Assignment Cover Sheet

Student name:	**Polina Shtefan**		
Student number:	**3120697**		
Faculty:	**Computing Science**		
Course:	**BSCH**	**Stage/year:**	**2**
Subject:	**Software Development 2**		
Study Mode:	Full time **\/**	Part-time	
Lecturer Name:	**Gemma Deery**		
Assignment Title:	**Review 2**		
No. of pages:			
Disk included?	Yes	No	
Additional Information:	(ie. number of pieces submitted, size of assignment, A2, A3 etc)		
Date due:	**30-Apr**		
Date submitted:	**30-Apr**		

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: _____ Stefan _____ **Date:**

_____**30/04/2025**_____

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.*

Coin Collection System

To build the coin system, I created a CoinManager class that tracks all the coins and their interactions with both players. It holds a list of Coin objects and ensures that coins are updated, drawn, and collected properly as the game runs.

I wrote the addTestCoins() method to spawn coins in a grid layout, but added small random shifts in their X and Y positions to make it feel more natural, not super perfect and robotic. I also skipped certain grid spots to leave the room or create variety. Coins are only added if inside the screen boundaries, avoiding any weird off-screen bugs.

Each coin has its class — it stores position, size, a hitbox, and a flag to say whether it's active or already collected. The update() method inside Coin keeps its hitbox in sync with its position and handles a simple animation loop using frame counters.

Inside CoinManager, I have an update() method that goes through all coins and, for those still active, updates them and checks if either player has collected them. That's done using the checkCollision() method, which uses a

rectangle intersection with the players' hitboxes. If there's a hit, the coin's set to inactive.

I also made a render() method to draw the coins on screen — they only show up if they're still active. Once collected, they disappear.

Instead of classic coin icons, I used animated strawberries to make collecting feel more playful and unique. Each strawberry cycles through frames from a sprite sheet, giving it a soft shimmering effect that stands out on screen and makes the game feel more alive.

Finally, I added allCoinsCollected() to check if all coins are gone. This is useful for level logic later (like checking when players can move on).

This setup makes it easy to drop in coins, animate them, and have both players interact with them individually. It's clean, works well with the rest of the game loop, and feels satisfying when you pick them up.