SET

Project Documentation

Built by
Thanachote Katanyutapant
6331310121

2110215 Programming Methodology
Semester 2 Year 2020
Chulalongkorn University

The Game:

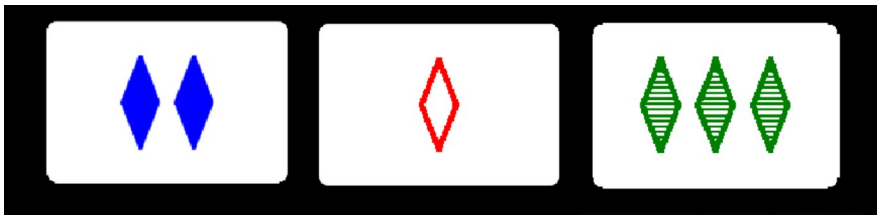
History: SET is a 1988 card game created by Marsha J. Falco, illustrated by John Langdon and Franz Vohwinkel and was published by Set Enterprises, Inc. This project is a porting project, making SET a java computer game, while keeping all the rules exactly as the original game but in a solo player mode.

Original game rules and objective:

According to the original game rules,

"The objective of the game is to identify 'SET's of three cards. A "SET" consists of three cards on which each feature is either the same on all of the cards, or different on all of the cards."

"Each card is unique in its four features: number (1,2, or 3) symbol (diamond, squiggle, or oval) shading (solid, striped, or open) color (red, green, or blue)"

