CS6.301: Design and Analysis of software systems **Spring 2021**

Assignment-2

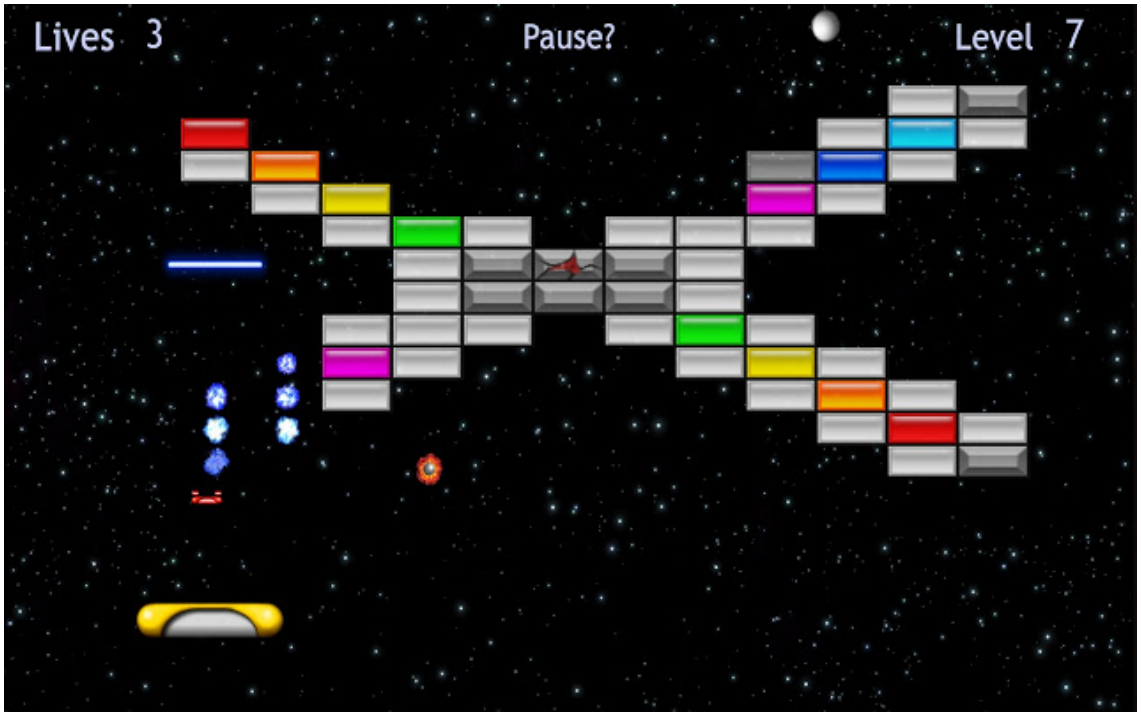
Released: 5th Feb, 2021 *Deadline: 17th Feb, 2021*

Instructions

- This assignment is designed to get you familiar with object oriented programming(OOPS) concepts. You have to use classes and objects to write good and modular code.
- **Make sure your code is modular and extensible to new features.**
- Submit all your code files and README.md as a zip file, named as <Rollnumber>.zip
- The README.md file should describe the game, various rules and all the functionalities. Also include a quick game guide.
- Ensure that the submitted assignment is your original work. Please do not copy any part from any source including your friends, seniors, and/or the internet. Any such cases would lead to a **straight 0** in the assignment.
- **Use Moodle only for all your queries.**

Introduction

For this assignment, you need to build an arcade game in **Python3 (terminal-based)**, inspired from the old classic brick breaker similar to this. The player will be using a paddle with a bouncing ball to smash a wall of bricks and make high scores! The objective of the game is to break all the bricks as fast as possible and beat the highest score! You lose a life when the ball touches the ground below the paddle.
Concepts of object oriented programming must be present in your code and the game must be a basic simulator of brick breaker.



OOPS Concepts(30 points)

Make sure your code exhibits OOPS concepts as it is the main objective of the assignment.:

- **Inheritance**: You could have one main brick class and have different types of bricks inherited from it.
- **Polymorphism**: Have one power-up class and override the functions to execute power-up in child classes.
- **Encapsulation**: Class and object based approach for all the functionality implemented.
- **Abstraction**: Intuitive functionality like move(), break(), collide(), etc, showing away inner details from the end user.

Startup(5 points)

Initially, the ball would appear randomly on any part of the paddle(top). The player can move and release the ball at their will(as shown in this video). Place the bricks in any design/pattern you wish.

