Additional Requirements

- When the player has won a level, this should move to the next level.
- You will need to transition to the ‘NEXT LEVEL’ screen when the player has won a level, it should hold the transition 'NEXT LEVEL' for around 2 seconds or transition via keypress. If the player reaches the last level, the game should show that you 'YOU WIN!' screen. Once the player has won the game, it should return back to the first level.
- If the player lost, your program will need to display the ‘GAME OVER’ screen for around 2 seconds or transition via keypress. When the player loses, it should return back to the first level.

Please outline in your readme the first json file name. When examining your code base, we may use our own levels to test and ensure that your level loading is working as intended.

We have provided you with a font called “PressStart2P-Regular.tff” which you will use for displaying the level messages.

Use font methods associated with the processing library

Powerups

Some bricks will contain a powerup and drop towards the bottom of the screen once destroyed, the paddle can collect it and have an affect applied to a certain object.

- Multi-ball ('multiball' in json), there are now 3 balls on screen, only when all balls are no longer on the screen does the player lose the level.
- Bigger Paddle for 60 seconds ('paddleup' in json), this increases the width of the paddle by an additional 10 pixels.

If the power up is active when the player finishes the level, the level will reset so no power ups are active.

This property is associated with the brick object.

You have access to the additional sprites within the 'AdditionalResources' file.

Brick change

- Some bricks can take multiple hits to break, the JSON object will have hp information for brick objects and have a brick limit to reach for the player to win.

```
{
    "x" : 0,
    "y" : 0,
    "id" : "brick_blue",
    "hp" : 1,
    "powerup" : "multiball"
},
```

Levels

- Each JSON level file will now have a property ('next_level') which will refer to the next level by its file name.

```
{
    "name" : "Level1",
    "next_level" : "level2",
    "bricks" : [
        {
            "x" : 0,
            "y" : 0,
            "id" : "brick_blue",
            "hp" : 1
        },
        {
            "x" : 20,
            "y" : 0,
            "id" : "brick_red",
            "hp" : 1
        },
        {
            "x" : 40,
            "y" : 0,
            "id" : "brick_pink",
            "hp" : 2
        },
        ...
    ]

}
```