DASS Assignment 2

Submitted by N.V.Sree Harsha Reddy

Rollno: 2019101040

Instruction to play Ball Brick game

- At first run the game in terminal

```
python3 main.py
```

- After that

```
Press 'f' to free the ball from paddle when started or when the paddle
has Grab powerup
Press 'a' to move the paddle left
Press 'd' to move the paddle right
Press 'e' to exit the game
```

- Collect powerups which are dropped when certain brick breaks to score more in the game the powerup letters stands for:

```
E stands for Extend the size of paddle
S stands for Shrink the size of paddle
M stands for Ball Multiplier
F stands for increase the speed of the ball(s)
T stands for thru ball which is when this ball collides with brick it
breaks irrespective of strength and not deviated from path
G stands for paddle grab when the ball collides the paddle it sticks
to paddle and press 'f' to release ball
```

- Types of Bricks: different colors for different strength of bricks

```
GREEN - Brick with strength 1
YELLOW - Brick with strength 2
BLUE - Brick with strength 3
MAGENTA - Brick with strength 4
RED - - Brick with unbreakable strength except when it is adjacent to
a exploding brick or collided  by Thru ball powerup then this behaves
as a brick with strength 5
BONUS:
Blinking with CYAN and WHITE - This is a exploding brick when collides
```

```
it breaks all the bricks adjacent to it irrespective of strength of
the brick if it has an another exploding brick then becomes chain
reaction and explodes until the adjacent of those bricks do not have
exploding bricks
```

- If ball collides paddle its x velocity will change according to where it hits the paddle from the centre of the paddle
- The ball reflects when it is collided with brick or paddle or frame(execpt bottom) and not with some special powerups like grab , thru ball etc
- If the ball collides bottom the one life will be gone and the ball comes on to the paddle
- If ball looses three lives then You lost the game

OOP concepts followed in code

- Inheritance
  - I have a main class named `GameObject` and `Ball,Paddle,Brick,PowerUp`as child classes as they have some properties in common like x, y, xvel, yvel ,xlength, ylength i.e is these are inherited form `GameObject` to these `Ball,Paddle,Brick,PowerUp` classes
- Polymorphism
  - I have a `move` function in `GameObject` its a basic move, and I have `move` function in `Ball,Paddle,Brick,PowerUp` classes. this child's `move` function overrides the functionality of parent's `move` function
- Encapsulation
  - All my classes in different files
  - `game.py` has `Game` class which prints and updates the array which is gonna show on termianl screen
  - `gameobject.py` has `GameObject` class which is a parent class for all game objects
  - `ball.py` has `Ball` class which is inherited class form `GameObject`
  - `paddle.py` has `Paddle` class which is inherited class form `GameObject`
  - `brick.py` has `Brick` class which is inherited class form `GameObject`
  - `powerup.py` has `PowerUp` class which is inherited class form `GameObject`
- Abstraction
  - All child classes of `GameObject` have `move()` and `checkCollision()` which hides their implementation form user
  - `Game` class has functions like `_update() ,scores,_printArray` etc
  - `GameObject` class has functions like `move(),retcoorlength(),draw()`
  - `Ball` class has functions like `move(), checkCollision(),checkCollisionwithBrick()` etc
  - `Paddle` class has functions like `move(),checkCollision(),removePowerUp()`
  - `Brick` class has functions like `colorBrick(), strengthColor()`
  - `PowerUp` class has functions like `move(), checkCollision()`