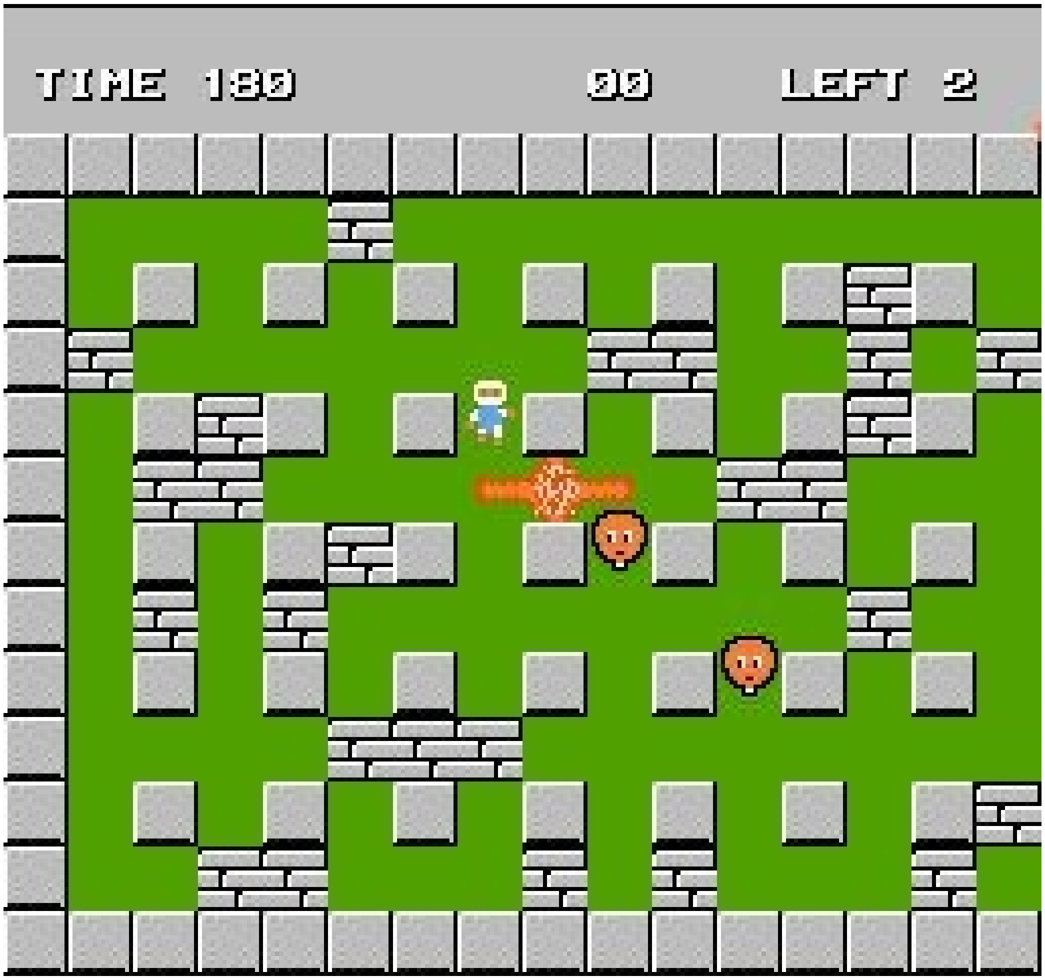
Bomberman

We are building a variant of the classic Bomberman, you might have played in your childhood. Remember this?



Link for the actual game(to play and understand) - http://www.retrogames.cz/play_085-NES.php?language=EN

Introduction -

You have to write a python program that simulates a basic version of Bomberman with different set of rules. You are responsible for destroying enemies.

As a template your code might contain the following classes corresponding to respective objects in the game:

1. Board
2. Person
3. Enemy
4. Bomberman
5. Bomb
6. Walls
7. Bricks

You have to implement it using OOP Principles [Inheritance, Polymorphism, Encapsulation and Modularity are must] as an example, Bomberman and enemy classes may inherit from Person class.

Rather than writing long methods, try to break up the functionality into more short methods, so that when you create subclasses, it will be easier to override specific parts of the functionality of the class. The better code you write, more marks you get.

