USE CASE DESCRIPTION/DIAGRAM

Critical notes:

- Missing Authentication use case diagram.
    - They may have merged it with GUI (there was no communication on whether it was merged or not)
    - No system frames on any use case diagrams.

GUI Critics:

Diagram

- Database should not be a use case but an actor in the diagram since the use case descriptions use it as a secondary actor. (Inconsistency between diagram and description)
- Sign-up and log-in should not be linked since they are independent use cases that are linked to the player.
- Missing relationship between “Forgot password” and “Database”, and “Leaderboard” and “Database”.
    - “Forgot Password” should include “Database” as it **must** retrieve the password from the database if the password is forgotten.
    - “Leaderboard” should include “Database” as it retrieves the player ranks to display on the leaderboard.
- Diagram is vague as it doesn’t illustrate all GUI interactions. Consider adding:
    - Winning page, Losing page, Tie Page
- If profiles retrieve data from the database, then the profile should have an include arrow pointing to the database.
- Login page confusing since there's no guide to how players can view settings Settings, Profile, Settings, Games, and Leaderboard. A homepage between Login should be included to include all other game access
- “Logout” should extend to “Settings” as the logout option can only appear when the player selects settings.
- The GUI page needs to be more detailed in terms of what occurs after the game result, what happens after quitting, and more details should be included regarding Profile as mentioned on the assignment.
- Arrow from “Login” to “Forgot Password” should extend the other way as the forgot password option should only appear after the player selects login.
- Arrow from “Play a Game” to “Quit Game” should extend the other way as the quit option only appears once the player plays a game.
- Extends arrows should point from “Settings”, “Profiles”, “Games”, “Leaderboard” to Homepage.
- Extends arrow show point from “Play a Game” to “Games”, because the game options should only be present when the player selects “Games”.

- "Sign up" should extend to "Login" as the signup option would only be present once player is on login page

Descriptions

- The arrows in the gui use case diagram for extends go the wrong way
- The diagram doesn't have a use case description for database and it should've been an actor if anything
- Play game and chat shouldn't be include it should be extends

Leveling Critics:

Diagram

- Diagram doesn't illustrate how the leveling system is reflected in the database.
- Arrows from player to "Award Point For Win", "Award Point for Winning Quick Game", and "Award points for Long Game" should be removed as it seems repetitive and can be simplified to having one arrow form "Award Experience Points Base" to imply the player received total points.
- Arrows from player to "Award Experience Points Base" and "Increase Level" should be the other way around to imply that the user achieved these.
- Missing post conditions for all use case description

Recommendation Aside from Diagram Suggestion: Have one base case as "Award Experience Points", from which extending from that would be "Award Points for Completing a Game", "Award Points for Long Game", and "Award Points for Winning a game". From "Award Points for Winning a Game" extends "Award Points for Winning a Quick Game".

Matchmaking Critics:

Descriptions

- "Matchmaking Player Space and Experience Level"
    - If the player is the one initiating the matchmaking, the player should ideally be the primary actor.
        - Secondary actor would be a game platform as it assists the matchmaking for the player.
- "Adjust Matchmaking Based on Session Intensity"
    - If the player is the one initiating the matchmaking, the player should ideally be the primary actor.
        - Secondary actor would be a game platform as it assists the matchmaking for the player.
- "Handling Matchmaking for Highly Skilled Player"
    - If the player is the one initiating the matchmaking, the player should ideally be the primary actor.
        - Secondary actor would be a game platform as it assists the matchmaking for the player
- Missing post conditions for all use case descriptions

In General (Descriptions):

- All descriptions in matchmaking should have the player as the primary actor and game platform as secondary.
- Many cases seem repetitive and complex when they can be simplified.
- Three different diagrams were unneeded, could have been merged into one since information is redundant and repeating

Diagram

- No use case description for “Alternative Matchmaking Strategy”
- Some cases are not independent, so lines should be extend relationships and not solid lines
- There should be a use case to calculate the session intensity and the players level in order to be able to adjust matchmaking based on session intensity.
- Since “Match Layers Based on Experience”, "Handle Matchmaking for Highly Skilled Players” and “Adjust Matchmaking Based on Session Intensity” are used to calculate matchmaking, they should be dependencies of a main matchmaking use case.
    - Player should be pointing towards the main matchmaking use case.
- Why is the use case “Handle Matchmaking for Highly Skilled Players” targeting highly skilled players only? What is the requirement or design for the other skilled players?
- “Handle Matchmaking for Highly Skilled Players” should not be linked to the game platform, but extends the match players based on experience level

In General (Diagrams):

- Poor visual hierarchy, diagram overall is very vague and should include more details to ensure clarity.
- Use cases can be broken down into simpler use cases
    - Ex. match players based on experience level should've broken down into smaller use cases so they can include players skills and intensity rather than having them as independent cases.

