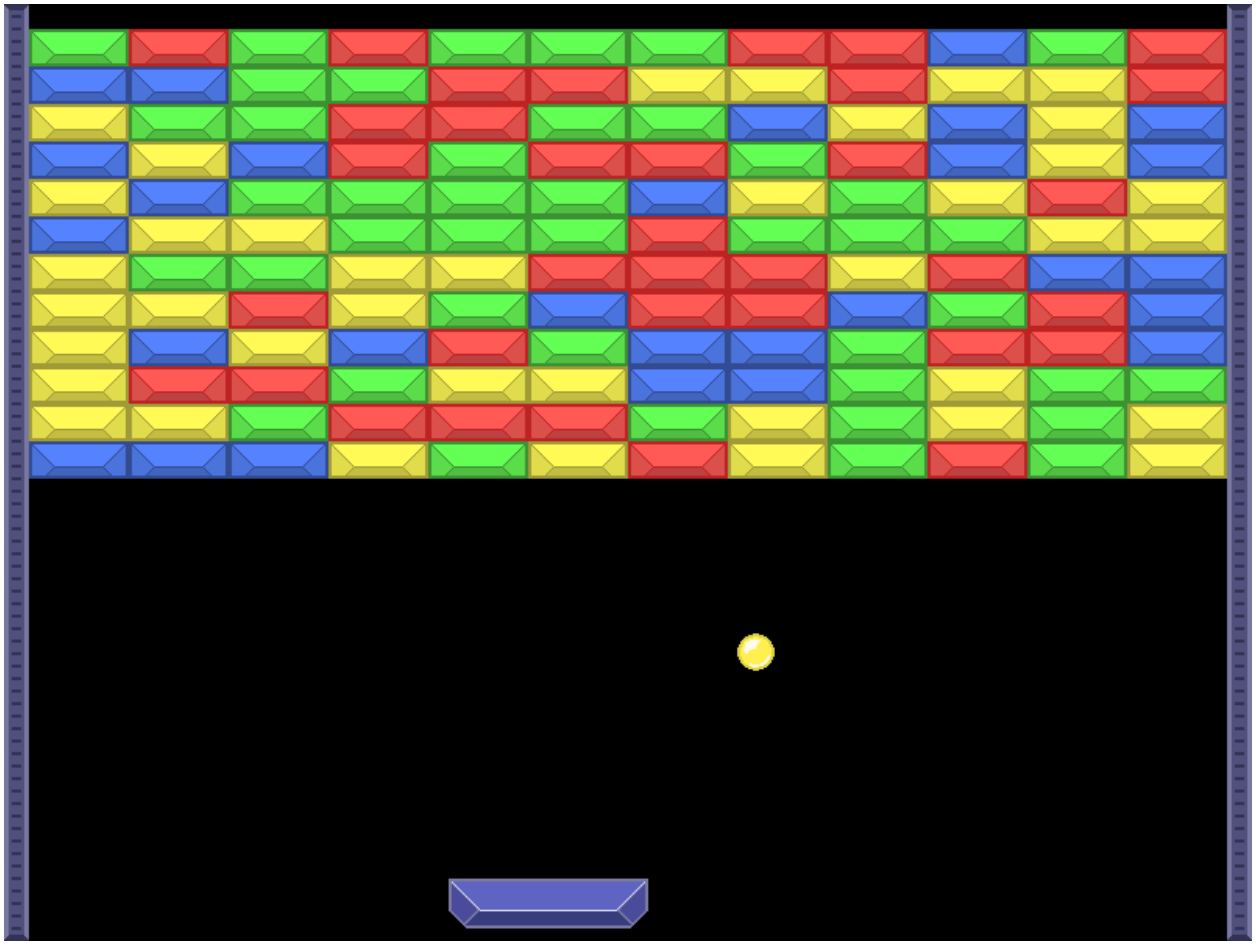
Object Oriented Programming (CS1004)

Semester Project

Arkadriod Brick Breaker Game



You need to get the idea and scope of the project by reading this document. Apply OOP concepts to implement the project. Within the scope of the project, feel free to make customization. Whatever your assumptions are, write down at the end of the rubric If your assumption(s) are not contradictory with the rubric/requirements, no marks will be deducted. This project understanding is the part of your evaluation. ANY INVALID QUERY/COMMENT WILL BECOME PART OF YOUR PROJECT.

