

Table of Contents

[Group Info 1](#_Toc206252025)

[Part 1 - Project Management 2](#_Toc206252026)

[Project Planning 2](#_Toc206252027)

[Time Schedule 1](#_Toc206252028)

[Group Meeting Records 3](#_Toc206252029)

[Effort Summary 1](#_Toc206252030)

[Peer Review 1](#_Toc206252031)

[Purpose 1](#_Toc206252032)

[Instructions 1](#_Toc206252033)

[Section 1 – Self-Review [Your Name] 2](#_Toc206252034)

[Section 2 – Peer Review (One per Team Member) 2](#_Toc206252035)

[Section 3 – Self -Reflection 3](#_Toc206252036)

[Peer Rating Summary Table 3](#_Toc206252037)

[Score Adjustment Table 4](#_Toc206252038)

[Part 2 - Requirement Analysis 5](#_Toc206252039)

[Functional Requirements (FR) 5](#_Toc206252040)

[Non-Functional Requirements (NFR) 9](#_Toc206252041)

[Use Case Diagram 10](#_Toc206252042)

[Activity Diagram 10](#_Toc206252043)

[UML Class Diagrams 11](#_Toc206252044)

[Part 3 - Implementation 11](#_Toc206252045)

[GitHub 11](#_Toc206252046)

[GitHub Advanced 11](#_Toc206252047)

[File Structure 11](#_Toc206252048)

[Coding Snippets 11](#_Toc206252049)

[Part 4 - Demonstration 12](#_Toc206252050)

[Demo Video Link 12](#_Toc206252051)

# Group Info

**Campus**: Gold Coast

**Group Number**: 28

**Group Members:**

1. Alexander Abbosh (s5173344)
2. Christopher Burrell (s5237645)
3. Vishva Pandya (s5255213)
4. Bailey Reeves (s5189766)

**Team Leader:** Christopher Burrell

**Lab Teacher:** Larry Wen

# Part 1 - Project Management

## Project Planning

The following section comprises of the project planning details for Milestone 1. It outlines the major tasks, assigned responsibilities, estimated and actual completion times, as well as records of group meetings. This ensures transparency, effective time management, and clear communication within the team while tracking progress toward the development of the Tetris project.

### Time Schedule

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **#** | **Task** | **Student** | **Planed Time** | **Actual Time** | **Cumulative Time** | **Finish Date** |
| 1 | Set up GitHub repository & permissions | CB | 1hr | 1hr | 1hr | 29/07/25 |
| 2 | Meeting- Project Kickoff | All | 1hr | 1hr | 2hr | 30/07/25 |
| 3 | Research JavaFX requirements & review demo | VP | 2hr | 2hr | 4hr | 31/07/25 |
| 4 | Draft Functional Requirements | BR | 1.5hr | 1.5hr | 6.5hr | 01/08/25 |
| 5 | Draft Non-Functional Requirements | CB | 1.5hr | 1.5hr | 7hr | 01/08/25 |
| 6 | Meeting – Review FR & NFR | All | 1hr | 0.5 | 7.5hr | 02/08/25 |
| 7 | Create Use Case Diagram | AA | 1.5hr | 2hr | 9.5hr | 02/08/25 |
| 6 | Create Activity Diagram | AA | 2hr | 2hr | 11.5hr | 03/08/25 |
| 7 | Implement Splash Window | AA | 1.5hr | 1.5hr | 13hr | 04/08/25 |
| 8 | Develop Main Screen navigation | VP | 2hr | 2hr | 15hr | 05/08/25 |
| 9 | Meeting – Review Splash Screen and Main Screen Nav | All | 1.5hr | 1.5hr | 16.5hr | 12/08/25 |
| 10 | Implement task checklists and Directory Setup | BR | 1.5hr | 2hr | 18.5hr | 13/08/25 |
| 11 | Implement Basic Gameplay Features (board, tetrominos, gameView) | BR | 3hr | 3hr | 21.5hr | 14/08/2025 |
|  | Implement Tetrominos Movement from Player Input |  | 3hr |  |  |  |
|  | Implement Tetrominos Rotation from Player Input |  | 3hr |  |  |  |
|  | Implement Gravity Logic |  | 3hr |  |  |  |
|  | Implement Piece Lock Logic |  | 2hr |  |  |  |
|  | Implement Erase Full Rows Logic |  | 2hr |  |  |  |
|  | Implement Game Over State |  | 2hr |  |  |  |
|  | Meeting – Review Gameplay Basics | All | 1hr |  |  |  |
| 12 | Implement High Score Screen (dummy data) |  | 2hr |  |  |  |
| 13 | Implement Pause/Resume Function |  | 2hr |  |  |  |
| 14 | Implement Tetrominos Colour (CSS) |  | 2hr |  |  |  |
|  | Adjust Aesthetics and Overall look of Game (CSS) |  | 3hr |  |  |  |
|  | Implement Exit Function |  | 2hr |  |  |  |
|  | Meeting – Functional Requirements | All | 1.5hr |  |  | 18/08/25 |
|  | Implement Colour Blondness Support feature in Configuration Screen |  | 3 |  |  |  |
|  | Stability Testing (20 min without crash) |  | 2hr |  |  |  |
|  | Performance Testing (Maintain min 30fps) |  | 3hr |  |  |  |
|  | Build verification on Java 21 + JavaFX |  | 2hr |  |  |  |
|  | Meeting – Non Functional Requirements | All | 1.5hr |  |  | 19/08/25 |
|  | Collect GitHub evidence (commits, pull requests, tags) | VP | 1hr |  |  | 20/08/25 |
|  | Record and edit demo video | All | 4hr |  |  | 20/08/25 |
|  | Compile, proofread, and finalise Milestone 1 report | CB | 3hr |  |  | 21/08/25 |
|  | Final review meeting & submission | All | 2hr |  |  | 22/08/25 |