Statistics

- This diagram merges Matchmaking and Leveling without any combined use case descriptions for the two
- As a positive note, there was some improvements made between the two diagrams for extends
- The case for “increase level” should extend “award experience points base” and not the game platform since level changes are influenced by the points than the platform itself

Game Logic Critics:

Descriptions

- Moving a checker and dropping a checker's exceptions should focus on issues related to moves, not general game-state problems.
- Missing use case description for Quit, check for win

Diagram

- No actor for the game host server (Initiating a game, dropping a checker, moving a checker, promoting to a king, play a card, draw a card, ending a game), end user, game logic server (Game Data Fetch by client-side Computer), but is mentioned as a secondary actor in the descriptions
- Connect four, Whist, Checkers moves for drop checker, play a card, or moving a checker might need to be an includes since initiating a game means you must have a game to start
- Quit should not only depend on checking for a win, instead, change Quit to something similar to ending a game to match the use case better and make the logic of the diagram make more sense
- Check for win should have an includes to the three games' moves, since every move you make should check if it is a winning move

CLASS STRUCTURE DIAGRAM

Leaderboard

- Different associations for each section of the diagram, some use <<uses>> some use extends, some do not use anything at all, and some are not linked at all -> there should be consistency between the diagrams
- Some classes are repeated but not linked to similar class structures
- "Calls" -> can use extends instead
- If methods in "Game Session" take userAccount as a parameter, it should be referenced through an association relationship where userAccount is created.
- Game stats and game record should be linked through dependency since game stats depend on game record.
- Game stats should have methods to update game wins and game losses.
- Game IO Handler and Game session though dependency as game session since it i shandling game related inputs and outputs.
- Players should be linked to the following:
    - Since the player has an account ID, name and game level it should be connected to userAccount as it tracks it in user account
    - Since the game stats and game record depend on the players information, they should be linked
    - Since game session required active players, players should be linked to game session
    - Player should associate GamePiece as GamePiece is used as a field in Player
    - Game should associate player as it uses player as a field

Game Logic

- Game should associate GameType as game uses GameType as a field.
    - Ensure naming conventions are consistent throughout diagram

- GamePiece should be associated with PieceType as PieceType is a field in GamePiece.
- ConnectGrid should be linked to Checker as ConnectGrid uses Checker as a field.
- checkerBoard should be linked to checker as checkerboard has checker as a field
- Checkerboard should be linked to checker game as checkers game uses checker board as a field.
- No methods on left part (probably game logic) of the structure diagram.
- Duplicated database stub with no link for one of them
- Why is checkers using the same ConnectGrid as ConnectFourGame when checkersGame already has a checkerboard???
    - Suggestion: hace one class that creates the board and takes in dimensions to create the board depending on the game board.
- Whistgame and stage type should have been linked to stageType as stage type is a field in whistGame.

In General:

- Many fields were used in other classes without referencing where the field originated from
    - Many association arrows were missing
- Overlap on certain classes such as having two classes to create both boards for checkers and connect four.

GUI (Assumed)

- Play a game should reference player and game domain classes because the methods in Play a game depend on these classes.
- Settings should be linked to audio, video, and control settings because settings have methods that depend on audio, video, and control settings.
- Game chat doesn't reference a game session so it's unclear how it interacts with other classes.
- Profiles should include more detail about different actions such as challenging a profile option, searching for profile option.

Matchmaking

- Diagram was well done with multiplicity and order. Diagram was consistent and was logically appropriate, and included correct extends, and calls requirements.

CORE SYSTEM DESIGN

Mockup

- Mockups were missing required game in game library and many games were not mentioned in their diagrams
- Order of records are unclear