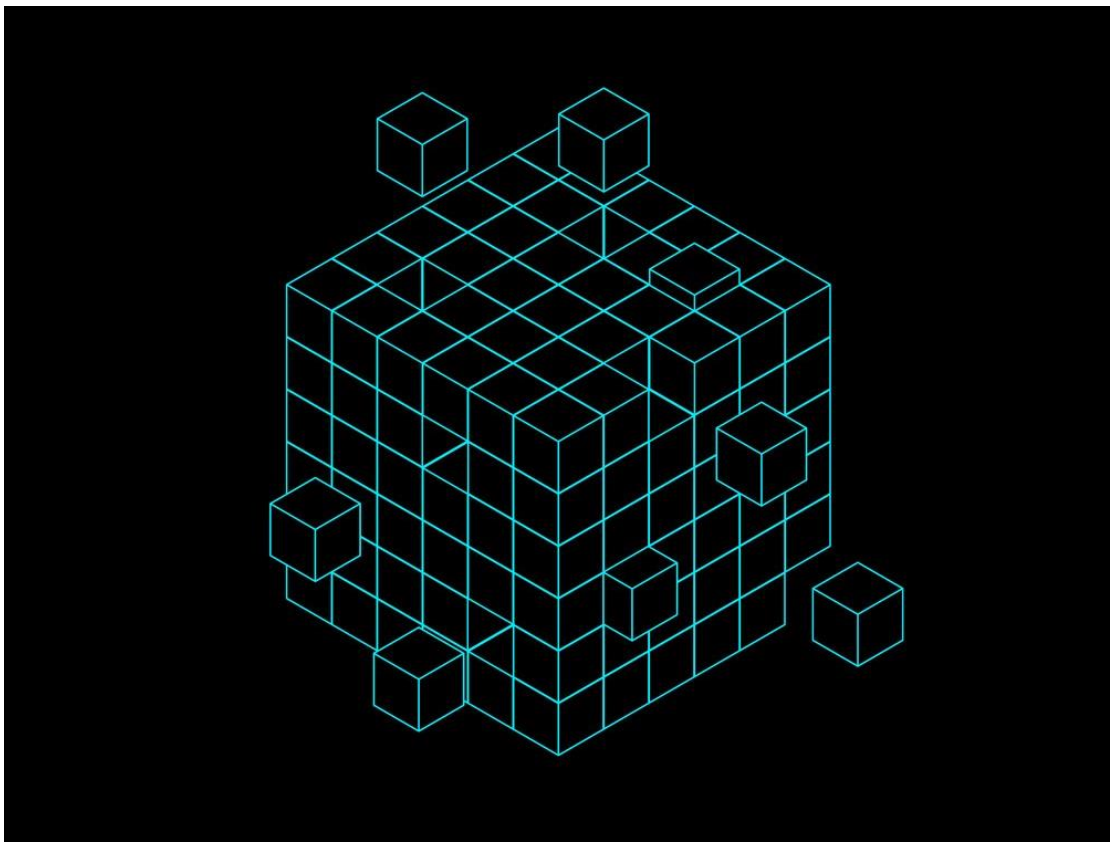
Face 5

Project Report



by

Alyssa Mihok, Ivana Pavlovic, and Jazbel Lopez

for CS 440

at the

University of Illinois Chicago

September 2020

Table of Contents

List of Figures 5

I Project Description 6

1 Project Overview 6

2 The Purpose of the Project 6

2a The User Business or Background of the Project Effort 6

2b Goals of the Project 6

2c Measurement 7

3 The Scope of the Work 7

3a The Current Situation 7

3b The Context of the Work 7

3c Work Partitioning 8

4 The Scope of the Product 8

4a Scenario Diagram(s) 8

4b Product Scenario List 9

4c Individual Product Scenarios 9

5 Stakeholders 10

5a The Client 10

5b The Customer 10

5c Hands-On Users of the Product 10

5d Maintenance Users and Service Technicians 10

5e Other Stakeholders 10

5f User Participation 11

5g Priorities Assigned to Users 11

6 Mandated Constraints 12

6a Solution Constraints 12

6b Implementation Environment of the Current System 12

6c Partner or Collaborative Applications 12

6d Off-the-Shelf Software 12

6e Anticipated Workplace Environment 12

6f Schedule Constraints 13

6g Budget Constraints 13

7 Naming Conventions and Definitions 13

7a Definitions of Key Terms 13

7b UML and Other Notation Used in This Document 14

7c Data Dictionary for Any Included Models 14

8 Relevant Facts and Assumptions 14

8a Facts 14

8b Assumptions 14

II Requirements 14

9 Product Use Cases 14

9a Use Case Diagrams 14

9b Product Use Case List 15

9c Individual Product Use Cases 15

10 Functional Requirements 17

11 Data Requirements 19

12 Performance Requirements 20

12a Speed and Latency Requirements 20

12b Precision or Accuracy Requirements 20

12c Capacity Requirements 21

13 Dependability Requirements 21

13a Reliability Requirements 21

13b Availability Requirements 21

13c Robustness or Fault-Tolerance Requirements 22

13d Safety-Critical Requirements 22

14 Maintainability and Supportability Requirements 23

14a Maintenance Requirements 23

14b Supportability Requirements 23

14c Adaptability Requirements 23

14d Scalability or Extensibility Requirements 24

14e Longevity Requirements 24

15 Security Requirements 24

15a Access Requirements 25

15b Integrity Requirements 25

15c Privacy Requirements 25

15d Audit Requirements 26

15e Immunity Requirements 26

16 Usability and Humanity Requirements 26

16a Ease of Use Requirements 26

16b Personalization and Internationalization Requirements 28

16c Learning Requirements 28

16d Understandability and Politeness Requirements 28

16e Accessibility Requirements 29

16f User Documentation Requirements 29

16g Training Requirements 29

17 Look and Feel Requirements 29

17a Appearance Requirements 29

17b Style Requirements 30

18 Operational and Environmental Requirements 30

18a Expected Physical Environment 30
18b Requirements for Interfacing with Adjacent Systems 31
18c Productization Requirements 31
18d Release Requirements' 31
19 Cultural and Political Requirements 32
19a Cultural Requirements 32
19b Political Requirements 32
20 Legal Requirements 32
20a Compliance Requirements 32
20b Standards Requirements 33
21 Requirements Acceptance Tests 33
21a Requirements - Test Correspondence Summary 33
21b Acceptance Test Descriptions 33

III Design 36
22 Design Goals 36
23 Current System Design 36
24 Proposed System Design 37
24a Initial System Analysis and Class Identification 37
24b Dynamic Modelling of Use-Cases 38
24c Proposed System Architecture 38
24d Initial Subsystem Decomposition 39
25 Additional Design Considerations 39
25a Persistent Data Management 39
25b Boundary Conditions 40
25c User Interface 40
25d Application of Design Patterns 40
26 Final System Design 41
27 Object Design 42
27a Packages 42
27b Subsystem I - User Interface 44
27c Subsystem II - Cube Data 45
27d Subsystem III - Game State Management 45

IV Project Issues 46
28 Open Issues 46
29 Off-the-Shelf Solutions 47
29a Ready-Made Products 47
29b Reusable Components 47
29c Products That Can Be Copied 47
30 New Problems 47
30a Effects on the Installed Systems 47

30b Potential User Problems 47
30c Limitations in the Anticipated Implementation Environment That May Inhibit the New Product 47
31 Migration to the New Product 48
32 Risks 48
33 Costs 48
34 Waiting Room 49
35 Ideas for Solutions 49
36 Project Retrospective 49

V Glossary 49

VI References / Bibliography 50

[5]A. Lau, C. Riviera, M. Searcy "Greedy Dwarf Project Report" 50

VII Index 50

List of Figures

Figure 1 - Work Context Diagram……. *9*

Figure 2 - Scenario Diagram for Average Experience……. *10*

Figure 3- Use Case for Playing Face 5 …….. *15*

Figure 4 - UML class diagram…….. *20*

Figure 5 - Class Design ……... *37*

Figure 6- Sequence Diagrams …… *38*

Figure 7 - Subsystems …….. *39*

Figure 8 - User Interface…….. *40*

Figure 9 - Model-View-Controller……..... *41*

Figure 10 - Class System Design…….. *42*

Figure 11 - Model……..……..... *43*

Figure 12 - View……..…….. *43*

Figure 13 - Controller……..... *44*

Figure 14 - User Interface Subsystem……... *44*

Figure 15 - Cube Data Subsystem…… *45*

Figure 16 - Game State Management……....*46*

I Project Description

1 Project Overview

This game is an extension of connect 4 that allows 4 players. Instead of a 6x7 slot game we will develop 3-Dimensional 12x12x12 cubes. Each player starts on a face of the cube; to win a player must traverse from their starting face to the face across. Before a player can make a move into the cube they first have to connect 5 of their colors on their starting face. During the game the player will take turns placing their color on the board. Once the path is blocked, the player can move in any other direction from as long as it connects to their last piece placed. Whoever makes it across first wins.

2 The Purpose of the Project

2a The User Business or Background of the Project Effort

Hasbro would develop and publish the game; as one of the original publishers of *Connect 4,* they would want to see the development of their classic board game into video game form and sold for profit. Though profit is not the only motivation, the development of this game means a source of entertainment and form of bringing people together. Naturally, the developers will gain experience in video game development. This will hold true for publishers who wish to sell a game and for developers with an interest to create video games.

2b Goals of the Project

The client, being a director of Hasbro or an individual(s), comes up with an idea that they wish to see come to life. They will find the means necessary to do so. The goal of this game is to feel like a board game all while connecting people together.

As the game is being developed, new ideas and features will arise. However we must keep in mind, if we keep adding features and ideas to be implemented it will feel as if the game will never be completed. Perhaps, there are too many features to implement that by the time deadlines roll around the features are not developed to their full potential. To avoid this, a manager of development will be required who will keep in mind the client's wishes and know the potential of their development team to best select which features should be implemented.

