

**Group 2: 6+10+3+10+15+11+15=70**

1. Vision Document: 6

Good document Overall.

Introduction should be brief, no need to describe the game features here.

Problem statement and product features should be separate sections. Problem statement should be brief, mention on the goal of the project and brief list of most important features. Product features should be detailed and lists all the features of the system to be developed.

Key stakeholders and interests have been identified correctly.

No risks are listed.

2. Actors: 10

“make moves” covers many functionalities such as “move pawn”, “add a fence”, ... etc. it could be split up to show these events.

3. Brief Use Cases: 3

Brief use cases descriptions style is mostly not followed.

The use case description should usually describe a flow of user and system actions, forming “action-reaction” pairs.

Each use case should cover a single flow of events in the main success scenario, with no conditions or branching, and each step should cover a single (or perhaps series of small, connected) action(s).

“Load game” is very brief, here’s how Load a Game could have been done:

“Load a Game: the user elects to load the previously saved game, the system retrieves the saved game from persistent storage, sets it up as the current game and informs the user that the load has been successful”.

“Place wall” , “Move Pawn” are not use case descriptions, but rather a list of rules for placing walls and moving pawns.

“Toggle color...” contains a use case description but is not written essential style and includes UI details “select a check box”, it also

includes many irrelevant details, it should focus only on user-systems actions.

“Set AI Difficulty” also includes unnecessary details, should start from “the player selects” the remaining description is good.

“Save game” should be written more clearly; “the player *saves the game, ...*”

#### 4. Setup a Game UC: 10

Stakeholders Interests are not clear. Should not include “wants to choose best moves and win” – think: it’s not something that the system can provide, and is not part of setting up a game! This section should focus on player interests with regards to game functionalities, for example:

- wants to choose the number of human and computer players
- wants to set the difficulty of computer players
- wants the ability to see a list of game rules
- wants the ability to load a saved game

Preconditions are not relevant; we should stick to things that must have happened within our system, so for this use case, and given your interpretation of its scope, there is no need to state a precondition.

Success Guarantee is very brief, should include for example:

- Game is initialized or loaded.
- The type of players is set.
- Difficulty for each computer opponent is set.
- Turn order is generated.
- Starting positions are initialized.
- Board is displayed

The Main Success Scenario misses many steps and over-summarizes in many places, for example no options for choosing player types or difficulty are described, these should be described in detail within the main success scenario.

It should not include UI elements such as “Dialog box”, should be written in essential style, such as “option”.

Alternative flows and the Exception are chosen appropriately but are not described well, they should be described step by step for each player and system actions for example:

User quits game (Can be done at any point during game initialization while the UI is user active.)

1. System presents option for user to confirm closing the game.
2. User confirms.
3. System closes game window and exits the program. [Use case ends]

#### 5. Take a Turn UC: 15

Stakeholders Interests could be expanded, player wants to be able to move a pawn, add a fence, see hints of available legal moves...

The Main Success Scenario over-summarizes.

Alternative flows and the Exception are mostly chosen appropriately but are not described well, they should be described step by step for each player and system actions.

#### 6. UI sketches: 11

Clear UI, capture some functionalities, but some are missing such as choosing number of players and difficulties in game setup, and walls inside board in taking a turn.

#### 7. Domain Model: 15

Very good for first iteration, captures most key concepts and associations.