Dumb Dino, Inc.

Cuadrado
Use-Case Specifications

Version <3.0>

Revision History

Date	Version	Description	Author
01/Oct/2022	1.0	First draft	Devin Setiawan John Zheng
09/Oct/2022	2.0	Second Draft	John Zheng
21/Nov/2022	3.0	Third Draft	Nikhil Singla

Table of Contents

Use-Case Specifications

1. Use-Case Model

1.1 Introduction

The configs and assets of the game are local to the device so it can be played anywhere. The users of the program will be a player. The player plays the game through a graphical interface.

1.2 General Actors Descriptions

1.2.1 Player - The user(s) that are interacting with the program and participating in the game.

1.3 Use-Case Model Hierarchy

1.3.1 *Player Activity*

Description

This package contains all functionalities that a player can do.

Use Cases

1. Play
2. Settings
3. Time Attack

Actors

1. Player

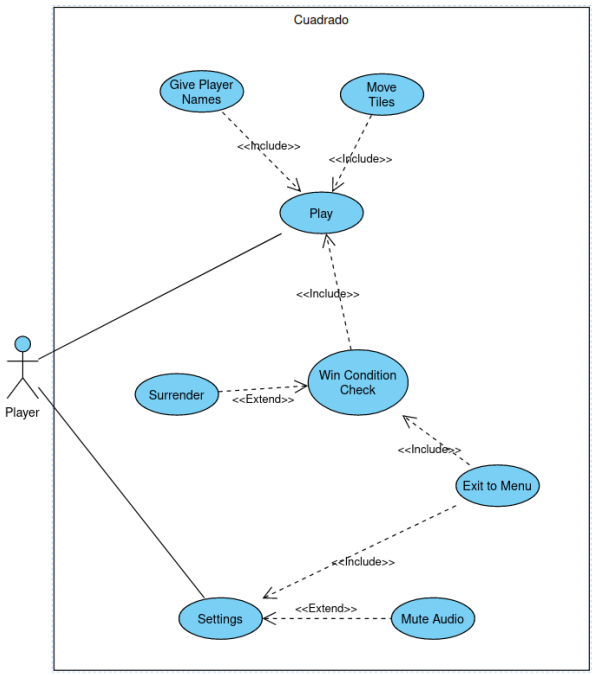
RelationShips

None

Packages Owned

None

1.4 Diagrams of the Use-Case Model



2. Play

2.1 Brief Description

The Play use case allows player to play the game. Players are given the ability to do various controls to achieve the winning condition which are calculated from points earned by the program. Player can complete the goal of attaining a winning status after the play sequence ended.

2.2 Flow of Events

2.2.1 Basic Flow

1. Player enter player's name
2. Program present a board with a new set of tiles
3. Player rolls the dice
4. Player moves the tiles
5. Program check if any board present are solved
6. Program add point to player
7. Program check for the completion of all objectives
8. Program finds a winner by tallying up the points
9. Player wins

2.2.2 Alternative Flows

2.2.2.1 (3a) Player waits for a roll

Given that it is not a player's turn, dice rolls are instead done by the other player in the game. The player in this given use case waits for a roll instead of rolling a number.

2.2.2.2 (5a) Unsolved board

In the case of no board solved, the program will not go to the next flow.

2.2.2.3 (6a) Point addition not given to player

Player does not have the round win status. Points are not given to the player.

2.2.2.4 (7a) Objectives unfinished

Objectives still exist on the board. Program goes back to flow number 4.

2.2.2.5 (9a) Player Lose

Player does not have the game win status. A winning status won't be given to the player

2.3 Preconditions

2.3.1 Game not started

2.4 Postconditions

2.4.1 Player wins the game

2.4.2 Player lose the game

2.4.3 Board reset

2.5 Extension Points

2.5.1 Give Player Names

Point of extension: (1)

2.5.2 Move Tiles

Point of extension: (4)

2.5.3 Win Condition Check

Point of extension: (5), (7), (8)