For example, we want the users to have a good experience; a hassle free connection between all the players. We want to be able to update the game: fixing bugs, adding new features.

2c Measurement

A way to measure the success of a game is in post-development. Where we see the reviews and feedback that is received once the game is out into the public. Another measure of success would be in-development; having good communication throughout the whole team , weekly meetings where everyone is reminded of the goals of the project so everyone is on the same page.

3 The Scope of the Work

The gaming industry produces games with the intention of selling them to consumers for entertainment purposes in order to make a profit.

3a The Current Situation

After the release of any new game sales tend to slow down or stop altogether. In order to continue making a profit, new games must be designed to drive sales in the gaming industry. As the sales from a previous release begin to slow down ideally there will be a new game ready for release.

3b The Context of the Work

The work is defined as the production of game software for entertainment purposes in order to drive sales in the gaming industry. A separate digital distribution store will be needed to host the software in order for it to be sold on PC, mobile, and console systems. Players will also be needed to install the game and the owner of the device which the software is installed on will need to regularly update the software.

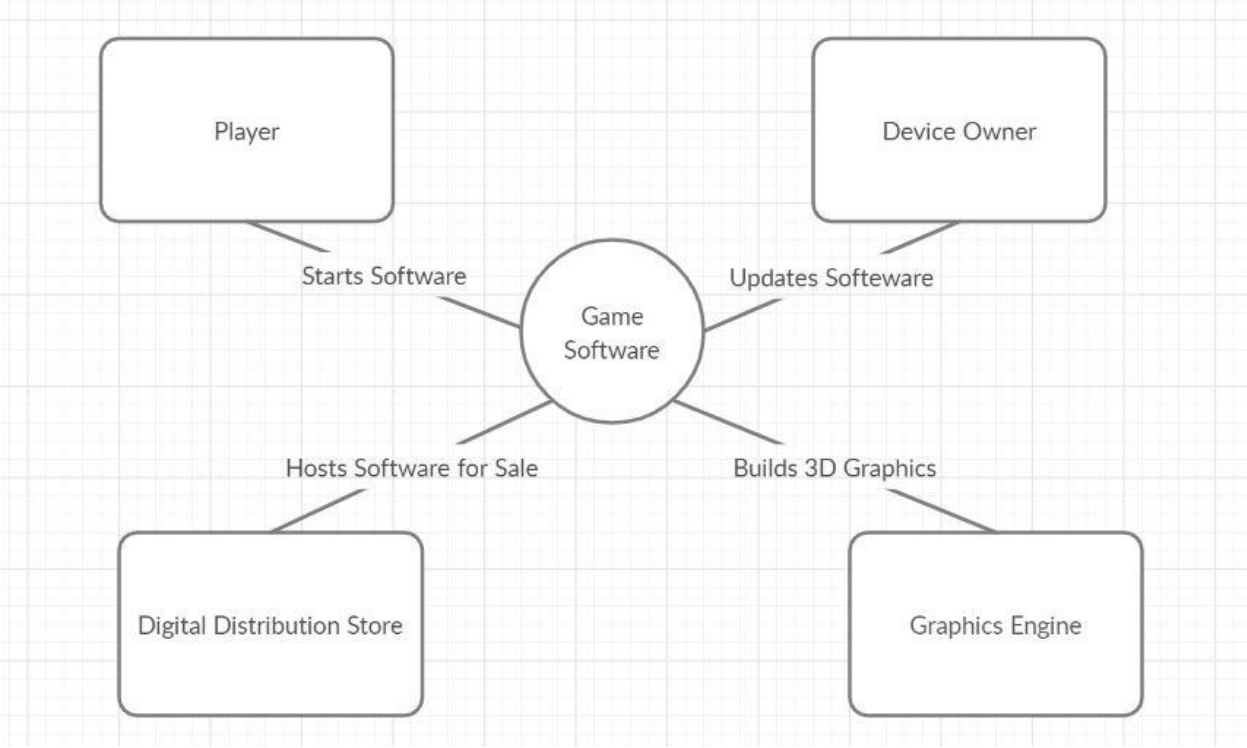


Figure 1 - Work Context Diagram

3c Work Partitioning

In the event of a drop in sales for a gaming company a new game software will be developed in order to drive sales.

3d Competing Products

The gaming industry is a competitive market and there are many other multiplayer games available on digital platforms.

4 The Scope of the Product

The product, being a game, must be able to load executable files, provide instructions, load in graphical elements, keep track of all players' turns and moves, control various bot players, and determine winners and losers once initiated by the player. It is up to the player when to start and stop the game software in order for the game to be played.

4a Scenario Diagram(s)

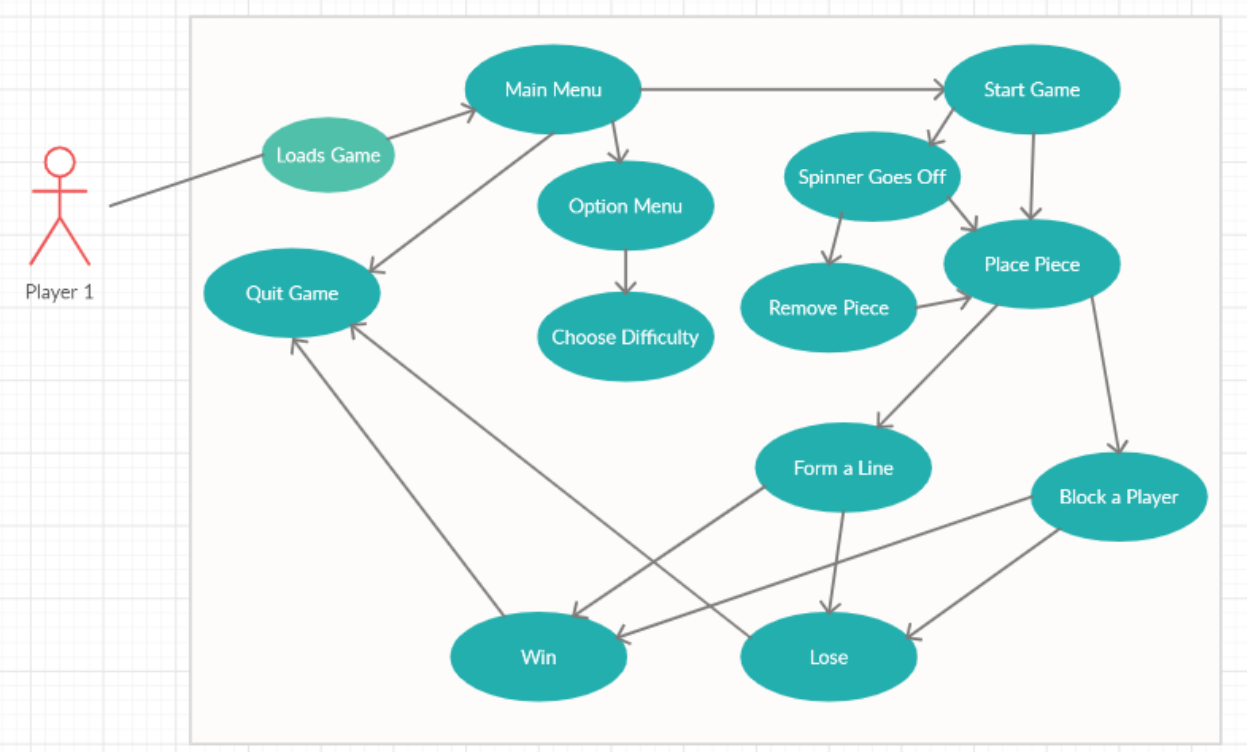


Figure 2 - Scenario Diagram for Average Experience

4b Product Scenario List

- Difficulty Selection
  - Player is brought to the main menu and changes difficulty
- Average Experience
  - Player plays a game against bot players

4c Individual Product Scenarios

Difficulty Selection: Shelby purchases and downloads Face 5 to her computer. When she launches the executable, she is taken to the Main Menu. She has the options from the menu to visit the Difficulty Settings Menu, start a game, or exit the game. She selects the Difficulty Setting menu and changes the difficulty of each of the bot players to Basic then returns to the Main Menu.

Average Experience: Ed purchases and loads Face 5 and is taken to the Main Menu. He has the option to choose to visit the Difficulty Settings Menu or start a game. He visits the Difficulty Setting Menu to change the difficulty of the bot players down to Basic. He then returns to the Main Menu and starts a game.

The cube-shaped board loads into view. Ed and the other computer players are then randomly assigned a number to determine the order of turns. All players take turns placing pieces on the board in the determined order. They create connections and block each other's paths.

After a few turns a Spinner appears on screen with each player's name and Ed is randomly selected using the Spinner. Ed has the option to place two pieces this turn or to place one piece and remove another player's piece. Ed chooses to remove a bot player's piece and then place his own.

After a few more turns Ed successfully creates two, five-piece connections on the cube-shaped board and wins a game. He plays a few more games with the bot players after that, winning and losing a few times before he decides to quit the game.

5 Stakeholders

5a The Client

Publisher - Hasbro

Motivation:

- The game improvement and upgrading it to a more competitive level.

5b The Customer

Publisher - Hasbro

5c Hands-On Users of the Product

Players:

- User demographics - Children, Adolescents, Video game enthusiasts
- User Role - consume the entertainment product, generate profits for the publisher or arcade owner
- Technological Experience – The experience with games may vary.

5d Maintenance Users and Service Technicians

The players would install and update the game on their computers/laptops when needed.

5e Other Stakeholders

Beta Testers:

- Little technical knowledge required
- Not involved in development, only involved in pre-release testing runs

Marketing experts:

- Any marketing company
- No technical knowledge required
- Not involved in development, solely involved in marketing the game to various demographic groups

Legal experts:

- Lawyers
- No technical knowledge
- Little involvement, but will intervene in the event of a legal dispute with another party

5f User Participation

Beta Testers

- Somewhat involved in the final testing process
- Contribution - report of found bugs or lags of the game
- Minimum amount of time - 3 hours

All other stakeholders

- No involvement in development
- Contribution - feedback and possible improvement recommendations

5g Priorities Assigned to Users

Key Users:

● Players - The product's success will be determined by how popular it is with players, thus players are the critical to the continued success of the game

Secondary user:

● The publisher - Hasbro

Unimportant users:

● All others

6 Mandated Constraints

6a Solution Constraints

Description: The product shall be downloaded onto Windows, Linux, or Mac operating systems from the Steam store.