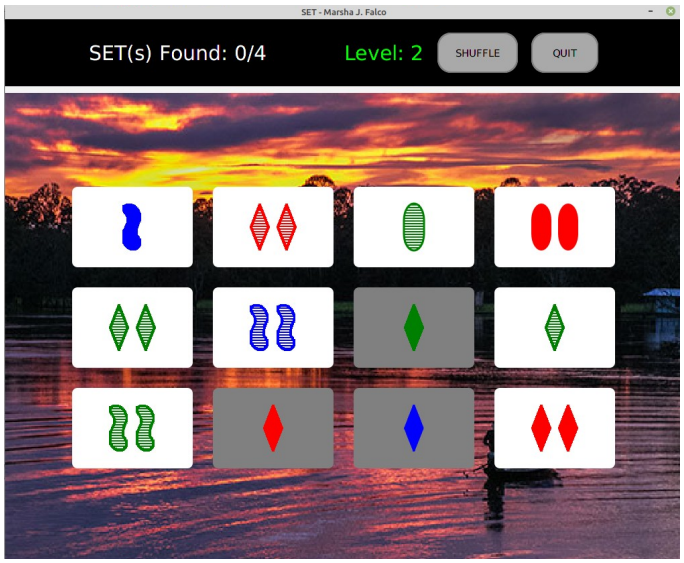


an example of a correct SET

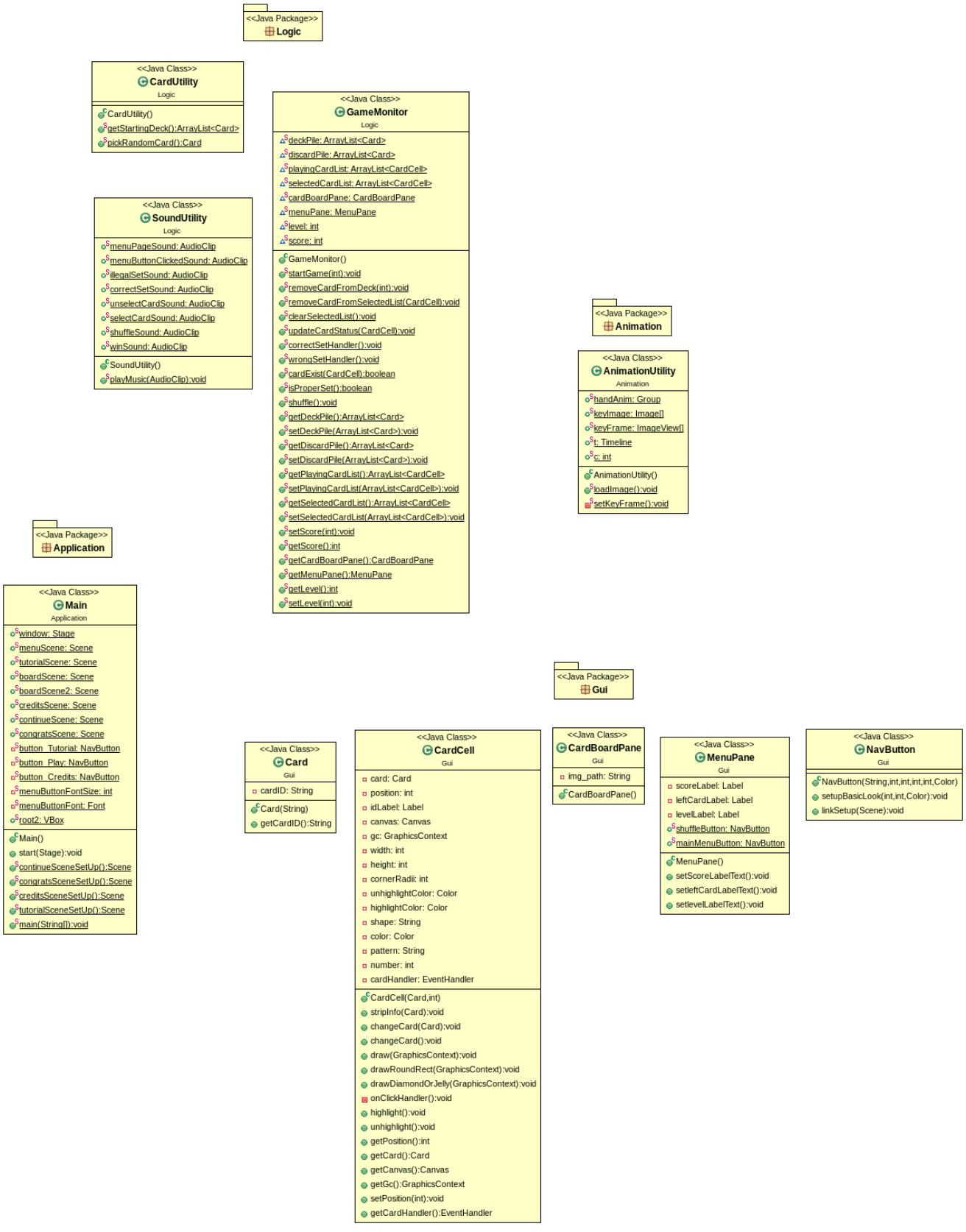
How to play this game program:

The game has 4 levels which increase in difficulty. To clear each level, the player must find 4 SETs from a pile of 12 playing cards which are randomly drawn for the player. Once they find a SET of three cards, the player should click on each cards once, and if it's correct, the program will inform the player and automatically draw out 3 new cards.

However, if the player couldn't find a SET from the 12 cards, the player could SHUFFLE to random new 12 cards. After the player has cleared the 4th level, the game ends and restarts itself to level 1.



The Code:



Objectaid Class Diagram

1. Package Animation

1.1. Class Animation Utility

1.1.1. Fields

+ *Group handAnim*	A Group for displaying keyFrame ImageView
+ Image[] keyImage	An array of Image to be shown each frames
+ ImageView[] keyFrame	An array of ImageView to be shown in the handAnim group.
+ Timeline timeline	A Timeline for playing the animation
+ int count	A counter for setting up keyFrames

1.1.2. Methods

+ static void loadImage()	Load all the images. Assign each image to an ImageView. Then call setKeyFrame for each imageView. Setup the timeline to play the animation indefinitely.
+ static void setKeyFrame()	Set up the handAnim to show different ImageView at different points in time.

2. Package Gui

2.1. Class Card

2.1.1. Fields

- String cardID	The ID which represents the type of the cards. First digit → shape Second digit → color Third digit → shading Fourth digit → number ‘9999’ means an empty card.

2.1.2. Constructors

+ Card(String cardID)	Set the cardID of this card to be the parameter.

2.1.3. Methods

+ String getCardID()	Return the cardID of this Card.

2.2. Class CardCell

2.2.1. Fields

- Card card	The Card of this CardCell.
- int position	Position of this CardCell in the playingCardList (in the GameMonitor). Value ranges from 0 – 11
- Canvas canvas	Canvas for this card
- GraphicsContext gc	GraphicsContext for this card for drawing the image of the cards according to the cardID
- int width	Width of this CardCell, set as 180.
- int height	Height of this CardCell, set as 120.
- int cornerRadii	CornerRadii of this CardCell, set as 8.
- Color unhighlightColor	Background Color of this CardCell when the card is not selected, set as white.
- Color highlightColor	Background Color of this CardCell when the card is selected, set as grey.
- String shape	Shape information of the card of this Cardcell, extracted using the method stripInfo.
- Color color	Color information of the card of this Cardcell, extracted using the method stripInfo.
- String pattern	Shading pattern information of the card of this Cardcell, extracted using the method stripInfo.
- int number	Number information of the card of this Cardcell, extracted using the method stripInfo.
- EventHandler cardHandler	An Event handler for the mouse clicked event. It will calls the method onclickHandler.