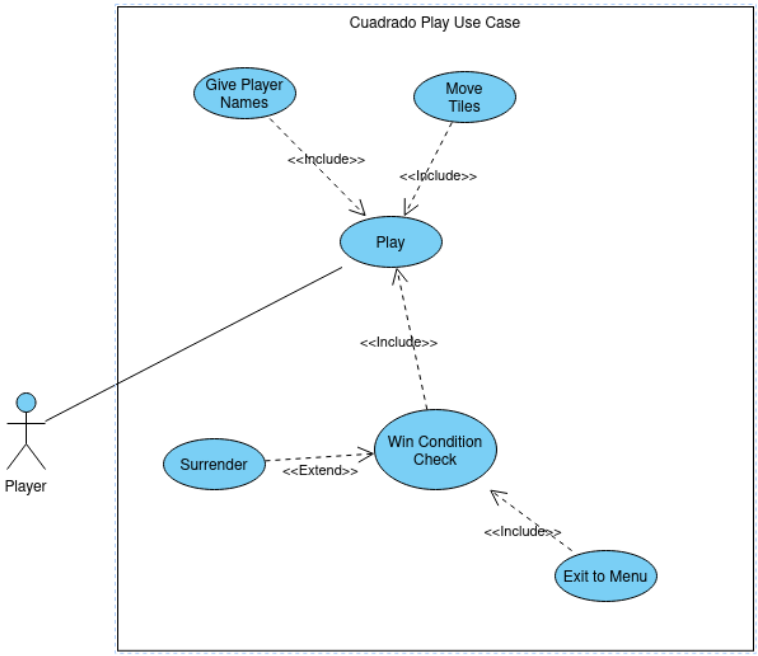
2.6 Relationships

'Player Name Handler' use case is included to the 'Play' use case to handle the retrieval and storing of player's name.

'Move Pieces' use case is extended into the 'Play' use case to give the functionality of moving pieces on the board.

'Win Condition Check' use case is included to the 'Play' use case to check winning condition for each round and also the final winning condition for the game

2.7 Use-Case Diagrams



NEW

3. Settings

3.1 Brief Description

The Settings use case allows the player to change aspects of the program to their preferences.

3.2 Flow of Events

3.2.1 Basic Flow

1. Player selects settings button on screen
2. Player is presented a menu of options from program
3. Player changes options
4. Player exits back to main menu

3.2.2 Alternative Flow

3.2.2.1 (3a) Player does not change any options

Player decides to enter the settings menu and not edit any values. The program will not change behaviors.

3.3 Preconditions

3.3.1 Game behavior is unchanged

3.4 Postconditions

3.4.1 Game behavior is changed

3.5 Extension Points

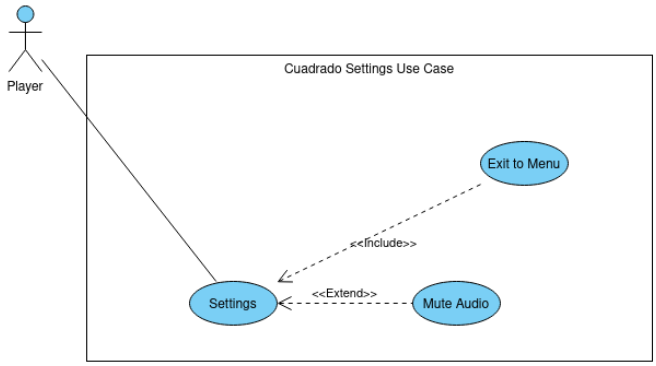
3.5.1 Mute Audio

Point of extension: (1)

3.6 Relationships

'Mute Audio' use case is included in the 'Settings' use case to handle an optional audio mute in the settings menu.

3.7 Use-Case Diagrams



NEW

4. Time Attack

4.1 Brief Description

The Time Attack use case allows the player to play the game in a Singleplayer mode under a time constraint. Players are given the ability to do various controls to achieve the winning condition which are calculated from points earned by the program. Players can complete the goal of attaining a winning status after the play sequence ends.

4.2 Flow of Events

4.2.1 Basic Flow

1. Player enter player's name
2. Program present a board with a new set of tiles
3. Player moves the tiles
4. Program check if any board present are solved
5. Program add point to player
6. Program checks if the time ran out, if not, loop to step 2
7. After time ends, program displays the tally of the points
8. Players can write their scores to compare

4.2.2 Alternative Flows

4.2.2.1 (5a) Unsolved board

In the case of no board solved, the program will not go to the next flow.

4.2.2.2 (6a) Point addition not given to player

Player does not have the round win status. Points are not given to the player.

4.2.2.3 (7a) Objectives unfinished

If time doesn't end, program rolls the die again to continue the game

4.2.2.4 (7a) Player goes back to main menu

Instead of retrying after game end, players can also go back to main menu

4.3 Preconditions

4.3.1 Game not started

4.4 Postconditions

4.4.1 Player times out the

4.4.2 Board reset

4.5 Extension Points

4.5.1 Give Player Names

Point of extension: (1)

4.5.2 Move Tiles

Point of extension: (4)

4.5.3 Win Condition Check

Point of extension: (5), (7), (8)

4.6 Relationships

'Player Name Handler' use case is included to the 'Play' use case to handle the retrieval and storing of player's name.

'Move Pieces' use case is extended into the 'Play' use case to give the functionality of moving pieces on

the board.

'Win Condition Check' use case is included to the 'Play' use case to check winning condition for each round and also the final winning condition for the game