Rationale: The Steam store is the most used gaming store platform.

Fit Criterion: The game will be compatible with Windows 10 and up, iOS 10 and up and Linux 5.0.0 and up.

Description: The product shall have a mobile version available for iOS and Android devices, available through the App Store or Google Play Store.

Rationale: The market for consumers with smartphones is consistently growing

Fit Criterion: The mobile version will be compatible with iOS 9 and up, and compatible with Android 9.0.

6b Implementation Environment of the Current System

The product will run on Windows, Mac iOS, and Linux on PC, as well as mobile platforms iOS and Android.

Motivation: The product must meet all publishing specifications for Steam, the App Store, and the Google Play store. The file size must be manageable for mobile devices

Considerations: The user must have internet to install and update the game, and enough space to install the game properly.

6c Partner or Collaborative Applications

- The product shall interface with Steam store, Google store or Apple Store for updates.

6d Off-the-Shelf Software

- The product shall be developed with Java.

6e Anticipated Workplace Environment

- Target audience is children, adolescents, and game enthusiasts, thus it is expected that the user will play the game in a safe, non-work environment.

6f Schedule Constraints

- The product will be developed according to deadlines and software release stipulations.

6g Budget Constraints

Resources must be allocated for:

- A team of software engineers (x2)
- Game designers (x2)
- Project lead (x1)
- QA / Software testers (x2)
- Full stack web developer for website (x1)
- Lawyer (x1)
- Office space + computers + internet, etc
- Content writer for marketing (x1)
- Server space
- Alpha & beta testing

7 Naming Conventions and Definitions

7a Definitions of Key Terms

Content:

(1) User/Player - Person playing the game and interacting with the software
(2) Hasbro - Hosts the game
(3) Windows, iOS, console - systems where the game will be installed
(4) Project - Face 5 game
(5) Face - The side of the game board facing a player
(6) Piece - a player’s game pieces that they use to advance on the board
(7) Path - the sequence of piece placements that a player has made in the cube

7b UML and Other Notation Used in This Document

- This document generally follows the UML standard found in “UML Distilled” by Fowler et. all.

7c Data Dictionary for Any Included Models

The program will have a few data resources:

- Files containing the information about the layout of the cube (game board).
- A table containing the location of each player’s pieces.

8 Relevant Facts and Assumptions

8a Facts

The game will be designed in a way where it still feels like a classic board game.
The game will be compatible with Windows, iOS, console, and phone applications.
The game will have continual updates; fixing any bugs and adding new features.

8b Assumptions

The user has an interest in video games; that be on mobile phones, consoles or computers.
No dependencies of outside people or actors.

The software will vary depending on the platform being used but the product will look similar across all them. There will be minor differences between platforms to accommodate different controls for computer, mobile or consoles.

II Requirements

9 Product Use Cases

9a Use Case Diagrams

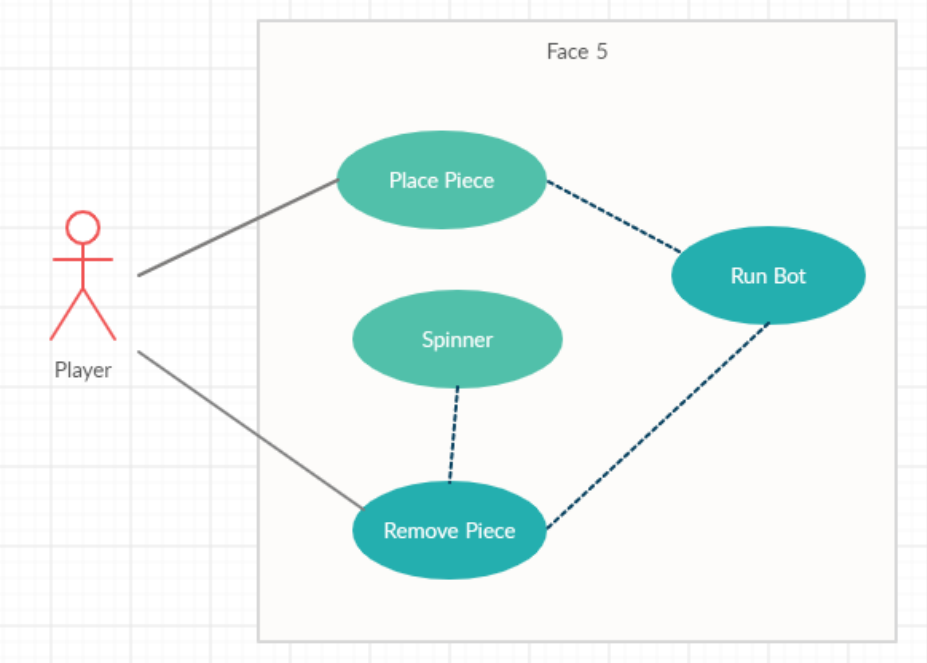


Figure 3 - Use Case for Playing Face 5

9b Product Use Case List

- Placing Pieces
- Remove Piece
- Place Two Pieces

9c Individual Product Use Cases

Use Case ID: 001 Name: Placing Pieces Pre-conditions: game is running, board is loaded and visible Post-conditions: N/A Initiated by: the user Triggering event: User inputs piece placement coordinates Additional actions: N/A
Sequence of Events:

1. User inputs piece placement coordinates
    2. System checks if coordinates are available for piece placement
3. Piece gets added to the visible board
    4. System checks adjacent locations for connections, if win conditions are met then player wins else bots play in response

Alternatives: Remove Piece

Exceptions: if coordinates are unavailable ask player for new input

Use Case ID: 002 Name: Remove Piece

Pre-conditions: game is running, board is loaded and visible, a new round of piece placements is about to begin

Post-conditions: N/A

Initiated by: the game software

Triggering event: random chance activation function returns true

Additional actions: N/A

Sequence of Events:

1. Random chance activation function returns true between rounds
    2. System loads and runs spinner
3. Player is selected by the spinner
    4. System presents Player with options of removing a piece or placing 2 pieces
5. Player selects to remove piece
    6. System loads locations of all opposing bots' pieces
7. Player selects a piece to remove
    8. System removes the piece from the board

Alternatives: Place Two Pieces

Exceptions: N/A

Use Case ID: 003 Name: Place Two Pieces

Pre-conditions: game is running, board is loaded and visible, a new round of piece placements is about to begin

Post-conditions: N/A

Initiated by: the game software Triggering event: random chance activation function returns true Additional actions: N/A
Sequence of Events: 1. Random chance activation function returns true between rounds 2. System loads and runs spinner 3. Player is selected by the spinner 4. System presents Player with options of removing a piece or placing two pieces 5. Player selects to place two pieces 6. System requests for an input coordinate for the piece 7. Player inputs piece placement coordinates 8. System checks if coordinates are available for piece placement 9. Piece gets added to the visible board 10. System checks adjacent locations for connections, if win conditions are met then player wins else bots play in response
Alternatives: Remove Piece Exceptions: if coordinates are unavailable ask player for new input

10 Functional Requirements

10.1 – Game Creation

Description: The system must provide a means for a user to start a new game.

Rationale: As this is a game, a user needs to be available to start a new game.

Fit Criterion: A user is able to start a new game within a minute, including a possible loading period.

Acceptance Tests: 10.1

10.2 – Game Save

Description: The system must provide a means for a user to continue playing a saved game.

Rationale: The system should allow a user to save the game that they can play later.

Fit Criterion: The system should allow a user to save at least one game, while allowing a creation of a new game.

Acceptance Tests: 10.1

10.3 – Game Statistics

Description: The system shall allow a user to see their high score.

Rationale: The high score feature will be used to keep a track of the best score and determine the best player of the game.

Fit Criterion: The user should see an accurate high score a few seconds after the game ended or of the game being saved. The high score should stay consistent, even after the user exited the system.

Acceptance Tests: 10.1

10.4 – Levels, Content, and Gameplay

Description: The system shall allow a user to progress to more advanced levels during the game.

Rationale: The levels will vary in difficulty, the first level will be simple, and how the player progresses, the levels will get more complicated.

Fit Criterion: The system shall have at least 5 different types of levels that uniquely vary in difficulty.

Acceptance Tests: 10.2

10.5 – Controls 1

Description: The system shall allow a user to play using a mouse.

Rationale: This game shall be released onto PC computer, so the game should therefore use the mouse as a primary player control.

Fit Criterion: The system shall respond to the proper control with the proper player functionality.

Acceptance Tests: 10.3

10.6 – Controls 2

Description: The system shall allow a user to play using a mobile device.

Rationale: This game should be released onto mobile platforms and should have player controls consistent with touch enabled smartphones.

Fit Criterion: The system shall respond to the proper control with the proper player functionality.

Acceptance Tests: 10.4

11 Data Requirements

Content: This is a model of the system's game subject matter using the Unified Modeling Language (UML) class model notation.

