Class Project: Connect 4

Authors: Dane Bramble, Josh Gorodinsky, Patrik Kozak, Ben Townsend

CIS 350: Fall Semester 2018

Table of Contents

Project Information 4

Features 4

Screenshots 4

Project Plan 5

Requirements & Definition 5

Development 5

Verification 5

Maintenance 5

Umbrella Activities 5

Responsibilities 5

Requirements & Specification 6

Use-Cases 6

Natural Language Requirements 6

Natural Language Requirements 6

Design 7

Development 8

Code Standards 8

Static Analysis 8

Code Documentation 8

Configuration Management 8

Verification 9

Integration Tests 9

Unit Tests 9

Code Coverage 9

Requirements Coverage 9

Postmortem 10

Earned Value 10

Variances 10

Lessons Learned 10

References 11

# Project Information

[Project Description]

This Project is a java application that allows up to 4 people to play a local game of Connect 4 together. The user will be able to place their chip on the game board in order to get four of their chips in a row in any direction. The first person to do this wins!

## Features

[Project Features Included]

* The ability to play Connect 4 on a game board
* An interact-able GUI with music
* Local Multiplayer for up to 4 players
* Up to 3 Computer Opponents (for a maximum of 4 players)
* The ability to select the name and color of chip for each player

## Screenshots

[Screenshots of Project]

# Project Plan

[Overview of Project Plan, including any customizations to the software process model]

## Requirements & Definition

[Plan for requirements and definition detailed here]

## Development

[Plan for development / implementation detailed here]

## Verification

[Plan for verification detailed here]

## Maintenance

[Plan for maintenance (e.g., problems found by prof or during updates for next release) detailed here]

In between releases, we plan to update any bugs are found during this period. In addition, we will continue refining the logic and the GUI until the final release. If time permits, we will include an additional game mode to expand the program.

## Umbrella Activities

[Plan for project management and status tracking / meetings detailed here]

## Responsibilities

[Describe the responsibilities of each member, how you plan to break down the work]

The responsibilities are as follows:

* Dane Bramble
  + Player/Computer Player logic, System Settings
* Ben Townsend
  + Game Rules / Board Logic
* Patrik Kozak
  + GUI
* Josh Gorodinsky
  + GUI

# Requirements & Specification

[Description of methods used (e.g., use- case diagrams, user stories, use case-descriptions)]

## Use-Cases

[Include use-case diagrams AND use-case descriptions here]

**Use Case Diagrams:**

Connect 4 Game

Player

Select Mode

Type Name

<extends>

Choose

Player

Options

<extends>

Select Chip Color

Choose Player Count

Exit Program

Return to Menu

Place consecutive chips

threatens

mitigates

Enforce

Game

Rules

Choose Chip Location

<includes>

**Use Case Descriptions:**

* Use-Case: Select Mode
  + Actors: Player
  + Type: Primary and essential
  + Description: The player selects which version of Connect 4 they would like to play.
  + Use-Cases: The Player must first complete the ‘Choose Player Count’ use-case.
* Use-Case: Choose Player Options
  + Actors: Player
  + Type: Primary and essential
  + Description: Each player decides what their name and color is before the game starts.
  + Use-Cases: The player must first complete the ‘Select Mode’ and ‘Choose Player Count use-cases.
* Use-Case: Choose Player Count
  + Actors: Player
  + Type: Primary and essential
  + Description: The player picks how many players are in the current game at the selection screen.
* Use-Case: Exit Program
  + Actors: Player
  + Type: Primary and essential
  + Description: The player is able to exit the program at any time to return to the desktop.
* Use-Case: Return to Menu
  + Actors: Player
  + Type: Primary
  + Description: The player can return to the selection screen menu at any point after leaving it.
  + Use-Cases: The player must first complete the ‘Select Mode’ and ‘Choose Player Count use-cases.
* Use-Case: Choose Chip Location
  + Actors: Player
  + Type: Primary and essential
  + Description: The player chooses where to drop the chip on the game grid.
  + Use-Cases: The player must first complete the ‘Choose Player Options’ use-case.
* Use-Case: Type Name
  + Actors: Player
  + Type: Secondary
  + Description: Each player types in the name that they want to display when it is that player’s turn.
  + Use-Cases: The player must first complete the ‘Select Mode’ and ‘Choose Player Count use-cases.
* Use-Case: Select Color
  + Actors: Player
  + Type: Secondary
  + Description: Each player chooses which color their chip will be.
  + Use-Cases: The player must first complete the ‘Select Mode’ and ‘Choose Player Count use-cases.
* Use-Case: Enforce Game Rules
  + Actors: Player
  + Type: Secondary
  + Description: The player can only make an action that aligns within the rules of the game.
  + Use-Cases: The Player must first complete the ‘Choose Chip Location’ use-case.
* Misuse-Case: Place Consecutive Chips
  + Actors: Cheater
  + Type: Primary
  + Description: A cheater could negate the rules of the game to place more than one chip at one time.