Use of Pygame or ncurses (any function from curses library) is not allowed.

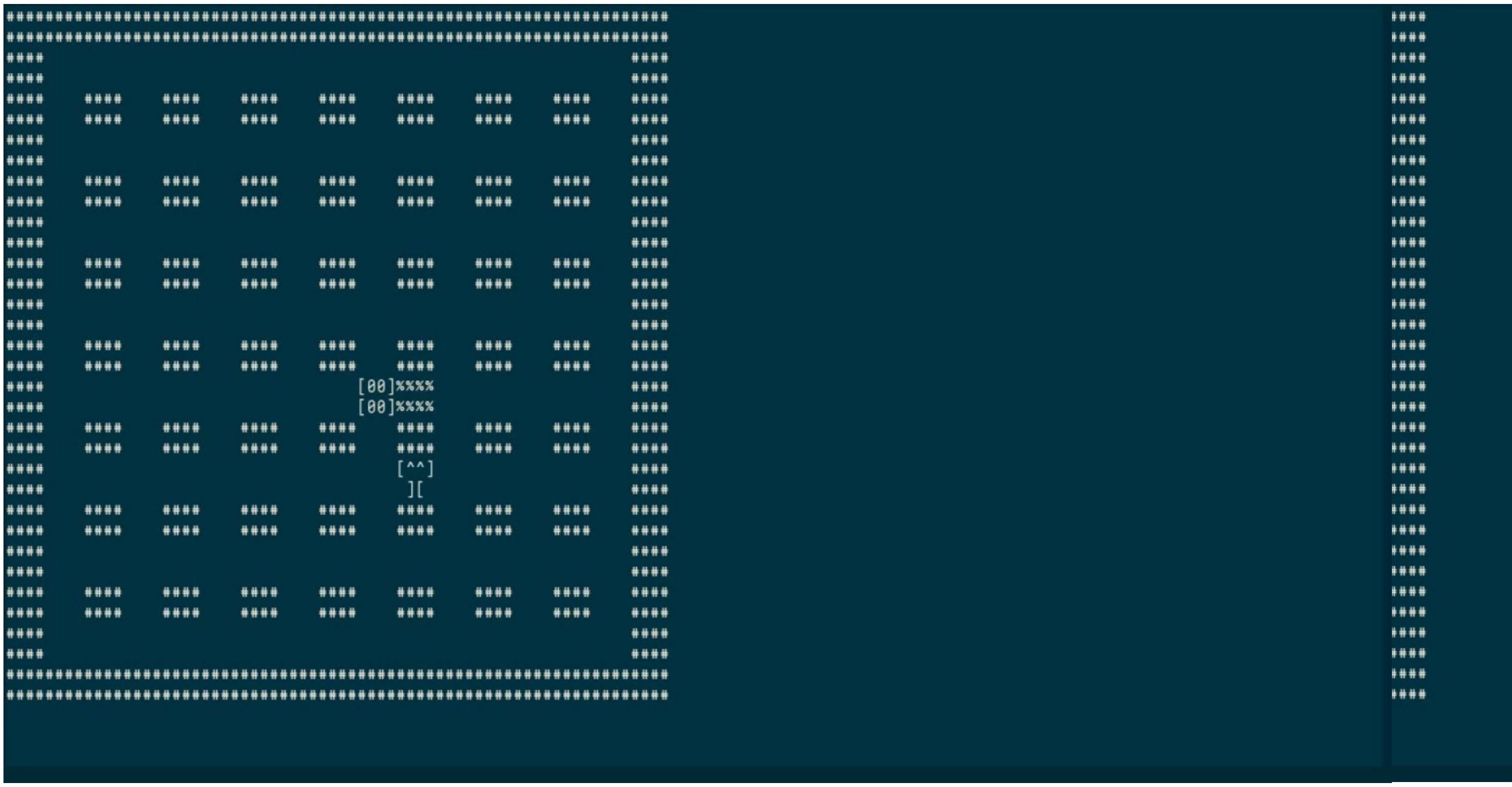
Rules

Symbols for the game are:

Walls	X
Bricks	/
Bomberman	B
Enemy	E

explosion	e
Bombs	O

Preferred board size - 40 X 80
Preferred size of each object - 2 X 4 (as shown in image below)



```
####################################################################
####################################################################
####                                                            ####
####                                                            ####
####    ####    ####    ####    ####    ####    ####    ####    ####
####    ####    ####    ####    ####    ####    ####    ####    ####
####                                                            ####
####                                                            ####
####    ####    ####    ####    ####    ####    ####    ####    ####
####    ####    ####    ####    ####    ####    ####    ####    ####
####                                                            ####
####                                                            ####
####    ####    ####    ####    ####    ####    ####    ####    ####
####    ####    ####    ####    ####    ####    ####    ####    ####
####                                                            ####
####                                                            ####
####    ####    ####    ####    ####    ####    ####    ####    ####
####    ####    ####    ####    ####    ####    ####    ####    ####
####                                [00]%%%%                    ####
####                                [00]%%%%                    ####
####    ####    ####    ####    ####    ####    ####    ####    ####
####    ####    ####    ####    ####    ####    ####    ####    ####
####                                    [^^]                    ####
####                                     ][                     ####
####    ####    ####    ####    ####    ####    ####    ####    ####
####    ####    ####    ####    ####    ####    ####    ####    ####
####                                                            ####
####                                                            ####
####    ####    ####    ####    ####    ####    ####    ####    ####
####    ####    ####    ####    ####    ####    ####    ####    ####
####                                                            ####
####                                                            ####
####################################################################
####################################################################
```

Game details -

- Board is bounded by ‘walls’ (denoted by ‘#’ in the image) which cannot be destroyed by the bomb (denoted by 2X4 object of ‘0’ as shown in the image). All the obstacles and enemies should be inside the board area. Obstacles might contain walls or ‘bricks’ (denoted by 2X4 object of ‘%’ in the image) - counterpart of walls which can be destroyed by the bombs. The board consists a grid of walls as shown in the image.

- Bomberman(denoted by‘[^^]][’ in image) spawns at top left of the board and can be moved around by controls defined by user. It cannot move to the positions occupied by any obstacle. It loses a life whenever it collides with an enemy and scores 100 points whenever an enemy is destroyed. It also scores 20 points for destroying any brick.

  Basic Controls -

Move left	a
Move right	d
Move up	w
Move down	s
Drop bomb	b

- Enemies can spawn anywhere on the board where there are no obstacles and move around in a pseudo-random motion. They get destroyed when they get caught in an explosion.
- A bomb can be dropped anywhere by the bomberman at the position where it is currently placed and it explodes after 3 frames of being dropped.
  The explosion of the bomb(denoted by ‘^’ in the image) will spread to a total of 4 positions (not including the bomb position and not occupied by any wall) **evenly around the bomb**.
  * Each position implies a 2X4 area on the board.
- Score and the number of lives left should be displayed below the board.

Bonus -

- The bomb should display the number of frames left for explosion instead of ‘O’. For example if one second is left then

```
[1111]   =>    [0000]  => explosion
[1111]     [0000]
```

- Implementation of extra levels (at least 3 levels should be implemented) for bonus.
- Different objects should have different coloured symbols. For example - enemy can be represented by an ‘E’
- Different power ups for bomberman which will appear by pressing the key ‘p’ on any valid position on the board and will remain there for a limited time if not collected. For example speed up, larger explosion, immortality, wallpass(ability to go through walls).
- Different kind of enemies which might move with different speeds or might require multiple explosions to be destroyed.

Note :

- **Plagiarism** will be strictly dealt with.
- **OOP Principles** should be strictly followed.
- End the game when all lives are over or User quits by pressing ‘q’.
- You can use any symbols which are not already assigned for any of the bonus items. All the details along with controls for your game should be submitted with the code in a README file.
- It is your game, make it as creative and fun as you can.

Evaluation Criteria -

Inheritance	15
Polymorphism	10
Modularity	10
Encapsulation	15

Functionality of Game	50
Bonus	25

References

- https://github.com/PiGames/Bomberman
- https://github.com/bloody-orange/BombermanASCII
- https://github.com/archive-/bomberman
- https://bitbucket.org/lbesson/mpri-bomberman/src