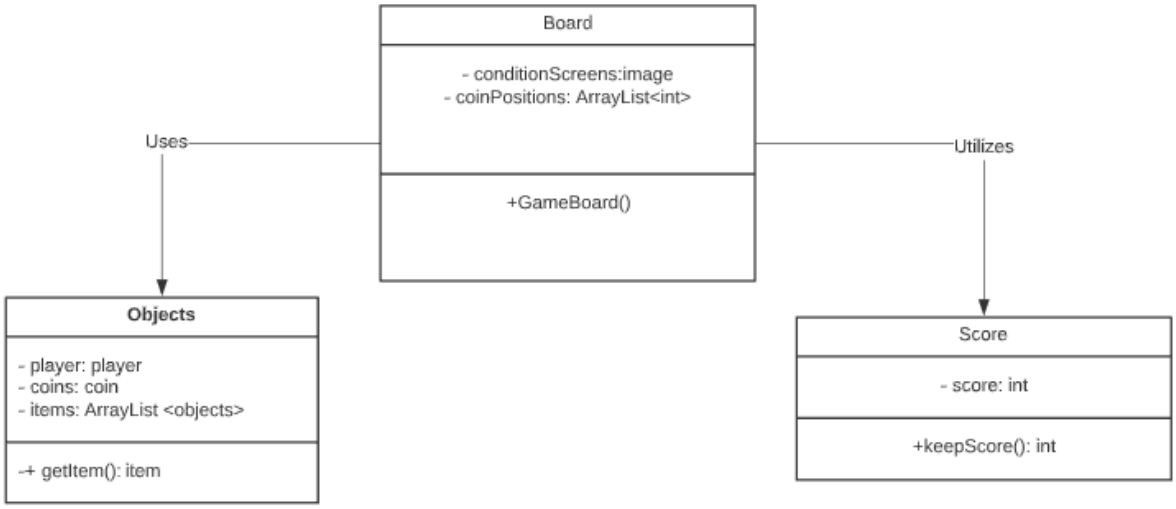


Figure 4 - UML class diagram

12 Performance Requirements

12a Speed and Latency Requirements

12.1 - System Response Time for Individual User

Description: The system response time should be of minimal notice to the user.

Rationale: Users have a few features that will help strategize their next move or any other features that require a response form the system.

Fit Criterion: Systems response time should respond within 2 seconds for 90% of requests and no more than 3.5 seconds.

Acceptance Test: 12.1

12.2 - System Frame Rate

Description: The system will have a frame rate of 60 frames per second

Motivation: The users have a feature of zooming in to strategize for move. The zoom in should be a smooth transition.

Fit Criterion: The system has a tolerance of 57 to 61 frames per second.

Acceptance Test: 12.2

12b Precision or Accuracy Requirements

12.3 - Update Accuracy

Description: The game is synchronized with 4 users. The system should update the user's moves all at the same time within 3 seconds of a new move being made.

Motivation: The game is meant to be a multiplayer game. System response is crucial for other players to get updates promptly to adjust for their move.

Fit Criterion: The system should update the players within 3 seconds for 90% of moves made. No response/update should be more than 4.5 seconds.

Acceptance Test: 12.3

12c Capacity Requirements

12.4 - Data to be stored

Description: The system should be able to store user points, moves, and displays during the game

Motivation: The game will have multiple games 4 user synchronized games. Each game will have stored values for the duration of the game.

Fit Criterion: The system must be able to store up to 1000 simultaneous games.

Acceptance Test: 12.4

13 Dependability Requirements

13a Reliability Requirements

13.1 - Time Between Failures:

Description: This game should not fail more than once a day.

Rationale: System failure can have a negative effect on the player experience but won't have critical or lasting consequences to the player. Failure should still be minimized but is tolerable.

Fit Criterion: Over a one-month period being played five hours a day, the system must not fail more than 28 times.

Acceptance Test: 13.1

13b Availability Requirements

13.2 - Game Availability:

Description: This game should be available for use 24 hours a day 365 days a year. The game should be able to start up again almost immediately after failure.

Rationale: All game processes begin whenever the player chooses to run the game. There should be no limit to the time the player can play the game because it is for entertainment purposes.

Fit Criterion: Game processes must be ready to run at any time of the day and after failure must be able to return to the ready state within 3 seconds.

Acceptance Test: 13.1

13c Robustness or Fault-Tolerance Requirements

13.3 - Action States:

Description: The game should be able to return to a saved state after the last successful player or bot action and before the current action should the current action fail.

Rationale: In order to prevent complete system failure after every individual failed action by either the player or the bot players, the game should be able to return to a recent and stable game state. This state system allows the player to simply re-enter a single action rather than restart the entire game.

Fit Criterion: The game must maintain a most recent success state upon each successful action and when a failure occurs load that game state.

Acceptance Test: 13.2

13d Safety-Critical Requirements

13.4 - Safety Issues:

Description: This game poses no threat to people, property or the environment.

Rationale: The game is entirely software and requires no personal information from the player and therefore poses no threat to any person. It has no physical waste production that would affect property or the environment.

Fit Criterion: N/A

Acceptance Test: N/A

14 Maintainability and Supportability Requirements

14a Maintenance Requirements

14.1 - Updates and Maintenance:

Description: This game should receive regular updates and bug fixes that take no longer than five minutes to download.

Rationale: The updates and bug fixes should be minor changes to the game that only improve already existing functionality.

Fit Criterion: With an average speed internet connection, all updates must be complete within five minutes.

Acceptance Test: 14.1

14b Supportability Requirements

14.2 - Player Support:

Description: This game should have an email address for receiving questions from users with responders that check for emails daily.

Rationale: The game is fairly straightforward but occasionally users may have questions about the game. An email address is an easy and accessible way to receive infrequent questions from players.

Fit Criterion: A help email address must be visible on the main menu and emails sent to it must receive a response within 24 hrs.

Acceptance Test: 14.2

14c Adaptability Requirements

14.3 - Platform Adaptability:

Description: This product must be able to run on Windows, Linux, Mac, iOS and Android devices.

Rationale: This product is entirely digital and designed to be distributed by an online game store or app store. In order to make the game accessible to the most users on these

platforms, the game must be compatible with Windows, Linux, Mac, iOS and Android devices.

Fit Criterion: The game must be compatible with Windows 10 and up, iOS 9 and up, Linux 5.0.0 and up and Android 9.0 and up.

Acceptance Test: 14.3

14d Scalability or Extensibility Requirements

14.4 - Independent Use:

Description: This game will be an independent software that doesn't require expansion based on number of users.

Rationale: The software will be distributed through an online store which will handle distribution and updates to any number of users. The game itself will be independent software that won't be affected by the number of users playing it all at the same time.

Fit Criterion: N/A

Acceptance Test: N/A

14e Longevity Requirements

14.5 - Maintenance and Time:

Description: This product is expected to have the budget for continued maintenance for three years.

Rationale: This product will need regular maintenance for two or three years after its initial release but the number of downloads is expected to drop off significantly after that due to the rapid growth of the gaming industry. Based on the number of downloads in the span of two years it can be determined whether continued maintenance after the third year would be profitable or not.

Fit Criterion: The budget for this game must be able to support updates and maintenance for a minimum of three years.

Acceptance Test: 14.4

15 Security Requirements

15a Access Requirements

15.1- Access Control:

Description: The product has minimal sensitive data collected and only those with security clearance level shall be able to access this data.

Rationale: Our users privacy is our top priority. Since the product does collect sensitive data we want to ensure that not just any employee can access that information.

Fit Criterion: Only employees who have passed the requirement security clearance shall have access to this sensitive data.

Acceptance test: 15.1

15b Integrity Requirements

15.2 - User access:

Description: The product shall not let the users access and manipulate any data.

Rationale: The data collected will be stored for the duration of games only. We want to ensure privacy of the users and this starts with having protection of who can access and manipulate the data.

Fit Criterion: The product shall identify any unwanted data tampering and report user.

Acceptance test: 15.2

15c Privacy Requirements

15.3 - Privacy Insurance:

Description: Users will have to read and agree to data being collected.

Rationale: Upon first time entering the systems environment the users will be notified of data being collected and will have to agree and submit before moving on to play the game.

Fit Criterion: N/A

Acceptance test: 15.2

15d Audit Requirements

15.4 - System Audit

Description: Upon request the system will gather necessary records for audit.

Rationale: Want to build a system that will pull any data necessary for audit.

Fit Criterion: N/A

Acceptance test: 15.1

15e Immunity Requirements

15.5 - Safety Checks:

Description: To protect the software there will be daily security checks.

Rationale: Due to the online gaming the system is prone to viruses, worms, etc. To protect itself the system will run through a series of tests to ensure that there are no unwanted tampering.

Fit Criterion: The system will send a message to the security team with an updated status of the test run.

Acceptance test: 15.3

16 Usability and Humanity Requirements

16a Ease of Use Requirements

16.1 – Efficiency of Use

Content: The user shall be able to understand the game fairly easy, within a couple of minutes or less.

Motivation: Due to the relatively simple features of the concept, gameplay, and controls, the user should have little trouble with understanding how to play the game during the first few minutes or less of playing the game.

Fit Criterion: Around 90% of users should be able to play the game with full understanding of the concepts of the game within the first few minutes.

Acceptance Tests: 16.1

16.2 – Ease of Remembering

Content: Since the game is fairly easy, the game would require only a few rules to be remembered.

Rationale: The gameplay would involve some critical thinking which would require remembering the knowledge of the game rules.

Fit Criterion: The users should be able to remember the rules fairly quickly while playing the game.

Acceptance Tests: 16.1

16.3 – Error Rates

Content: The player is projected to fail at one point since the game has a limited number of moves.

Rationale: Since there is not an infinite number of moves, and the game has its expiration time, the players will have to fail at one point.

Fit Criterion: 80% of the players are anticipated to fail during the first three times they play the game due to being introduced to the rules of the game.

Acceptance Tests: 16.1

16.4 – Overall Satisfaction

Content: The product shall make the users want to use it.

Rationale: Due to the pick-up-and-play motifs of the game, the user is expected to be pleased with a short run, played whenever they feel like.

Fit Criterion: The users should want to play the game at least once more after their first time of play.

Acceptance Tests: 16.2

16.5 – Feedback

Content: The product shall be used by people with a little to no training, and possibly no understanding of English.

Rationale: The game's rules are fairly simple and the user will intuitively understand the gameplay. The user might only need English for understanding the game rules.

Fit Criterion: 80% of the game users should be able to state that they understood the game rules after at least one time play of the game.

Acceptance Tests: 16.3

16b Personalization and Internationalization Requirements

16.6 - Localization

Content: The product shall retain the buyer's buying preferences.

Rationale: The buyer can choose the preferences of the game, like what language or what symbols or letters the game should use.