## Natural Language Requirements

[Include requirements here]

* Use-Case: Select Mode
  + **UR07:** The Connect 4 Game shall provide the player with the ability to choose which version of connect four to play at the beginning of the program.
* Use-Case: Choose Player Options
  + **UR03:** The player should be able to customize their name and color before the game starts.
  + **SR03:** The Connect 4 Game should limit the custom name to a reasonable length during the name selection.
* Use-Case: Choose Player Count
  + **UR06:** The player shall be able to choose the amount of players in a given game at the beginning of the program.
* Use-Case: Exit Program
  + **UR05:** The player shall be able to ability to exit the program at any time.
* Use-Case: Return to Menu
  + **UR04:** The player shall be able to return to the menu once the menu has been left.
* Use-Case: Choose Chip Location
  + **UR01**: The player shall be able to decide where their chip will be placed on the game board.
  + **SR07:** The game board shall limit the location choice to column-wise only.
* Use-Case: Type Name
  + **SR04:** The Connect 4 Game should provide each player with the ability to enter in a custom name before the game starts.
  + **SR05:** The Connect 4 Game should limit the amount of characters allowed for a custom name.
* Use-Case: Select Color
  + **SR08:** The Connect 4 Game should provide each player with the ability to select the color of their game chip before the game starts.
  + **SR09:** The Connect 4 Game should prevent each player from choosing the same color.
* Use-Case: Enforce Game Rules
  + **SR01**: The Connect 4 Game shall enforce the rules of Connect 4 at all times during a Connect 4 game.
  + **SR02**: The Connect 4 Game shall be able to determine when a game is won during a match
  + **UR02**: The player shall be able to use the rules of the game to win a match
* Misuse-Case: Place Consecutive Chips
  + **SR06**: The Connect 4 Game shall prevent a potential cheater from placing consecutive chips on the game board at any point during a Connect 4 Game.

## Natural Language Requirements

[Include traceability (matrix) from requirements to use cases]

**System Requirements Traceability Matrix:**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | SR01 | SR02 | SR03 | SR04 | SR05 | SR06 | SR07 | SR08 | SR09 |
| UR01 |  |  |  |  |  | X | X |  |  |
| UR02 | X | X |  |  |  | X |  |  |  |
| UR03 |  |  | X | X | X |  |  | X | X |

**Use-Cases Traceability Matrix:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | UR01 | UR02 | UR03 | UR04 | UR05 | UR06 | UR07 |
| Select Mode |  |  |  |  |  |  | X |
| Choose Player Options |  |  | X |  |  |  |  |
| Choose Player Count |  |  |  |  |  | X |  |
| Exit Program |  |  |  |  | X |  |  |
| Return to Menu |  |  |  | X |  |  |  |
| Choose Chip Location | X |  |  |  |  |  |  |
| Type Name |  |  | X |  |  |  |  |
| Select Color |  |  | X |  |  |  |  |
| Enforce Game Rules |  | X |  |  |  |  |  |

# Design

[Description of methods used (e.g., class diagrams)]

[Design specifications]

# Development

[Description of methods used (e.g., Checkstyle, FindBugs, Javadoc) AND any additional libraries that you used]

For development, we are using both Checkstyle and SpotBugs to help find any issues with style or logic that we may have. In addition,

## Code Standards

[Include report on code standards and justification for any variances]

## Static Analysis

[Include report on static analysis and justification for any variances]

## Code Documentation

[Include link to javadocs (likely included as separate file) and justification for any non-documented areas]

## Configuration Management

[Include link to Git repository and Git log]

**Link to Github:** <https://github.com/dkbramble/CIS350_Project>

**Git log:**

[Explain / describe method for tracking releases]

# Verification

[Description of methods used (e.g., integration & systems and/or unit testing)]

## Integration Tests

[Include manual and integration test procedures]

## Unit Tests

[Include references to unit tests in code]

## Code Coverage

[Include code coverage reports, must include: coverage of automated tests, coverage of manual tests, and combined coverage]

## Requirements Coverage

[Include traceability (matrix) from requirements to test procedures]

# Postmortem

[Include a reflection on how well the project has gone thus far]

## Earned Value

[Include the earned value calculations for your current status and any explanation of over/under runs]

## Variances

[Include any additional variance (time, coverage, functionality, …) explanations necessary]

## Lessons Learned

[Include lessons learned]

# References

[Include references here]