Table . Time Schedule chart. Abbreviations: Alexander Abbosh (AA), Christopher Burrell (CB), Vishva Pandya (VP), Bailey Reeves (BR).

### Group Meeting Records

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **#** | **Meeting Topics** | **Attendance** | **Loc** | **Software** | **Date and Time** | **Comments** |
| 1 | Project Kickoff & Role Allocation | AA, CB (VP,BR chat only) | Online | Teams | 12am-2pm  30/07/2025 | Introduced team members, discussed project scope (Tetris), allocated roles based on strengths **CB**: Project Planning, Task Allocation, Peer Review, FR/NFR. **AA**: Splash Window, Use Case Diagram, Activity Diagram, Assist FR/NFR , **VP**: Main Screen, Config Screen, UI Navigation, GitHub Evidence, **BR**: Score system, Game Logic, pause functionality, exit function. Decided GitHub repository setup to be first task. |
| 2 | Review FR & NFR | All | Online | Teams | 2pm – 3pm  01/08/2025 | CB and BR presented initial FR and NFR drafts. Group refined descriptions to ensure testability and consistency. Agreed to finalise FR/NFR before diagrams are started. |
| 3 | Review Use Case Diagram | All | Online | Teams | 3pm-3:30pm  02/08/2025 | AA presented first draft of Use Case diagram. Team confirmed inclusion of all FRs. Suggested adding explicit “Pause Game” and “Exit Game” cases. |
| 4 | Review Splash Screen and Main Screen Nav | AA,CB,VP (BR Chat only) | Online | Teams | 12am-2pm  06/08/2025 | VP demonstrated Main Screen navigation flow. AA showed splash screen timing. Group confirmed navigation links between all screens worked as intended. Decided to move to Activity Diagram and high score logic next. |
| 5 | Project check-in | All | Online | Teams | 10am-10:30am  13/08/2025 | Reviewed progress on Milestone 1 tasks and confirmed all allocated sections were on track for completion by the due date. Each member provided a quick status update on their assigned work (FRs, NFRs, diagrams, and implementation tasks). Discussed any outstanding issues, including minor adjustments to wording in requirements and ensuring GitHub evidence was up to date. Agreed on the final timeline for compiling the report and recording the demo video. |

Table . Group Meeting Records. Abbreviations: Alexander Abbosh (AA), Christopher Burrell (CB), Vishva Pandya (VP), Bailey Reeves (BR).

### Effort Summary

|  |  |  |
| --- | --- | --- |
| **Student Name (#Id)** | **Planed hours** | **Actual hours** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| Total working hours: |  |  |
| Average working hours per person |  |  |

## Peer Review

### Purpose

The peer review process in OOSD is designed to promote collaborative learning, develop feedback skills, and encourage shared responsibility in team-based projects. It helps students reflect on their own contributions, evaluate team dynamics, and engage in transparent, constructive dialogue for continuous improvement.