Fit Criterion: The product shall allow the user to select a chosen language.

Acceptance Tests: 16.3

16c Learning Requirements

16.7 - Learning Requirements

Content: The product shall be easy for six year old children to learn.

Rationale: The product shall be able to be used without any previous training, and it shall be able to be easily understood by the first time users.

Fit Criterion: The first time users should be able to successfully win the first level of the game after at least one time of play.

Acceptance Tests: 16.1

16d Understandability and Politeness Requirements

16.8 - Understandability and Politeness

Content: The product shall use symbols and words that are naturally understandable by the user community.

Rationale: The product's symbols and words shall be clear and easily understandable for users.

Fit Criterion: A six year old child should be able to easily understand how to play the game after one play.

Acceptance Tests: 16.1

16e Accessibility Requirements

16.9 - Accessibility Requirements

Content: The product shall allow one-handed usage and include colorblind modes.

Rationale: Common disabilities that affect gamers typically include motor functions inhibitions and sight-based debilities.

Fit Criterion: The game will be usable by partially sighted users.

Acceptance Tests: 16.1

16f User Documentation Requirements

N/A

Description: Since this is a game, the user documentation will not be required to be supplied as part of the product.

16g Training Requirements

N/A

Description: Since this is the game, there won't be any training requirements.

17 Look and Feel Requirements

17a Appearance Requirements

17.1 -Look and Display

Description: The system will maintain a specific color pallet

Rationale: To ensure that the system is distinguishable from other titles it will maintain a specific color pallet that will be welcoming to all players.

Fit Criterion: 75% from a sampling of players will find that they can distinguish the colors of the game.

Acceptance test: 16.1, 17.1

17.2- Look and Display:

Description: The system shall display readable fonts throughout the game.

Rationale: The system should display readable fonts where users will be able to see with ease throughout the entire game.

Fit Criterion: From a sampling of players 75 % of users will be able to read all fonts properly.

Acceptance test: 16.1, 17.2

17b Style Requirements

17.3 - Style and Feel:

Description: The overall style of the logo will be fun and playful; attracting all ages.

Rationale: The market for time played is very competitive in video games. Having our style feel more welcoming will help stand out and attract more users.

Fit Criterion: After users' first time using the software 75 percent of users will feel motivated to play through the game.

Acceptance test: 16.2

18 Operational and Environmental Requirements

18a Expected Physical Environment

18.1 - Environments:

Description: The product could be played on Windows, Linux, or Mac operating systems from the Steam store.

Rationale: This game is meant to be played with friends or strangers. Allowing the system to run on multiple platforms will ensure this.

Fit Criterion: The product shall operate on all desired platforms.

Acceptance Test: 18.1

18b Requirements for Interfacing with Adjacent Systems

18.2 Interfacing systems:

Description: Our product should interface with Windows, Linux, or Mac.

Rationale: The system shall run on multiple operating systems. Each system will have different downloadable requirements. Therefore, there will be different downloadable packages available.

Fit Criterion: The system will be a big file to download and the users operating system must have enough memory before the game can be installed.

Acceptance Test: 18.2

18c Productization Requirements

18.3 -Download

Description: The product will be a downloadable game that will be compatible with Windows, Linux, or Mac operating systems from the Steam store.

Rationale: The user will be able to download the game through the products website.

Fit Criterion: From a sampling of users the 85% will be able to download the game with ease.

Acceptance test: 18.2

18d Release Requirements'

18.4 - Releases:

Description: There will be two releases a year; one major update release and one release to fit any bugs.

Rationale: As long as budget is sufficient the system will have two updates. One with major updates to the game and the second release to fix any bugs.

Fit Criterion: N/A

Acceptance test: 18.3

19 Cultural and Political Requirements

19a Cultural Requirements

N/A

Description: Since this is a game for entertainment the cultural requirements are not needed.

19b Political Requirements

N/A

Description: Since this is a game for entertainment the cultural requirements are not needed.

20 Legal Requirements

20a Compliance Requirements

20.1 - Legal 1

Content: No copyrighted works or intellectual property owned by another party shall be used.

Rationale: The product shall comply with intellectual property law.

Fit Criterion: The product shall comply with the lawyers specifications.

Acceptance Tests: 20.1

20.2 - Legal 2

Content: All works referenced or used in any way should be cited with written documentation of consent for use.

Rationale: The product shall comply with intellectual property law.

Fit Criterion: The product shall comply with the lawyers specifications.

Acceptance Tests: 20.1

20b Standards Requirements

N/A

Description: Since this is a game for entertainment the standard requirements are not needed.

21 Requirements Acceptance Tests

21a Requirements - Test Correspondence Summary

	Requirements																
	10.1	10.2	10.3	10.4	10.5	10.6	12.1	12.2	12.3	12.4	13.1	13.2	13.3	13.4	14.1	14.2	14.3
Tests																	
10.1	x	x	x														
10.2				x													
10.3					x												
10.4						x											
12.1							x										
12.2								x									
12.3									x								
12.4										x							
13.1											x	x					
13.2													x				
14.1															x		
14.2																x	
14.3																	x

	Requirements																							
	14.4	14.5	15.1	15.2	15.3	15.4	15.5	16.1	16.2	16.3	16.4	16.5	16.6	16.7	16.8	16.9	17.1	17.2	18.1	18.2	18.3	18.4	20.1	20.2
14.4		x																						
15.1			x			x																		
15.2				x	x																			
15.3							x																	
16.1								x	x	x				x	x	x	x							
16.2											x							x						
16.3												x	x											
16.4																								
17.1																	x							
18.1																			x					
18.2																				x	x			
18.3																						x		
20.1																							x	x

21b Acceptance Test Descriptions

<u>10.1 - Basic Game Test:</u>

Description: This test loads the game, starts a new game, saves the game, displays the high score to the player, exits the game and loads the save file from the main menu. This tests basic functions of the game that are necessary for players to start and play the game.

10.2 - Change Level Test:
Description: This test checks for five separate unique levels of varying difficulty. There must be a way for players to change levels and view them.

10.3 - Mouse Test:
Description: This test checks for the player to have the ability to play using the mouse. A mouse connected to a computer must be used to play the game.

10.4 - Mobile test:
Description: This test checks for the ability for players to play using a mobile device with touch screen. The player must be able play a full game with the mobile application.

12.1 - Response time:
Description: This test times the response of the system to each player's input move. The time for the system to respond must not exceed 3.5 seconds.

12.2 - Frames
Description : This test uses a frame counter to ensure that animation stays between 57 and 61 frames per second.

12.3 - Updated Moves
Description: This test times how long it takes for the system to update the board after a move is made. The average time should be 3 seconds with 90% of moves not exceeding update time greater than 4.5 seconds.

12.4 - Capacity
Description: This test runs 1000 simultaneous games in order to test the game's capacity for players.

13.1 - Failure Check
Description: This test runs the game for 5 hours a day for one month and counts the number of failures the system experiences. After each failure it should take less than 3 seconds to reopen the game and the system should not fail more than 28 times a month.

13.2 - Save States

Description: This test should keep track of save states between turns and if the system should experience a failure the game should load the save state.

14.1 - Update Time
Description: Test for the length of time it takes to download updates to the system after each update is developed and before it is released. The time should not exceed 5 minutes.

14.2 - Email Questions
Description: Test email response time. It should take less than 24 hours to get a response from the question email.

14.3 - Compatibility
Description: This test checks if the game can be downloaded to Windows 10, Linux 5.0.0, iOS 9 and Android 9.

14.4 - Budget for Updates
Description: This test ensures that there is enough money in the budget for continued updates for three years.

15.1 - Data Protection
Description: This test ensures that players do not have access to other player's personal information but is accessible to the developers.

15.2 - Data Protection 2
Description: This test makes sure that the data from each game is erased after game ends.

15.3 - Security Checks
Description: This test makes sure that the game is virus free every day. Security issues will be reported by the system to the security team and manager.

16.1 - Beta Testing
Description: A test with live players of all ages will be run to check for ease of use, ease of player memory, number of players' fails and wins over the course of a week.

16.2 - Survey
Description: A survey should be run after the bet testing period asking testers if they like the game, if they would continue to play it and how easy the game was to navigate.

16.3 - Understanding

Description: Between games during the beta testing period check for player's understanding of the game as well as the language used.

16.4 - Options for Accessibility
Description: This checks the use of one handed and colorblind mode available from the main menu.

17.1 - Change Background
Description: Checks for the ability to change the background on the game menu.

18.1 - Cross Compatibility
Description: This tests whether the game can be played by players using different devices.

18.2 - Downloads
Description: This tests if the download of the online store works and can play the game.

18.3 - Final Failure Check
Description: This tests that each release works 95% of the time before being released.

20.1 - Compliance With the Law
Description: This tests that the game is in compliance with intellectual property law.

III Design

22 Design Goals

- Compatibility: Design should be compatible with Android, iOS and PC platforms with cross compatibility for all platforms.
- Extensibility: Face 5 should be able to be patched and updated regularly as well as when bugs or other errors occur.
- Reliability: This product should be available for the user to play 24 hours a day with minimal issues.
- Usability: The user interface should be easy to read and understand for the majority of players.
- Performance: Face 5 should respond quickly to user input and interaction with up to 4 players.

23 Current System Design

Face 5 is an original game, so there is no pre-existing system.