2.2.2. Constructors

+ CardCell (Card card, int position)	Set the card and position of this Cardcell. Set the preferred witdth and height according to the fields. Create a new canvas with the size of width and height and add the canvas to the CardCell. If the card is not a ‘9999’ empty card, then call the method stripInfo to get all the information about the cards from the cardID. Then draw on the GraphicsContext of the created canvas according to the info stripped. Add an onclickEvent handler. Set the background of the card to unhighlightColor.

	Otherwise, just set the background of the card to black.

2.2.3. Methods

+ void stripInfo (Card card)	Set the 4 fields: shape, color, pattern and number according to the cardID of the card.
+ void changeCard (Card card)	This method changes the card of this CardCell into another card from the deck (including an empty-card). Call stripInfo. Clear the old graphicsContext. Set the background into black if it's an empty card, otherwise setit to unhighlightColor. Redraw the graphicsContext of the CardCell accordingly. It is called when the player shuffle the board.
+ void changeCard ()	This method changes the card of this CardCell into an empty card when a correct set is found and there's no card left in the deckPile. Set the card into a new empty card. Clear the old graphicsContext. Set the background into black. Remove the onclickHandler of this Cardcell.
+ void draw (GraphicsContext gc)	Decide what shapes to draw this CardCell. Call drawRoundRect if it's oval. Otherwise call drawDiamondOrJelly
+ void drawRoundRect (GraphicsContext gc)	Draw an oval according to the data from stripInfo
+ void drawDiamondOrJelly (GraphicsContext gc)	Draw either diamond or squiggle (jelly) shape by displaying the images from the resource folder according to the data from stripInfo.
- void onClickHandler ()	Calls the static method updateCardStatus in the GameMonitor to update this CardCell.
+ void highlight ()	Change the background of this CardCell to the highlightColor.
+ void unhighlight()	Change the background of this CardCell to the unhighlightColor.
+ int getPosition()	Returns the position of this CardCell

+ Card getCard()	Returns the Card of this CardCell
+ Canvas getCanvas()	Returns the canvas of this CardCell
+ GraphicsContext getGc()	Returns the GraphicsContext of this CardCell
+ EventHandler getCardHandler()	Returns the cardHandler of this CardCell.
+ void setPosition (int position)	Set the position of this Cardcell to position.

2.3. Class CardBoardPane

2.3.1. Fields

- String img_path	The path to the background of the board. There are 4 backgrounds, which will be set in the constructors.

2.3.2. Constructor

+ CardBoardPane()	Setup the CardBoardPane with the size (1010 x 700) Set Alignment to center. Set both Vgap and Hgap to 30. Set the img_path according to the level. Set the background with the image from img_path. Add 12 CardCells from the playingCardList into this CardBoardPane by making a 3 x 4 grid.

2.4. Class NavButton

2.4.1. Constructor

+ NavButton (String text, int width, int height, int cornerRadii, int borderWidth, Color highlightColor)	Create a button with text. Set the preferred width and height. Set the font of the button text to 20 if height is greater than 60, otherwise 15. Call setupBasicLook.

2.4.2. Methods

+ void setupBasicLook (int cornerRadii, int borderWidth, Color highlightColor)	Setup the borderstroke, textFill color, background. Set onMouseEntered and onMouseExited visual changes.

+ void linkSetup (Scene nextScene)	Set the button onClickEvent to be changing the scene into the nextScene.

2.5. Class MenuPane

2.5.1. Fields

- Label scoreLabel	Label showing the current score (# of SETs found)
- Label levelLabel	Label showing the current level
- NavButton shuffleButton	Shuffle Button that will shuffle the cards on the board for the player. See GameMonitor.Shuffle
- NavButton mainMenuButton	A button that will bring the player back to the main menu scene.

2.5.2. Constructor

+ MenuPane()	Set the preferred width x height to 100 x 1010. Set Alignment to Pos.Center. Set Background to black. Instantiate scoreLabel and levelLabel. Setup the two labels with setScoreLabelText and setLevelLabelText. Instantiate two NavButtons: shuffleButton and mainMenuButton. Link the shuffleButton with the shuffle method from the GameMonitor. Link the mainMenuButton with menuScene and also clear the selectedList once clicked. Add all the components to the this menupane.

2.5.3. Methods

+ void setScoreLabelText()	Set the scoreLabel: SET(s) Found: + GameMonitor.getScore() + /4 Example: SET(s) Found: 3/4
+ void setLevelLabelText()	Set the levelLabel: Level: GameMonitor.getLevel() Example: Level: 2

3. Package Logic

3.1. Class CardUtility

3.1.1. Methods

+ ArrayList<Card> getStartingDeck()	Returns a starting deck for each level. The numbers of cards in the deckfor each level is level1: 1*2*2*3 level2: 3*3*2*2 level3: 3*2*3*2 level4: 3*3*3*3
+ Card pickRandomCard()	Pick a random card from the current deckPile. Remove that card from the deck. Return the randomized Card.