Movement(10 points)

- A and D keys move the paddle to left and right directions respectively.
- Movement of the ball. (Collision handling is described in subsequent section)

Collision Handling(20 points)

Must handle elastic collisions as follows:

- **Ball and bricks(5 points)**:
  - Handle collisions with four sides of the brick. The reflections should be based on the side the ball hits the brick.
  - Each time a collision happens, the strength of brick should decrease by one unit (except for unbreakable bricks) and when the strength becomes zero, the brick should disappear.
- **Ball and paddle(10 points)**:
  - Handle collisions with only the top face of the paddle
  - The direction of movement of the ball after collision with the paddle will depend on the distance from the center of the paddle and the collision point, i.e further the ball hits from the center, more the deflection (See gameplay video).
- **Ball and wall(5 points)**:
  - Handle collisions with 3 sides of the walls (except the bottom one).
  - The reflections should be based on the side the ball hits the wall.
  - The ball is lost when it hits the bottom wall (missing the paddle).

Bricks(10 points)

- Required different colored bricks (at least 3) , with their color indicating their strength.
- Bricks with more strength require more number of hits to make them disappear.
- With decrease in strength of a brick due to hits from the ball, the color of the brick must change accordingly.
- Also required to implement Unbreakable bricks which cannot be broken by the ball.

Score and Time(5 points)

- Display Score, Lives remaining and Time played on the screen. Calculate the score as per your wish considering the number of bricks that have been broken.
- A life is lost when all the balls in the game are lost (touches the ground below the paddle).

Power-Ups(20 points)

Each of these power-ups are appear as catch-able object randomly on destroying a brick (the power-up should appear at the same place where the brick is destroyed). The power-up is activated on catching it with the paddle. Movement of power-up drop should be straight downwards and with constant speed. Power-up movements should be unaffected by any other component except the paddle itself. Use different symbols for different power-ups.
All power-ups are are present only for a fixed amount of time and lost at loss of a life.

1. **Expand Paddle**: Increases the size of the paddle by a certain amount.
2. **Shrink Paddle**: Reduce the size of the paddle by a certain amount but not completely.
3. **Ball Multiplier**: Each of the balls which are present will be further divided into two.
4. **Fast Ball**: Increases the speed of the ball.
5. **Thru-ball**: This enables the ball to destroy and go through any brick it touches, irrespective of the strength of the wall.(Even the unbreakable ones which you couldn't previously destroy)
6. **Paddle Grab**:Allows the paddle to grab the ball on contact and relaunch the ball at will. The ball will follow the same expected trajectory after release, similar to the movement expected without the grab.

Bonus(20 points)

Additionally, you can implement the following features to be eligible for some extra marks:
Exploding bricks:

- This is a special type of brick which on breaking with the ball would explode resulting in the destruction of all the bricks adjacent to it(diagonally, vertically and horizontally)
- These bricks are not found individually and are to be placed in groups(linearly/diagonally) of minimum size 6 and contact with either one of them would lead to a chain reaction among this group (refer gameplay video) .

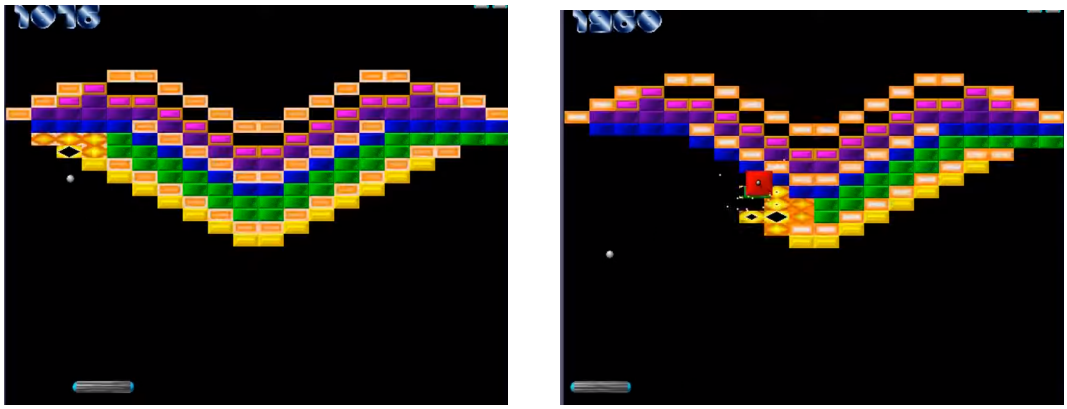


Figure 1: Example of explosive bricks

Library Usage

- Your code **must** be in **Python3**. Python2 is not allowed.
- **No curses libraries (like pygame) are allowed**. **Only libraries allowed are colorama and numpy**. In case of any doubts about whether a particular library is allowed, please post it on the Moodle thread and get it clarified from one of the TAs.

Deliverable

Submit a single <RollNumber>.zip file containing the README.md file and your code files. Ensure your code is modular with multiple python files. Here is a sample of what the .zip file looks like when extracted: