GRIFFITH COLLEGE DUBLIN

Student name:	**Nahal Kianpour**					
Student number:	**3124767**					
Faculty:	**Computing Science**					
Course:	**BSCH/BSCO/EXCH**			**Stage/year:**	**2**	
Subject:	**Software Development 2**					
Study Mode:	Full time	☑		Part-time		
Lecturer Name:	**Gemma Deery**					
Assignment Title:	**Report 3**					
Date due:	**30/04/2025**					
Date submitted:	**30/04/2025**					

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: _Nahal Kianpour Lirharani_ **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.

1. Features I Worked On..... 3
1.1 Features I Contributed To..... 3
2. What Was Completed..... 3
2.1 Completed Functionality..... 3
3. Who Did What..... 3
3.1 Team Role Breakdown..... 3
4. Challenges Faced / What I Learned..... 4
4.1 Challenges and Key Learnings..... 4
5. Git Log..... 5
5.1 Git Commit History..... 5

1. Features I Worked On

1.1 Features I Contributed To

In this project, I created a full countdown timer system using a custom Timer class and HeadsUpDisplay, which shows a live mm:ss countdown on screen. When time expires, the game shows a centered “Time’s Up!” message using a showGameOver flag to ensure it’s triggered only once. I wrote a full suite of unit tests to validate all time-related behaviors.

I also fixed broken menu functionality by removing the old MainClass.main(null) call and launching the Game instance directly when the Start button is clicked. To improve gameplay balance, I increased the jump height by adjusting jumpSpeed, allowing players to reach upper platforms.

For win conditions, I added the allCoinsCollected() method in CoinManager to detect when all collectible items are gone. I connected it with Game.java to trigger a centered green “You Win!” message and stop the timer. I also debugged and fixed an issue where coins spawned off-screen and remained uncollectible by adding bounds-checking in the spawn logic.

2. What Was Completed

2.1 Completed Functionality

- **Timer and HUD System**: Designed and implemented a custom Timer class with mm:ss formatting, frame-accurate countdown using delta time, and a single-trigger onTimeUp() method to handle game-over logic. Created a HeadsUpDisplay class to draw the live timer in the top-left of the screen. Ensured clean state handling using showGameOver and timeUpHandled flags.

- **"You Win" Condition with CoinManager**: Developed the allCoinsCollected() method to detect win condition when all coins are picked up. Integrated this logic into the Game class to stop the timer and display a centered green “You Win!” message using a showWinMessage flag.

- **Coin Spawn Bug Fix**: Fixed an issue where some coins were spawning outside the visible screen area by adding screen-bound checks in addTestCoins() using Game.GAME_WIDTH and GAME_HEIGHT.

3. Who Did What

3.1 Team Role Breakdown

- **Nahal Kianpour (me):**
  Built the level system and integrated environment rendering using LevelManager and LoadSave. Implemented the UPS (Updates Per Second) performance tracker and

set up a modular testing framework using a testcases package. Refactored core Game logic to prevent null exceptions and maintain layout consistency with global constants.

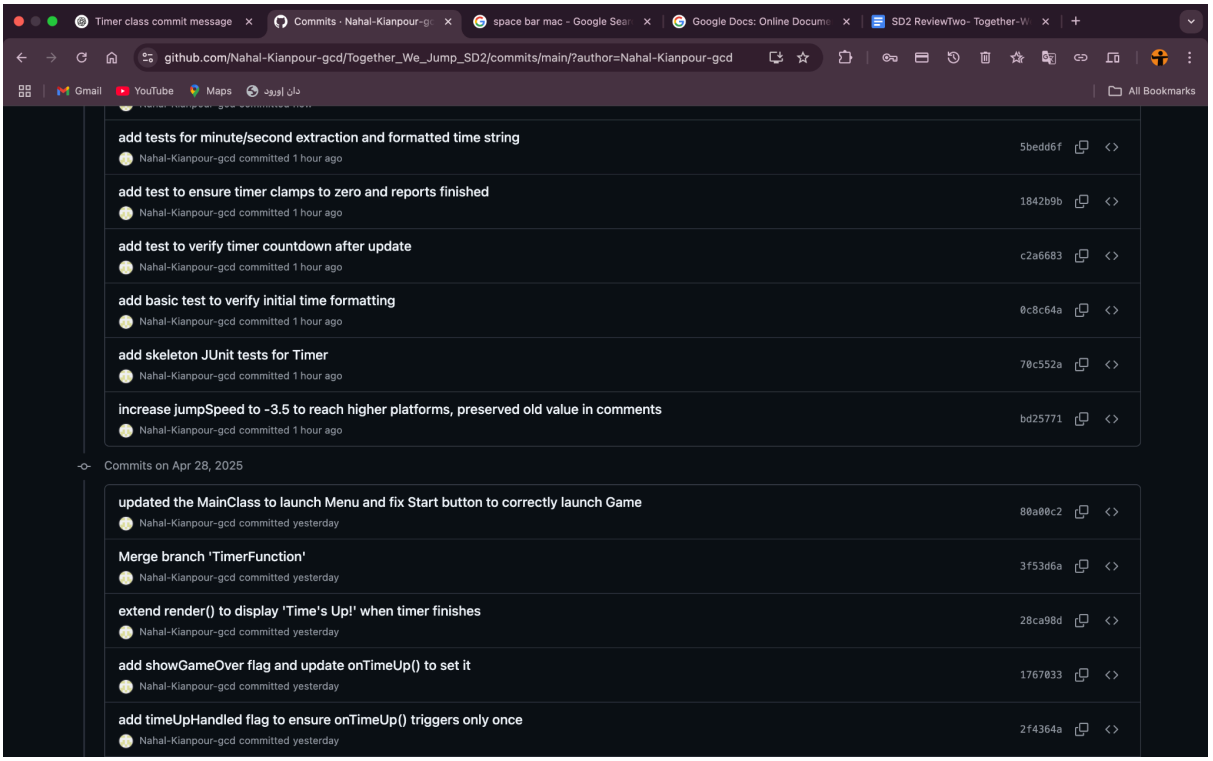
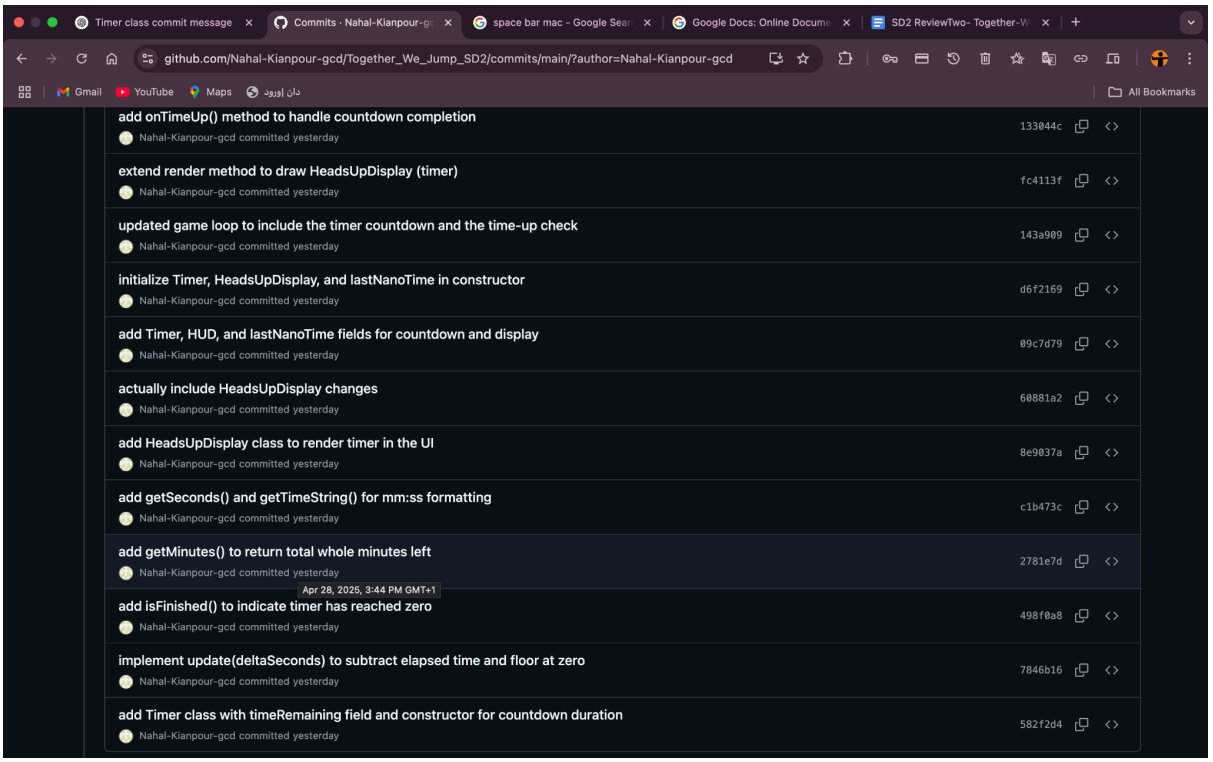
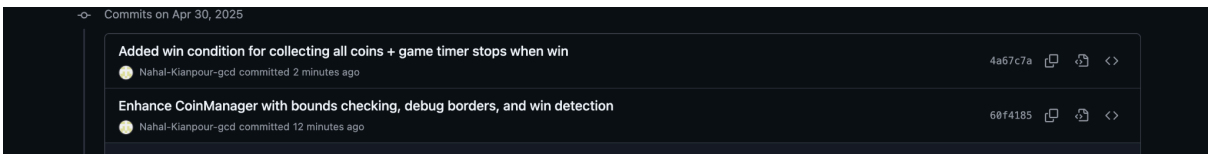
In the later stages, I developed a complete countdown timer system with HUD integration, frame-accurate delta time handling, and a centered "Time's Up!" message. I also implemented the coin win condition using CoinManager, added the "You Win!" screen logic, and fixed a bug where coins spawned outside the screen.