24 Proposed System Design

24a Initial System Analysis and Class Identification

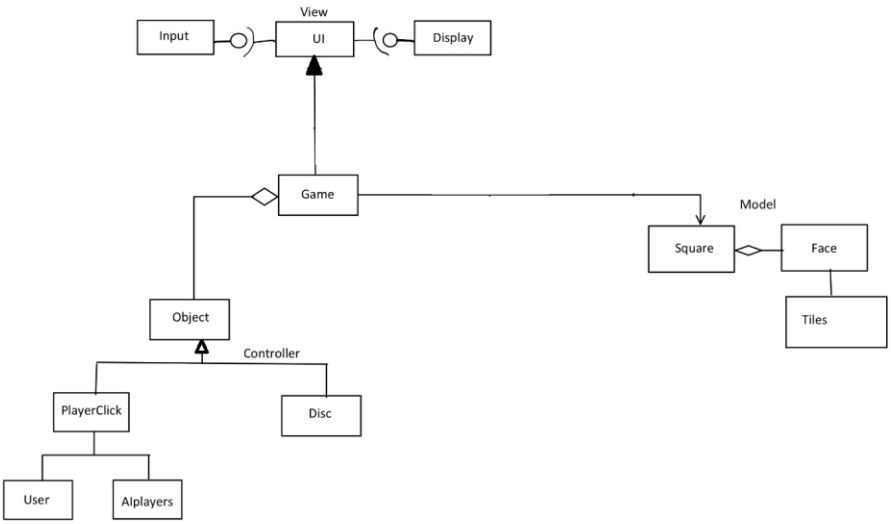


Figure 5 - Class Design

The key portions of this will be the discs and the squares. This can be seen in the figure above, through the breakdown of classes.

PlayerClicks can be done by the user of an AI player. Discs class and PlayerClicks class both inherit from the object class. Object includes all game entities that could interact with PlayerClicks and Disc.

The Square class will be a collection of Faces which are made up of tiles. The tiles will be the individual slot entries. The board will be made up of faces and tiles. It will get updated as players place pieces on the board.

The UI class will wait for input and display class and update from there. It will get user input and the display be responsible for the graphics and music/sound.

The Game Class will be the logic of the game and manage game state. It will have a reference of the cube and the UI classes. It will also have the collection of objects throughout the game. If we think of this as a road map the game class is the main interaction. All information will pass through here and the Game class will distribute it accordingly.

24b Dynamic Modelling of Use-Cases

Use Case 1: Player Clicks

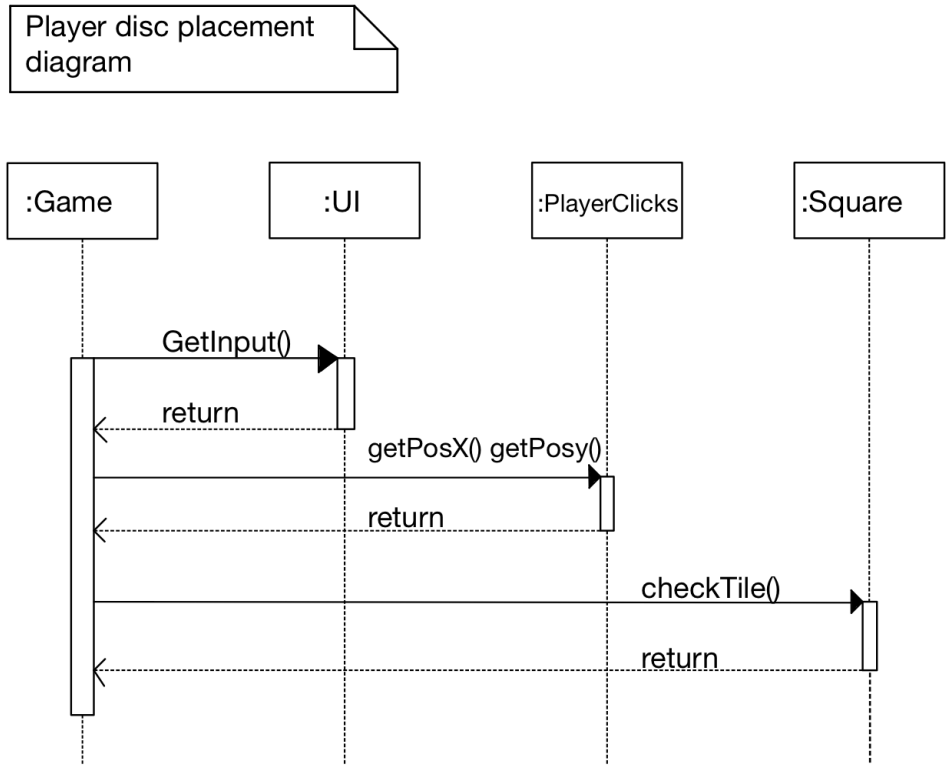


Figure 6- Sequence Diagrams

The player clicks where they would like to place a disc, the information is then sent to the Game class which then sends the information to the appropriate classes.

24c Proposed System Architecture

The development team decided that the model view controller would be the best fit for the system. This model will ensure that the user gets a quick response; which is crucial for this multiplayer game. The game class will be the model, display class will be view and lastly the input class will be the controller.

24d Initial Subsystem Decomposition

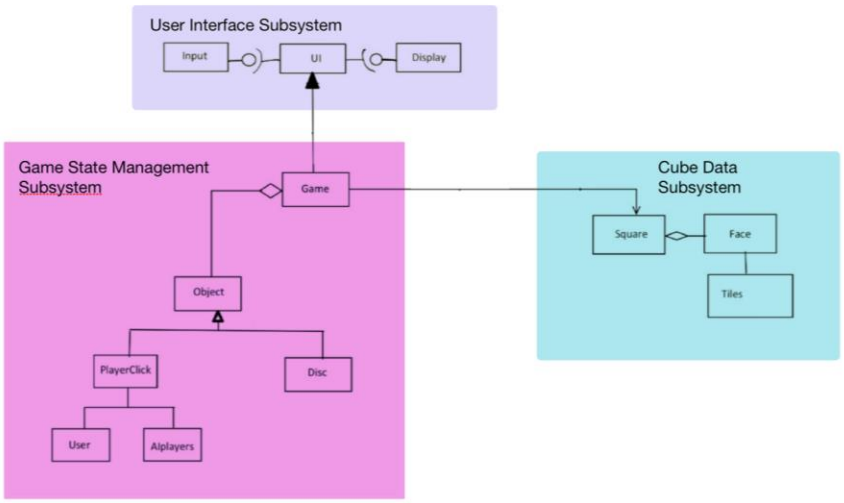


Figure 7 - Subsystems

User Interface subsystem:

This subsystems functionality is to read in the data from the user and display the appropriate graphic in response to the user's input.

Game State Management Subsystem:

The overall logic of the game will be controlled by this subsystem. It will keep track of all the users data. The control flow is established here.

Cube Data Subsystem:

This subsystem will be responsible for the storage of the layout of the cube. Upon request from the game state management this subsystem will store the proper data required to update the game state.

25 Additional Design Considerations

25a Persistent Data Management

This product will collect scores and possibly multiple files over the course of the game. The scores will be placed in and read from a text file, updated after the end of each game. The save files will be created each time a player saves the game with a different player's name. If the player's name is already in saved games, the save file will be overwritten.

25b Boundary Conditions

The boundary conditions would be in the options configuration file, such that it saves the settings and updates them accordingly when changed in game. This is such that the user does not have to change their settings back each time they boot up the game.

25c User Interface

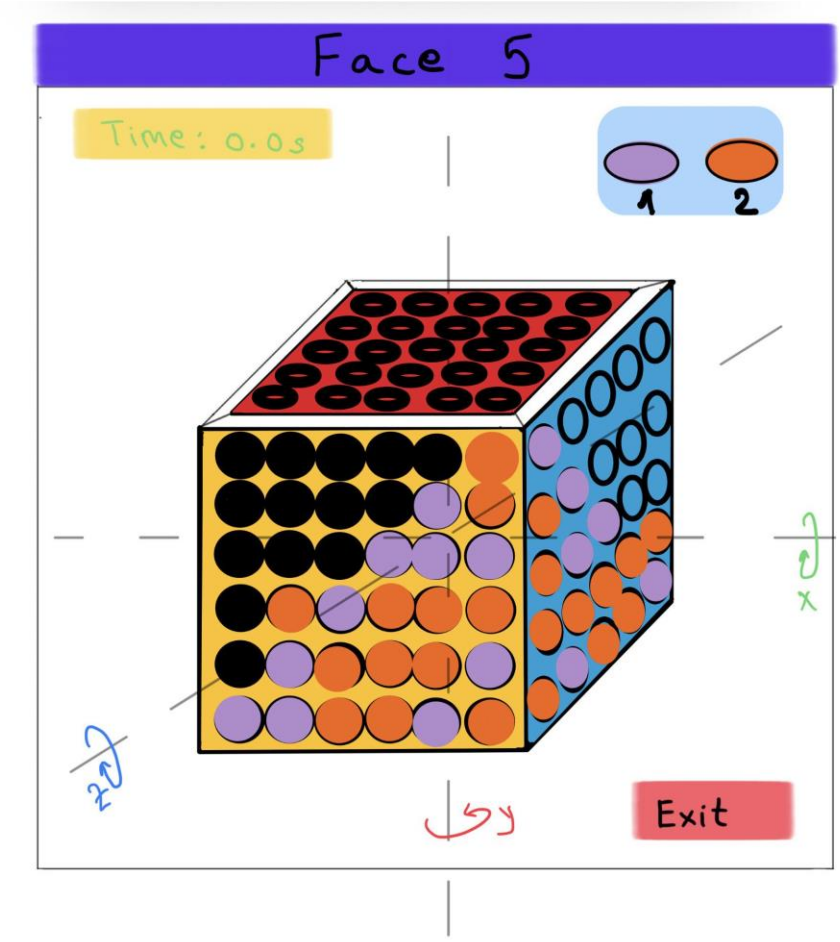


Figure 8 - User Interface

25d Application of Design Patterns

The most important design patterns for this product are Model View Controller, Mediator, and Observer.

- **MVC:** The game state is the model, the graphic display is the view, and the user input is the controller.
- **Mediator:** The Game class manages communication between all other classes.
- **Observer:** The Square must be notified when the user puts a new disc or removes it, and also when the user manages to connect 5 discs on the first face of the square.

26 Final System Design

26a Package Breakdown

The system is packaged into three main sections following the Model-View-Controller (MVC) format to allow the system to be ported to multiple platforms, such as mobile or PC.