### Instructions

The peer review form is to be completed and submitted at the end of each project milestone (Milestone 1, Milestone 2, and the Final Milestone). It is structured into three sections:

* **Section 1: Self-Review** – A reflection on one’s own role and contributions during the milestone.
* **Section 2: Peer Review** – Constructive evaluation of each team member’s performance, completed individually for every teammate (excluding oneself).
* **Section 3: Self-Reflection** – A critical reflection on the feedback received from peers, focusing on its relevance and impact on individual learning and improvement.

Note: Every team member, including the team leader, must complete Sections 1, 2, and 3 individually and include them in the report. The team leader is also responsible for summarizing the results in the Peer Review Summary Table.

### Section 1 – Self-Review [Your Name]

Please answer the following questions to reflect on your own role and contribution during this milestone.

1. **What was your primary role in this milestone?** (e.g., front-end developer, tester, documentation lead)
2. **What specific tasks or deliverables did you complete?** (Provide clear examples of your work.)
3. **What challenges, if any, did you face while contributing to the project?** (e.g., technical issues, time constraints, coordination etc.)
4. **What is one aspect of your contribution that you would like to improve in the next milestone?**
5. **On a scale of 1 to 10, how would you rate your overall contribution to the team?** (1 = very poor, 10 = excellent)

### Section 2 – Peer Review (One per Team Member)

**Instructions:**

This section must be completed **once for each of your team members**, excluding yourself. Please **replicate the peer review questionnaire below** and complete it separately for everyone in your group. You are required to review all other members of your group individually.

**For instance**, in a team of four students, each member is expected to submit three individual peer reviews one for each teammate.

**Ensure** your responses are constructive, evidence-based, and respectful. Your feedback should aim to support your peers' development and contribute to a fair and reflective team environment.

**Reviewee Name:**

1. **What was this team member’s main role during this milestone?** (State the key responsibility assigned to them.)
2. **What were their key contributions?** (Mention both the quantity and quality of their completed work.)
3. **How would you evaluate their professionalism?** (Consider teamwork, responsiveness, reliability, and communication etc.)
4. **What is one respectful and specific suggestion for how they could improve in the next phase?**
5. **On a scale of 1 to 10, how would you rate their overall contribution to the team?** (1 = minimal contribution, 10 = outstanding contribution)

*Note: After completing the peer review process, team members are expected to share the written feedback with each reviewee. This step ensures that all students can reflect on the feedback received, promoting open communication and continuous improvement within the team.*

### Section 3 – Self -Reflection

Please reflect on your learning and team experience during this milestone.

1. **Was the peer review process fair and helpful in identifying your contributions and role within the team?** (Explain your experience. Was it clear, balanced, and accurate in reflecting what you did?)
2. **Did the feedback you received help you identify any areas for personal or team improvement?** (Explain one insight or takeaway from the feedback shared by your teammates.)
3. **On a scale of 1 to 10, how useful was this peer review cycle for your learning and team development?** (1 = Not useful at all, 10 = Extremely useful)
4. **How did you respond to the feedback you received from your peers?** (Did you agree with it? Did it lead you to change anything in how you approached the project?)

### Peer Rating Summary Table

This table is used to consolidate peer review scores for each team member. Upon completing all individual peer reviews, the peer rating scores should be entered into the summary table as follows:

* Each **row** represents a student who has provided peer review scores (reviewer).
* Each **column** represents a student who has received peer review scores (reviewee).
* A numerical score between **0 and 10** should be entered to reflect the reviewer’s assessment of the reviewee’s overall contribution during the milestone.
* Self-ratings are not permitted; students must not evaluate their own performance.

All scores entered should be consistent with the corresponding written evaluations completed in Section 2. This summary table facilitates the calculation of average peer review scores and supports further analysis by the teaching team.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Reviewer/ Reviewee | Member 1 | Member 2 | Member 3 | Member 4 | Member 5 |
| Member 1 | - | 10 | 10 | 10 | 10 |
| Member 2 | 10 | - | 10 | 10 | 10 |
| Member 3 | 10 | 10 | - | 10 | 10 |
| Member 4 | 10 | 10 | 10 | - | 10 |
| Member 5 | 10 | 10 | 10 | 10 | - |
| Total Score | 40 | 40 | 40 | 40 | 40 |
| Average Score | 10 | 10 | 10 | 10 | 10 |

### Score Adjustment Table

Based on individual student’s average review score, the student’s mark would be adjusted based on the table below.

|  |  |
| --- | --- |
| % Difference from Group Avg | Score Adjustment |
| Over 50% above/below | Instructor discretion |
| 40–50% above | +10% |
| 21–40% above | +8% |
| 11–20% above | +5% |
| ±10% | No change |
| 11–20% below | –10% |
| 21–40% below | –20% |
| 40–50% below | –30% |

# Part 2 - Requirement Analysis

## Functional Requirements (FR)

The following section provides the Functional Requirements which define the essential features and behaviours of the Tetris game as demonstrated in the Milestone 1 scope. They specify the expected interactions between the user and the system, covering all main components:

* Splash Screen
* Main Menu
* Configuration Settings
* High Score Display
* Core Gameplay Functionality

Each requirement is expressed in terms of precondition, event, and post condition formatting to maintain clarity, testability, and alignment with the milestone demonstration.

**Functional Requirements (FR)**

**FR-01 Splash Screen**

*Description:* Displays a brief introductory screen with group and course details before the main menu.

*Precondition*: App not yet on Main Screen.

*Event:* User launches the application.

*Postcondition:* A cantered splash window shows group identity and course info for a few seconds, then closes to the Main Screen.

**FR-02 Main Screen Layout**

*Description:* Provides the main navigation hub with all core menu options.

*Precondition:* App has launched, or user has returned from another screen.

*Event:* Main Screen is displayed.

*Postcondition:* Main Screen shows four buttons: Play, Configuration, High Scores, Exit.

**FR-03 Play Navigation**

*Description:* Starts a new game session from the main menu.

*Precondition:* Main Screen visible.

*Event:* User clicks Play.

*Postcondition:* Game Screen opens with a new game initialised.

**FR-04 Configuration Navigation**

*Description:* Opens the settings interface to adjust gameplay options.

*Precondition:* Main Screen visible.

*Event:* User clicks Configuration.

*Postcondition:* Configuration Screen opens.

**FR-05 High Scores Navigation**

*Description:* Opens the leaderboard screen showing top results.

*Precondition:* Main Screen visible.

*Event:* User clicks High Scores.

*Postcondition:* High Score Screen opens showing a Top 10 list (dummy data acceptable in Milestone 1).  
  
**FR-06 Return Main Menu**

*Description:* Allows user to return to the main menu from any other screen.

*Precondition:* User is on any screen other than the Main Screen (e.g., High Scores, Configuration, Game Over).

*Event:* User clicks the Back button or selects the main menu option.

*Postcondition:* Main Screen opens, displaying all primary navigation options (Play, Configuration, High Scores, Exit).

**FR-07 Exit Navigation**

*Description:* Initiates the process to close the application.

*Precondition:* Main Screen visible.

*Event:* User clicks Exit.

*Postcondition:* Exit confirmation dialog appears with Yes and No.

**FR-08 Exit Confirm – Yes**

*Description:* Confirms and closes the application.

*Precondition:* Exit dialog visible.

*Event:* User selects Yes.

*Postcondition:* Application terminates.

**FR-09 Exit Confirm – No**

*Description:* Cancels exit request and returns to main menu.

*Precondition:* Exit dialog visible.

*Event:* User selects No.

*Postcondition:* Dialog closes; Main Screen remains.

**Configuration Screen (per provided UI)**

**FR-10 Config Controls Respond**

*Description:* Ensures all sliders and checkboxes update their display state immediately when interacted with.

*Precondition:* Configuration Screen visible.

*Event:* User interacts with controls.

*Postcondition:*

* Field Width (cells): Slider/selector updates value label live.
* Field Height (cells): Slider/selector updates value label live.
* Game Level: Slider/selector updates value label live.
* Music (On/Off): Checkbox toggles state and label (On/Off).
* Sound Effect (On/Off): Checkbox toggles state and label (On/Off).
* AI Play (On/Off): Checkbox toggles state and label (On/Off).
* Extend Mode (On/Off): Checkbox toggles state and label (On/Off).

*\* No gameplay binding required at Milestone 1; interaction must be visible and correct.*

**Gameplay (Milestone-1 scope)**

**FR-11 Game Field Setup**

*Description:* Displays the playfield and spawns the first tetromino.

*Precondition:* User started a game (FR-03).

*Event:* Game Screen loads.

*Postcondition:* A 10×20 playfield is rendered; an active tetromino spawns at the top; colours/shapes render consistently.

**FR-12 Movement & Rotation Controls**

*Description:* Enables player to control the active tetromino.

*Precondition*: Game running and not paused.

*Event:* User presses arrow keys (Left/Right to move; Up or designated key to rotate; Down for soft drop if implemented).

*Postcondition:* Tetromino moves/rotates smoothly within bounds and without overlapping fixed blocks.

**FR-13 Gravity & Lock**

*Description:* Moves tetrominoes downward automatically and fixes them in place when resting

*Precondition:* Game running.

*Event:* Gravity tick or piece reaches rest (no valid downward move).

*Postcondition:* Piece locks into the field; next tetromino spawns unless game over.

**FR-14 Clear Full Rows**

*Description:* Removes any fully filled horizontal lines.

*Precondition:* A piece has just locked.

*Event:* System checks board state.

*Postcondition:* All complete rows are removed in one operation; rows above drop; colours remain correct; multiple-row clears are handled.

**FR-15 Pause / Resume**

*Description:* Temporarily halts and resumes gameplay.

*Precondition*: Game running.

*Event:* User presses P.

*Postcondition:*

* If running → game enters Paused state and shows a visible “Paused” message.
* If paused → game resumes and message is cleared.

**FR-16 Game Over**

*Description:* Detects when no further pieces can spawn.

*Precondition:* Game running.

*Event:* New tetromino cannot spawn due to blocked spawn cells.

*Postcondition:* Game ends; “Game Over” state/message shown with a way to return to the Main Screen.

## Non-Functional Requirements (NFR)

This section describes the quality attributes and operational constraints of the system rather than the specific behaviours it performs. These define how the system should operate under certain conditions and ensure it meets user expectations for usability, reliability, performance, and supportability. One NFR for each of the **URPS** categories in the FURPS+ model has been provided, each of which can be implemented using basic Java/JavaFX features.

**NFR-U01 Colour Blindness Support (Usability)**  
*Requirement:* The Configuration screen must include a “Colour Blindness” setting allowing the user to select a type (e.g., Deuteranopia, Protanopia, Tritanopia). The game will replace affected colours with an alternative colour for better visibility.  
*Verification:* Select each type in Configuration, start a game, and confirm colours are replaced according to the chosen setting.  
  
**NFR-R01 Game Stability (Reliability)**

*Requirement:* The application must run for at least 20 minutes, navigating between screens and playing without crashing or showing error messages.

*Verification:* Manually run the application in a lab test for 20 minutes; pass if no crashes or unhandled exceptions occur.

**NFR-P01 Smooth Gameplay (Performance)**

*Requirement:* Gameplay must maintain at least 30 frames per second (FPS) on standard lab hardware while pieces move and rotate.

*Verification:* Play for 5 minutes; visually confirm smooth piece movement with no noticeable lag.

**NFR-S01 Easy Build and Run (Supportability)**

*Requirement:* The game must compile and run from a clean clone using Java 21 and JavaFX, with no manual code changes required.

*Verification:* Freshly clone the GitHub repository, build, and run using the provided instructions; pass if successful without errors.