- **Polina Shtefan:**
  Implemented the animation system for characters, organized sprite constants, structured the utilz package, and handled visuals. Later, she also built the core CoinManager and Coin classes to support coin collection logic and in-game currency behavior.
- **Thanh Phuong Hoang:**
  Created the menu system and connected it to the main game flow. She fixed the menu-to-game transition using button listeners and ensured the Start button correctly launches the gameplay session.
- **Veronika Hrytsyk:**
  Implemented the initial collision detection system and created hitboxes to enable player interaction with level objects. She later finished the gravity system and finalized character–platform collision logic for smoother movement and physics.

4. Challenges Faced / What I Learned

4.1 Challenges and Key Learnings

- **Countdown timer and game state handling:**
  The timer system originally froze the game after reaching 0. I implemented a showGameOver flag and a one-time onTimeUp() trigger, which taught me how to manage time-based events cleanly without affecting core updates.
- **Win condition detection and visual feedback:**
  Implementing the win state based on coin collection required careful coordination across game states. I introduced a showWinMessage flag and refined rendering logic to prevent overlaps with game-over messages.
- **Coin positioning bug:**
  Some coins were spawning off-screen, so I updated the placement logic to check bounds using Game.GAME_WIDTH and GAME_HEIGHT. This taught me how to validate object spawning using global constraints.
- **Collaboration and modularity:**
  Working through Git and syncing regularly helped me see the value of modular class design and commit clarity. It was easier to integrate features like the timer and coins into the team codebase without breaking others' work.

5. Git Log

5.1 Git Commit History