

Milestone 5 – UNO Card Game

Problem / Question

Create a graphic representation of a game of UNO that can play the entire game – all 108 cards and determine winner – Use JavaFX

Hypothesis

- Create java classes to handle data model, logic, and visual representations of the UNO card game. Utilize advanced techniques of Java and JavaFX to produce the end result.
 - Design processes incrementally – confirming results in stages

Project Overview

1. Develop game logic and flesh it out in a story board
 2. Design Abstract Data Types (ADT) to model data and logic
 3. Create UML documents and define dependencies / relationships
 4. Create Flowcharts (2), initial logic; updated version with classes
 5. Write Java code – confirm that it follows flowchart logic
 6. Create videos to demonstrate effective progress in each step
 7. Modify code to final stage to produce visually appealing game
 8. Modify story board to more accurately represent final product
 9. Produce a poster that highlights the process

Classes / Research

Initial Classes

Card
Deck
Hand
GameConstants

Additional Classes

DiscardPile

DrawPile

UNO

Visual Classes

AbstractView
CardView
DiscardPileView
DrawPileView
HandView
UNOImageBox

Flow Chart / UML

The diagram illustrates the UNO game logic through a flowchart and UML class definitions.

Flowchart Summary:

- UNO Sequence:** Starts at "Start UNO.main", initializes game objects, and displays the splash screen. Then it enters a loop: starts the main game loop, initializes the play field, starts a round, plays a round, ends the round, calculates total points, adds total points to active players' hands, displays end-of-round information, and finally ends.
- PlayField Sequence:** Handles the main game loop, starts a round, plays a round, ends a round, and draws cards.
- PlayRound Sequence:** Manages the active player count, looks at the top of the discard pile, and handles various card draw and play decisions based on flags (Wild Draw 4, Draw 2, Reverse, Skip) and matching cards.
- Card Class:** Represents a card with attributes like actionCard, color, drawAmount, drawCard, originalColor, originalValue, reverseCard, skipCard, value, and wildCard. It includes methods for comparing cards, getting card names, colors, and strings, and for setting colors and values.
- Deck Class:** Manages a deck of cards, including methods for initializing, drawing cards, clearing, and checking if empty.
- Hand Class:** Manages a player's hand, including methods for initializing, adding cards, drawing cards, finding cards to play, getting scores, and sorting.
- TestGame Class:** A test harness for the game, containing methods for running the game, dealing cards, discarding, and populating the play field.
- GameConstants Class:** A collection of constants used throughout the game, such as card colors, values, and names.

Annotations:

Name: Kelly Lamb
Class: CST-105 Computer Programming I
Instructor: Professor Bass
Assignment: Milestone #4 (UNO Card Game)
I certify that this is my own work.

Week 5- UNO UML Diagram
CST-105
Kelly Lamb
I certify that this is my own work.

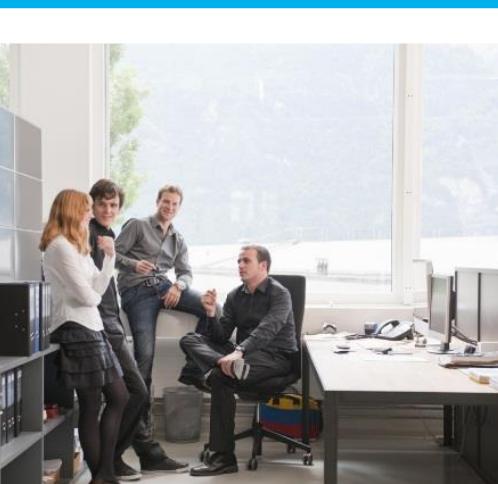
Procedure - Milestones

Step 1



Design initial logic flowchart and story board

Step 2



Design Data Model With ADT and UML

Step 3



Create
Visual
classes and
merge with
data model

Results - Screenshots

The image displays two screenshots of a UNO card game application. The top screenshot shows the initial setup with four players, each starting with 7 cards. The bottom screenshot shows the game in progress, with Player 3 having won, indicated by the large text "Player 3 Wins Score: 525".

Top Screenshot (Initial Setup):

- UNO Logo:** Large, stylized "UNO" logo.
- CST - 105:** Text in the bottom left corner.
- Kelly Lamb:** Text in the center.
- Player 1 - Player 4:** Each player has a box labeled "Score: 0" and "Count: 7". Below each box is a red UNO card icon.
- Draw Pile:** A stack of cards labeled "Count: 79".
- Discard Pile:** An empty pile labeled "Count: 1".

Bottom Screenshot (Game in Progress):

- Player 1 - Player 4:** Each player has a box labeled "Score: 0". Below each box is a red UNO card icon.
- Draw Pile:** A stack of cards labeled "Count: 74".
- Discard Pile:** A stack of cards labeled "Count: 22".
- Large UNO Logos:** Two large, stylized "UNO" logos are positioned on the right side of the screen.
- Text:** "Player 3 Wins Score: 525" is displayed prominently in the center.

Conclusion

- Game Initializes, plays, and determines a winner (Success)
 - Screenshots: 1. Title; 2. Play; 3. UNO Called; 4. Winner Screen
 - Number of classes: 14 (including anonymous classes)
 - Number of lines of code: 1399

Works Cited

- UNO – Surprisingly fun and easy game to play but provides some tough logic requirements and decisions
 - Flowchart, ADT, UML, and Story board are great tools to help in the design process
 - JavaFX offers a robust graphics programming platform with animation but requires significant research and understanding
 - Game results support hypothesis; game completes with a winner!

Base Code Snippets

- docs.oracle.com/javafx/2/get_started/jfxpub-get_started.htm
- docs.oracle.com/javase/tutorial/uiswing/misc/splashscreen.html
- examples.javacodegeeks.com/desktop-java/javafx/javafx-transition-example/
- [Introduction-to-Java-Programming-10th-ed.-Comprehensive-Version-Liang-2014-01-06.pdf](#)
- tutorials.jenkov.com/javafx/index.html
- www.genuinecoder.com/javafx-splash-screen-loading-screen/
- www.javacodegeeks.com/javafx-tutorials
- www.javatpoint.com/javafx-tutorial
- www.tutorialspoint.com/javafx/index.htm