## Use Case Diagram

A diagram of a computer game

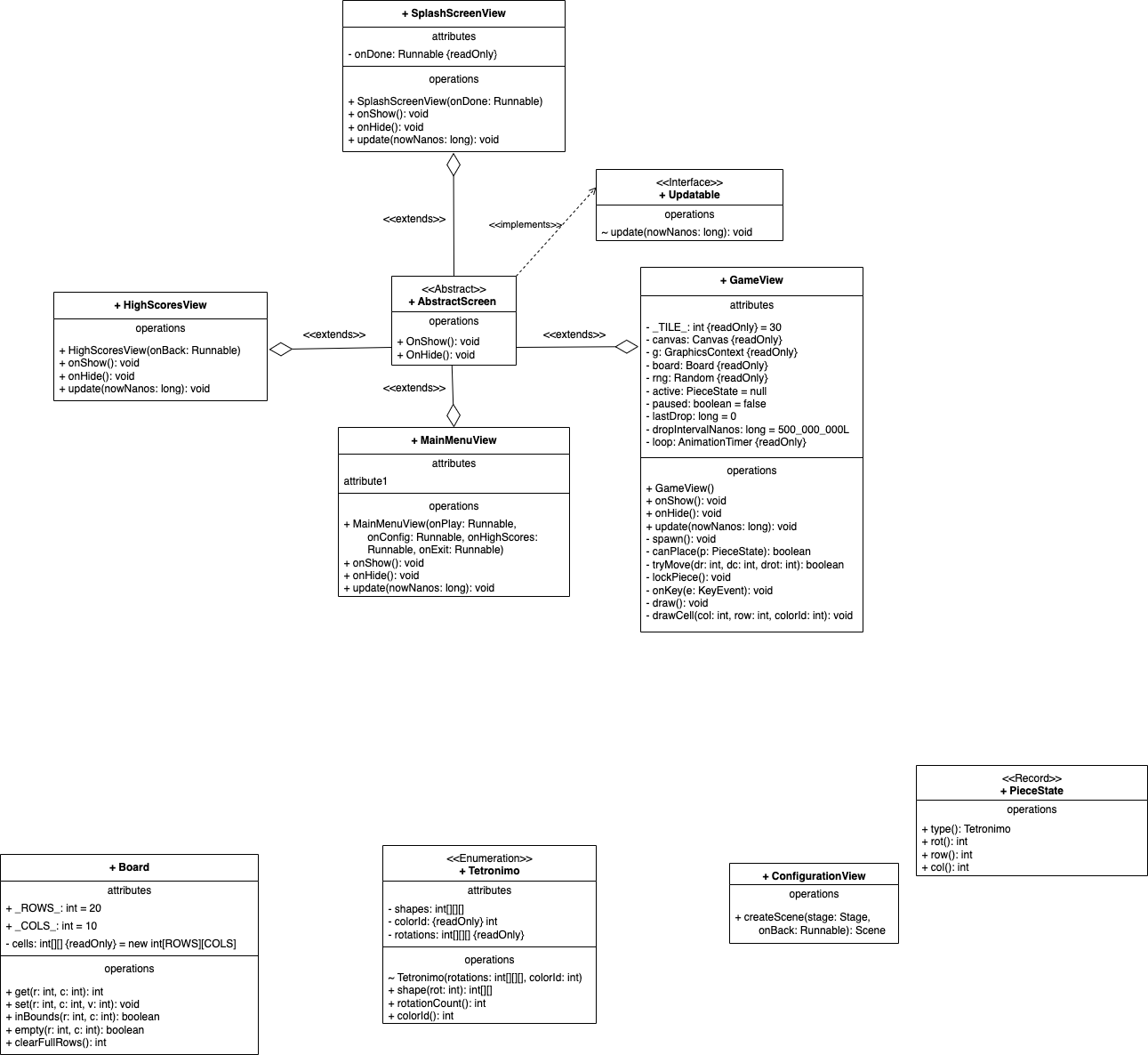
AI-generated content may be incorrect.

## Activity Diagram

A diagram of a computer game

AI-generated content may be incorrect.

## UML Class Diagrams

**Diagram 1: AbstractScreen and its related classes**

# 

**Diagram 2: Board Class**

A screen shot of a computer

AI-generated content may be incorrect.

**Diagram 3: Tetronimo Class**

A black screen with white text

AI-generated content may be incorrect.

**Diagram 4: ConfigurationView Class**

A black background with white text

AI-generated content may be incorrect.

**Diagram 5: PieceState Class**

A black and white text on a black background

AI-generated content may be incorrect.

# Part 3 - Implementation

## GitHub

[Provide the link to your GitHub repository and confirm the lab teacher has access.]

**GitHub Link:**

**Screenshot of Commit History:** (Insert image here)

## GitHub Advanced

**Screenshot of Pull Requests:** (Insert image here)

**Screenshot of code review:** (Insert image here)

**Screenshot of tags:** (Insert image here)

## File Structure

Provide screenshot of GitHub repository structure (IntelliJ with Maven).

## Coding Snippets

Provide code snippets (with explanation) demonstrating:

* JavaFX
* Enhanced for loop
* Enhanced switch
* Interface usage
* Abstract class usage
* Record type usage

# Part 4 - Demonstration

## Demo Video Link

Provide the link to your demo video hosted on a public platform (e.g., YouTube).

**Video Link:**