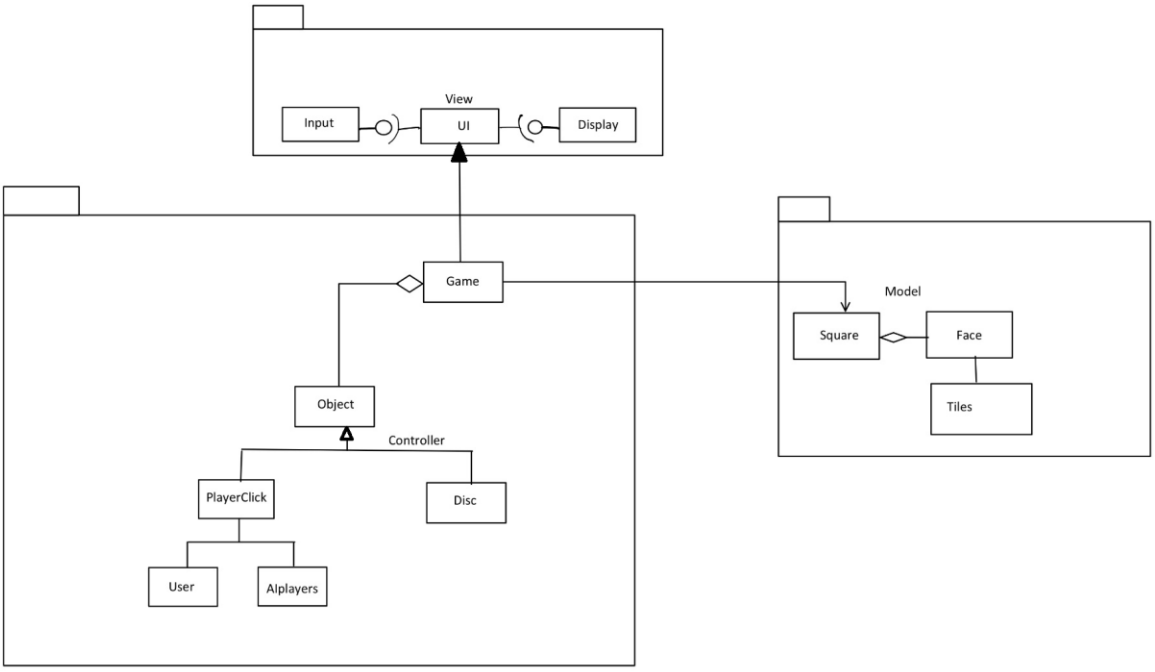


Figure 9 - Model-View-Controller

The diagram shows that the Model is class Square which consists of faces which further consist of tiles. The square and the faces on the square together determine where objects are placed and the relationship between. The View is represented by the User Interface, Input, and Display, which interface with the user and respond to its input by placing the discs at wanted locations or

by removing them from the same. Controller is represented by the Game class, which contains references to all the objects and communicates both between the Model and the View. This layout serves as the package diagram, with each package serving its primary purpose in the MVC layout.

26b System Overview

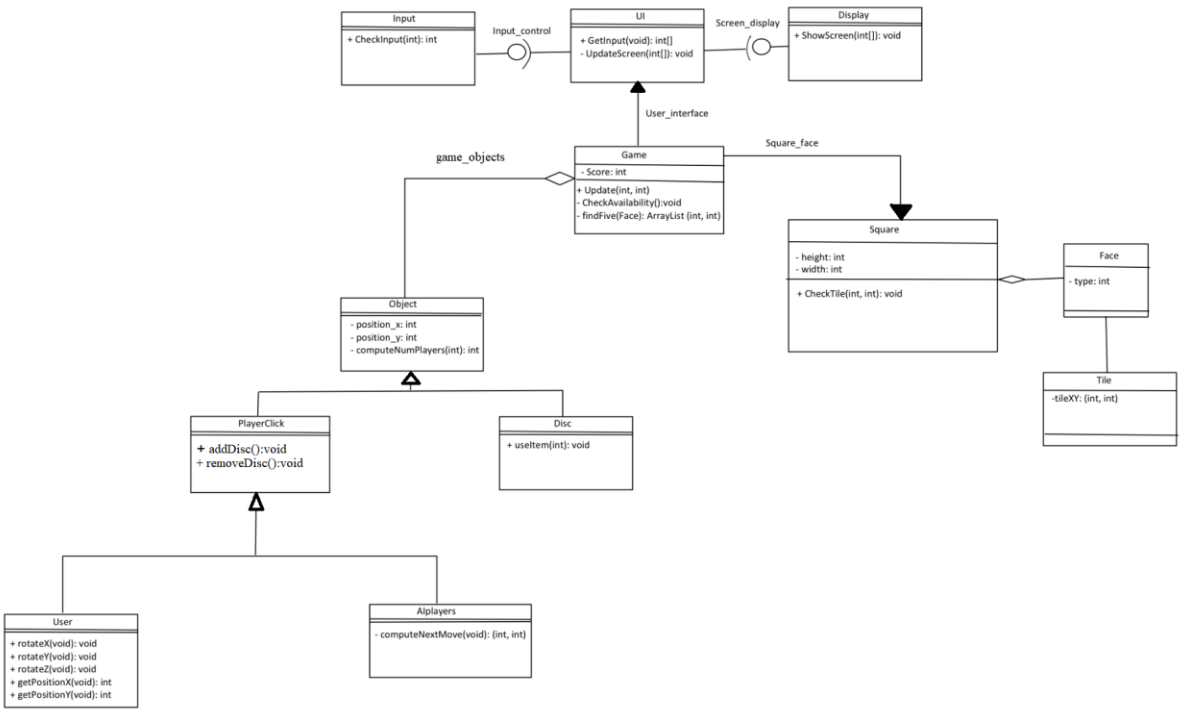


Figure 10 - Class System Design

The class system is combined of a few main subsystems that follow the packages shown in 26a. This diagram is a detailed description of the classes, which will be discussed in depth in each subsystem.

27 Object Design

27a Packages

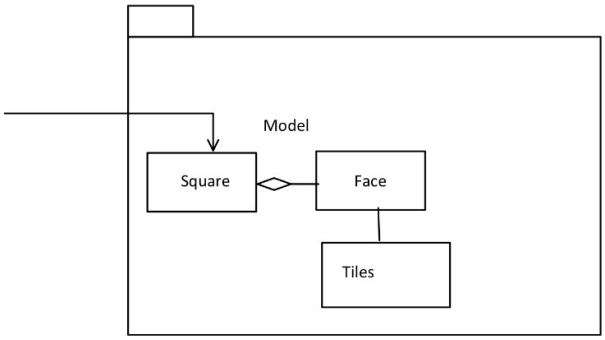


Figure 11 - Model

The system will be divided into three main packages following the Model-View-Controller format. In the Model package we have the Square object that is made of Faces and Faces are made of Tiles that all make up elements and information to be presented in the View.

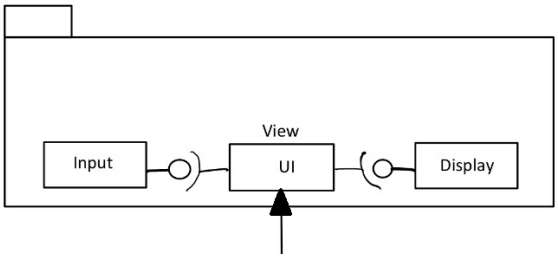


Figure 12 - View

The View package contains the user interface, takes user input, and shows the display to the player.

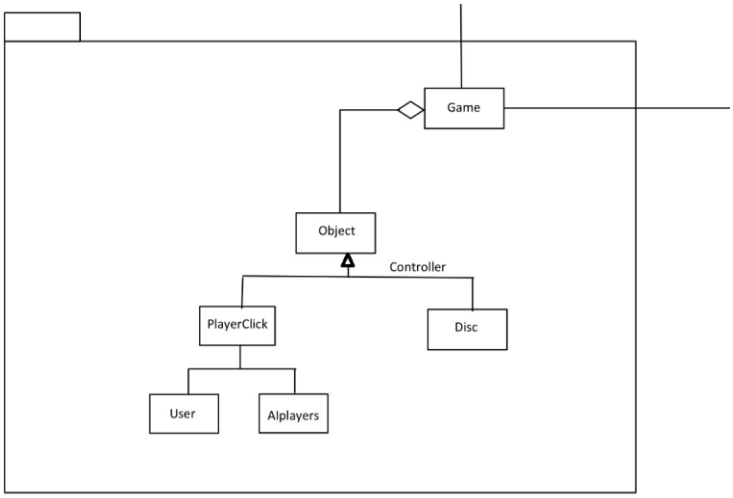


Figure 13 - Controller

The Controller is represented by the Game class and all of its components. The Game class consists of all game Objects including Discs and PlayerClick objects that interact with each other through the Game class. The Controller then communicates these interactions with both the View and Model to create visuals for the player.

27b Subsystem I - User Interface

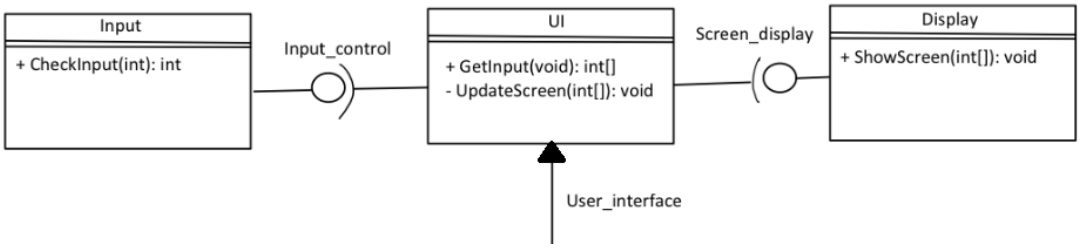


Figure 14 - User Interface Subsystem

The User Interface Subsystem creates and displays the interface for the user to interact with. Its main purpose is to take in input and respond to the changes that happen in Game and Square as the game is in progress.

The display must be able to update and respond quickly to changes in the game state from the input and user interface. Input type will change based on the platform being used by the player such as touch screen presses from mobile devices and mouse clicks from PC players.