Due Date: 7th May 2024, 11:59 PM (Hard Deadline)

Instructions:

- **It's an Individual Project, the deadline will not be extended.**
- Make sure that you read and understand each and every instruction. If you have any questions or comments, you are encouraged to discuss your problems with your teacher. Any invalid query that is beyond the project scope will become part of your project.
- Plagiarism is strongly forbidden and will be very strongly punished. If we find that you have copied from someone else or someone else has copied from you (with or without your knowledge) both of you will be punished. You will be awarded (straight zero in the project — which can eventually result in your failure) and appropriate action as recommended by the Disciplinary Committee (DC can even award a straight **F** in the subject) will be taken.
- Try to understand and do the project yourself even if you are not able to complete the project. Note that you will be mainly awarded on your effort not on the basis whether you have completed the project or not.
- **Divide and conquer**: since you have many days so you are recommended to divide the complete task in manageable subtasks. We recommend to complete the drawing and design (i.e. number of classes and their relationships) phase as quickly as possible and then focus on the intelligence phase.
- Before writing even one line of code, you must design your final project. This process will require you to break down and outline your program into classes, design your data structure(s), clarify the major functionality of your program, and pseudo-code important methods. After designing your program, you will find that writing the program is a much simpler process.
- You can use Visual Studio or Eclipse or any other IDE on **UBUNTU** to implement the project.
- No Marks will be given if you do not use the **object-oriented principles** you have learned during the course.
- Anything you haven't learnt in OOP course, or use of any built in functions are not allowed
- **Imagination Powers**: Use your imaginative powers to make this as interesting and appealing as you can think of.
- Combine all your work in one .zip file.
- **Name the .zip file as ROLL-NUM-SECTION.zip (e.g. 230002- B.zip).**
- Submit the .zip file on Google Classroom, submission will not be accepted on any other platform for example email.
- You must follow the submission instructions, as failing to do so will get you a zero in the Project.

Objective:

The objective of this project is to develop a classic brick breaker game called Arkadriod using Object-Oriented Programming (OOP) concepts. This project aims to enhance students understanding and implementation skills of OOP principles such as encapsulation, inheritance, polymorphism and abstraction.

Overview:

Arkadriod is a modern adaptation of the classic brick breaker arcade game. Players control a paddle at the bottom of the screen to bounce a ball and break bricks arranged in various patterns at the top of the screen. The player progresses through levels by breaking all the bricks on the screen without letting the ball fall off the bottom of the screen. The game ends when all levels are completed, or the player runs out of lives. Play this game from the following links, our task is to implement this in C++ using OOP.

https://poki.com/en/g/brick-breaker

The following image is just for reference.

Features:

- **Paddle control**
  Players can move the paddle horizontally (bottom paddle with mouse, upper paddle with keyboard keys w & r) to bounce the ball and prevent it from falling.
- **Brick layout**
  Bricks are arranged in different pattern on each level, with varying durability and point values.
  Brick pattern consists of 4 different types of bricks.
  Green brick (gets broken with one ball collision)
  Pink brick (gets broken with two ball collisions)
  Blue brick (get broken with three ball collisions)
  Red brick (get broken with three ball collisions)
  Yellow brick (get broken with two ball collisions)

- **Ball movement**
  The ball should bounce off surfaces, changing direction based on the angle of impact. (look at the provided link of game above https://poki.com/en/g/brick-breaker)

- **Power-ups & downs**
  Implement power-ups that can have various effects on gameplay, such as increasing paddle size, adding extra balls, or slowing down the ball speed.
  You need to implement four power-ups with different functionalities as mentioned below:

  Green power-up (shape is traingle)
  - When green power-up hit the paddle, it increases the size of paddle (becomes double).

  Pink power-down (shape is square)
  - When pink power-up hits the paddle, it reduces the size of paddle (becomes half).

  Blue power-up (shape is circle)
  - When blue power-down hits the paddle, it slows down the ball speed (for 5sec)

  Red power-down (shape is rectangle (two squares horizontally placed make a rectangle))
  - When red power-down hits the paddle, it speed up the ball.(for 5 seconds)

  Yellow PowerUp **(square shape)**

  - Add two more balls in the game, for next 5 seconds

- **Score**
  Keep track of the player's score, based on the number of bricks destroyed.

Levels

Design multiple levels with increasing difficulty, featuring different brick arrangements and increasing ball speed. You can design three levels.

- In level 1, bricks pattern is a rectangle
- In level 2 has the following pattern
- Level 3 is explained below.

- **Lives and game over**

  Players have a 2 number of lives for each level, and the game ends when all lives are lost. For every next level, the lives count will be reset to two (2)
- **User interface**

  Develop a user-friendly interface with menus for starting the game, pausing, resuming and score history.

File handling:

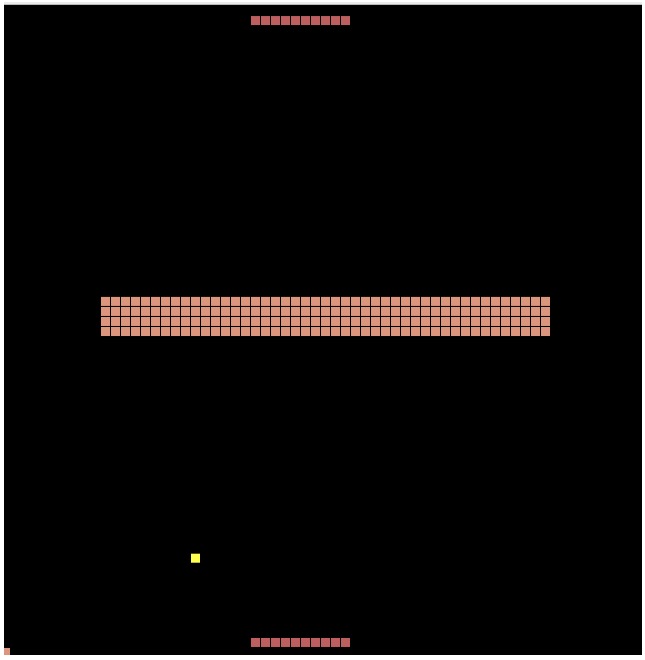
- High scores will be stored in files with the player names.

Level 3

- Brick pattern is following, with recursion
- You need to make a extend game, like there is need of two paddles. Upper paddle (controlled by keyboard) and lower paddle, controlled by mouse drag.
- The power Up & power down affects the same for both paddles.

- If the ball falls down for any paddle, game will get over.

The following image is just for reference, in your case the brick pattern will contain different kind of bricks.



Bonus points:

- Players can resume the game (after leaving midway). You need to store name and his current state of game.
- The score bar & your Roll number on the canvas.
- The use of polymorphism & copy constructor anywhere in the project will be considered as a bonus.
- Separate header and cpp files.
- Ball color will change to brick color after collision, means ball hits red brick then ball color changes to red and so on
- Paddle color changes to ball color after every collision.

Marks deduction:

1. If code has been written without classes/structs.
2. If you have not applied Object oriented principles, zero marks will be awarded.

Implementation Details:

1. **If more than 1(one) Global Variable has been used in coding (Maximum 1 variable allowed).**
2. **You are not allowed to make global functions.**
3. Brick, Food, Paddle & Ball are the essential classes, you can make more classes but these four are mandatory. Now, it depends on your logic how to use their instance(s).