GRIFFITH COLLEGE DUBLIN

Student name:	**Lucas Rodrigues Crema**				
Student number:	**3133143**				
Faculty:	**Computing Science**				
Course:	**BSCH**		**Stage/year:**	**2**	
Subject:	**Software Development 2**				
Study Mode:	Full time	☑		Part-time	
Lecturer Name:	**Gemma Deery**				
Assignment Title:	**Personal Report for Review 2**				
Date due:	**18/04/2025**				
Date submitted:	**18/04/2025**				

Plagiarism disclaimer:

I understand that plagiarism is a serious offence and have read and understood the college policy on plagiarism. I also understand that I may receive a mark of zero if I have not identified and properly attributed sources which have been used, referred to, or have in any way influenced the preparation of this assignment, or if I have knowingly allowed others to plagiarise my work in this way.

I hereby certify that this assignment is my own work, based on my personal study and/or research, and that I have acknowledged all material and sources used in its preparation. I also certify that the assignment has not previously been submitted for assessment and that I have not copied in part or whole or otherwise plagiarised the work of anyone else, including other students.

Signed: Lucas Rodrigues Crema **Date: 18/04/2025**

Please note: **Students** MUST **retain a hard / soft copy of** ALL assignments as well as a receipt issued and signed by a member of Faculty as proof of submission.

For review 2, I added a store to the game and an inventory, the player can interact with the store to sell the items that they currently have in the store and turn them from blocks to points.

On the GameManager Class, I initialized an instance of store:

```
store = new Store("bank.png", Constants.SCREEN_WIDTH/2, Constants.GROUND_HEIGHT, Constants.STORE_WIDTH, Constants.STORE_HEIGHT);
```

Followed similar logic to draw store as the one to blocks and players:

```
//Draw store
graphics.drawImage(store.getImage(), store.getX(), store.getY(), store.getWidth(), store.getHeight(), panel);
```

Added checkStoreProximity method to update so it constantly checks if player are trying to access the store:

```
public void update()
{
    player.update();
    player2.update();

    checkStoreProximity(player);
    checkStoreProximity(player2);
```

Main method that checks if player's position is the same as store's and then checks if specified button is being pressed:

```
225     //Method checks if either player is close to the store and pressing their designated button
226     public void checkStoreProximity(Player player)
227     {
228         int storeX = store.getX();
229         int storeY = store.getY();
230         int storeWidth = store.getWidth();
231         int storeHeight = store.getHeight();
232
233         //If in the same position, check if pressing button
234         if (player.getX() >= storeX - storeWidth / 2 && player.getX() <= storeX + storeWidth / 2 && player.getY() >= storeY - storeHeight / 2 &&
    player.getY() <= storeY + storeHeight / 2 )
235         {
236             //If pressing button, call method to sell inventory
237             if (activeKeys.contains(Constants.STORE_SELL_P1) || activeKeys.contains(Constants.STORE_SELL_P2))
238             {
239                 player.sellInventory();
240             }
241         }
242     }
```

Added values related to store on Constants:

```
//Store
public final static int STORE_WIDTH = 70;
public final static int STORE_HEIGHT = 70;

public final static int STORE_SELL_P1 = 86;
public final static int STORE_SELL_P2 = 77;
```

Added getters and setters for variables under Blocks (still need to be properly implemented):

```
//Getters and Setters
public double getHardness()
{
    return hardness;
}


public void setHardness(double hardness)
{
    this.hardness = hardness;
}


public double getValue()
{
    return value;
}


public void setValue(double value)
{
    this.value = value;
}
```

ArrayList under player class that holds blocks:

```
private List<Block> inventory = new ArrayList<>();
```

collectBlock add any blocks to the inventory and sellInventory adds value of blocks to score and empties inventory after doing so:

```
//Adds blocks to inventory to be sold
public void collectBlock(Block block)
{
    inventory.add(block);
}

//"Selling" inventory by adding values to score and then clearing inventory
public void sellInventory()
{
    for (Block block : inventory)
    {
        score += block.getValue();
    }

    inventory.clear();
}
```

Ended up not using any external links.

Main challenge I encountered was getting checkStoreProximity() to work as intended, but ultimately just had to compare the X,Y, width and height values of player to store and then check if specified button was being pressed or not.