27c Subsystem II - Cube Data

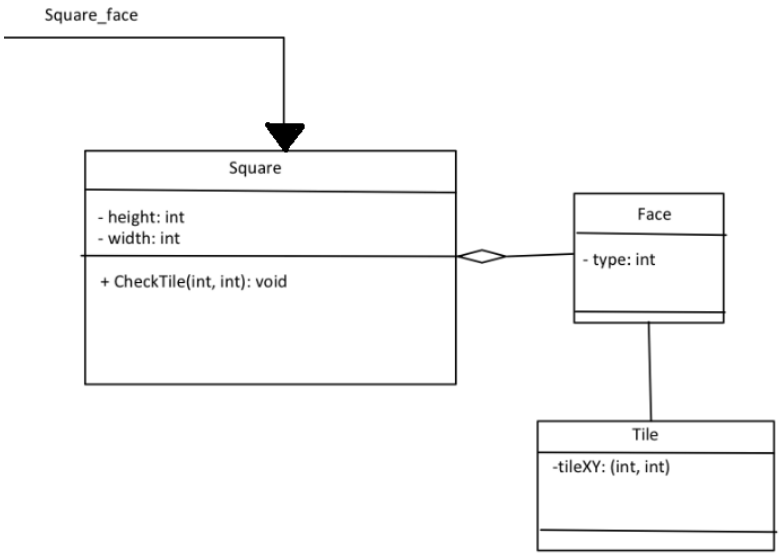


Figure 15 - Cube Data Subsystem

The Cube Data Subsystem tracks the locations for each piece on the cube. The Square class is made up of faces and each face contains tiles. The Tile class holds the coordinates of each possible piece placement and the Square class has the ability to check if a coordinate is free or already contains a player's piece.

27d Subsystem III - Game State Management

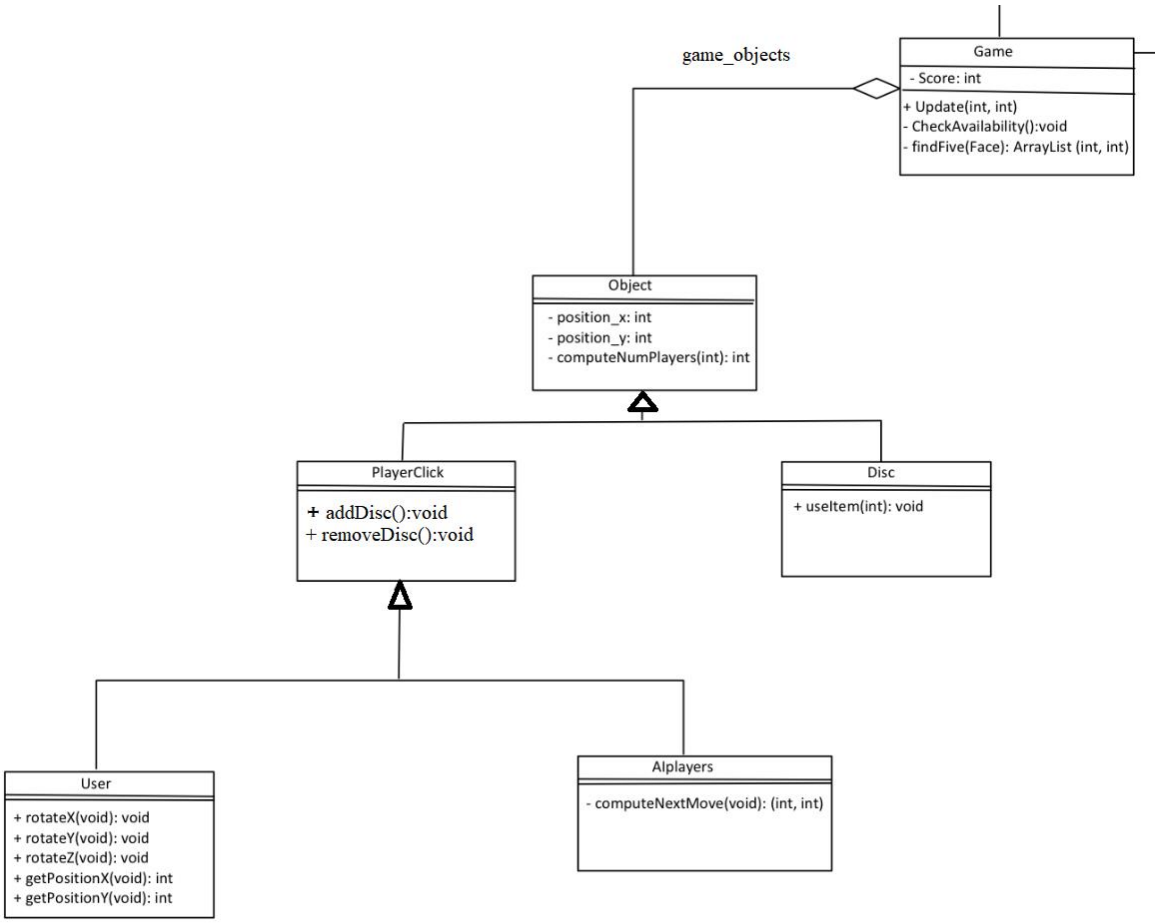


Figure 16 - Game State Management

The Game State Management Subsystem contains the functions for all relevant game events and interactions between the game objects such as making connections or even winning the game. Using the locations and objects from the Cube Data Subsystem, a player can add or remove discs from the board. The Game class checks with the Cube Data to determine if the input from the user is valid before changing states. Every different type of object has fields for its own position on the board and the AI players must communicate within the Game class in order to compute their next move when playing with AI.

IV Project Issues

28 Open Issues

Users with color blindness or other visual impairment may have difficulty seeing the display. Options for users with disabilities may need to be developed

Console platforms were not considered for initial development but console development could be considered if there is a demand.

29 Off-the-Shelf Solutions

29a Ready-Made Products

Game development software could be used to create Face 5 such as Unity. The use of game development software could drastically change the development process as well as the final product's visual design and structure but would also reduce the work needed to be done by developers.

29b Reusable Components

Not applicable.

29c Products That Can Be Copied

Concepts and design points could be taken from basic online Connect 4 games and built upon to create Face 5.

30 New Problems

30a Effects on the Installed Systems

The game may interfere with screen filters installed on the user's device.

30b Potential User Problems

Content: Many users of the product would not have any trouble with the product. However, users with disabilities such as motor functions or vision problems are most likely to have issues.

Motivation: Disabled users would have an ill-experience with the game, and as such, taking steps to address those problems is only appropriate.

30c Limitations in the Anticipated Implementation Environment That May Inhibit the New Product

The product will run on mobile and PC platforms. It will be thoroughly tested before release to ensure there are no limitations. Physical limitations would be a property of the device it is running on, not this product.

31 Migration to the New Product

Not Applicable.

32 Risks

There is a high probability that a market analysis will show a trend towards people being uninterested in this style of online gaming. This trend may be a major issue when it comes to determining the necessary budget for the maintenance of the software for several years. Trying to keep up with the current trends by adding features or more elements of gameplay could also lead to an overworked team and increased schedule pressure. Maintaining the simplicity of the design is important to keep the probability of schedule pressure low.

With varying graphical capabilities on multiple different platforms and cross compatibility there may be issues with low quality graphics but the simplicity of the design makes this unlikely.

33 Costs

Here is a team we expect to work on the project:

- 2 Software Engineers
- 2 Game Designers
- 1 Project lead
- 2 QA / Software testers
- 1 Full stack web developer for website
- 1 Graphics artists
- 1 Sound and music designer
- 1 Lawyer
- 1 Content writer for marketing

For a total of 12 people on this team. However throughout the creation of this project not all team members will be present for a short period of time. Such as the lawyer, graphic designer and sound/music designer they will only pop in and out at various times of the development process. For the development of this project there are 20 requirements to fulfill and expect the product to be complete within 1 year; though it is possible for the game to be finished earlier. With all that being said the game is expected to cost between $500,000 to $700,000.

34 Waiting Room

Content: So far we have a game that is for 4-6 users. Our team was considering adding a more complex 3D shape such as a dodecahedron. This shape will allow up to 12 players.

Motivation: Past releases 2 this would be a major feature update allowing for more users to play together.

Considerations: There are some considerations for this major update. The system will allow the users to play on the cube or dodecahedron. For the dodecahedron the developers will have to allow for more connections in the game. The system will allow for a group of users to join the game together or a matchmaking algorithm for users who would like to jump in a random game.

35 Ideas for Solutions

As outlined in the previous section of new problems about the users with disabilities, alternative or simple controls and colorblind options would be able to help address the issues they would have.

36 Project Retrospective

During the creation of this product our team met once a week. Though during our meeting we would catch each other up with each other's progress and discuss possible changes, we found that on some occasions we would have to meet up more than once a week. In the future more frequent but shorter meetings with an agenda may help improve our team's dynamic.

V Glossary

Model View Controller:

A design pattern used to separate internal representations of information from the ways information is presented to and accepted from the user.

Path:

The sequence of piece placements that a player has made in the cube.

Platform:

The hardware and operating system that the program is running on.

User Interface:

The means for the user to view output and interact with the system. This will change from one platform to another.

VI References / Bibliography

[1] Robertson and Robertson, Mastering the Requirements Process.

[2] A. Silberschatz, P. B. Galvin and G. Gagne, Operating System Concepts, Ninth ed., Wiley, 2013.

[3] J. Bell, "Underwater Archaeological Survey Report Template: A Sample Document for Generating Consistent Professional Reports," Underwater Archaeological Society of Chicago, Chicago, 2012.

[4] M. Fowler, UML Distilled, Third Edition, Boston: Pearson Education, 2004.

[5]A. Lau, C. Riviera, M. Searcy “Greedy Dwarf Project Report”

VII Index

Design 35,
Project Description
Project Issues
Model View Controller
Path
Platform
User Interface
Requirements
Unity
User Interface