3.2. Class SoundUtility

3.2.1. Fields

+ AudioClip menuPageSound	Audioclip for main menu scene.
+ AudioClip illegalSetSound	Audioclip to be played when the player selected 3 cards, but turned out to be an invalid SET.
+ AudioClip correctSetSound	Audioclip to be played when the player found a correct SET.
+ AudioClip unselectCardSound	Audioclip to be played when the player unselect a selected CardCell.
+ AudioClip selectCardSound	Audioclip to be played when the player select a CardCell.
+ AudioClip shuffleSound	Audioclip to be played when the player clicked the SHUFFLE button.
+ AudioClip winSound	Audioclip to be played in the congratsScene.

3.2.2. Methods

+ playMusic (AudioClip audioclip)	If the audioclip is the menuPageSound, play indefinitely. Play the audioclip.

3.3. Class GameMonitor

3.3.1. Fields

+ ArrayList<Card> deckPile	Stores all the cards that are currently left in the deck
+ ArrayList<Card> discardPile	Stores all the cards that are completed as SETs and discarded from the playingBoard
+ ArrayList<CardCell> playingCardList	Stores all 12 CardCells each representing the cards on the board
+ ArrayList<CardCell> selectedCardList	Stores the currently selected Card. If the player 1) unselects a card 2) selected 3 cards 3) shuffle 4) go back to main menu , remove card(s) from this list.
+ CardBoardPane cardBoardPane	The only cardBoardPane of the boardScene
+ MenuPane menuPane	The only menuPane of the boardScene.
+ int level	Current level of the game.
+ int score	Current number of SETs found.

3.3.2. Methods*

*Playing music details are omitted. Please look at the SoundUtility

*Getters and Setters omitted.

+ void startGame (int currentlevel)	If the deckPile is null, instantiates it and set it to a startingDeck. If discardPile is null, instantiates it. Otherwise, clear it. If playingCardList is null, instantiates it. Otherwise, clear it. If selectedCardList is null, instantiates it. Otherwise, clear it. Set the score to 0 Add 12 new CardCells to playingCardList If menuPane is null, instantiates it. If cardBoardPane is null, instantiates it. Otherwise, instantiates it then clear all the nodes in the root2 of Main. Then add menuPane and cardBoardPane to root2. Then set the scene to boardScene.

	Set the scores and levels text in the menupane.
+ void removeCardFromDeck (int index)	Remove the card from deckPile
+ void removeCardFromSelectedList (CardCell cardCell)	Remove the specified cardCell from the selectedCardList and unhighlight that cardCell.
+ void clearSelectedList()	Empty the selectedCardList. Unhighlight all the highlighted cardCells.
+ void updateCardStatus (CardCell cardCell)	This method is called whenever a cardCell is clicked. If the cardCell is already in the selectedCardList, then remove it from the selectedCardList. Otherwise add it to the selectedCardList and highlight the cardCell. Then check if there are 3 selected cards or not. If it's 3, then call the method isProperSet to check if it's a legal set. Then call correctSetHandler or wrongSetHandler accordingly.
+ void correctSetHandler()	This method is called when the player found a correct a SET. Start a thread that do the following: Make all other non-empty cardCells unclickable for 0.5s Increase score by 1 Remove the 3 cards from the playingCardList and change it into 3 new cards if there are still cards in the deck. Add the 3 selected cards to the discardPile. Update the score label text in the menupane If the score is 4, Increase the level by 1. Clear the 12 GraphicsContext of the 12 cardCells. Switch the scene to a continue Scene or congratulation scene (if it's the last stage) Empty the selectedCardList
+ void wrongSetHandler()	Start a thread that do the following:

	Make all other non-empty cardCells unclickable for 0.3s Unhighlight all the cards. Empty the selectedCardList.
+ boolean cardExist(CardCell cardCell)	Returns true if the cardCell is in the selectedCardList.
+ boolean isProperSet()	Returns true if the 3 cards in the selectedCardList is a correct SET.
+ void shuffle()	Start a thread that do the following: Make all other non-empty cardCells unclickable for 0.5s Disable the shuffleButton itself for 0.5s Empty the selectedList. Shuffle the cards on the playing boards.

4. Package Application

4.1. Class Main

4.1.1. Fields

+ Stage window	Stage of the program
+ Scene menuScene	Show the game logo, 3 NavButtons, the background animation
+ Scene tutorialScene	Show how to play the game
+ Scene boardScene	Show menuPane and cardBoardPane
+ Scene creditsScene	Show info about the game
+ Scene continueScene	The continue Scene between each levels, showing two options for the player, continue or quit.
+ Scene congratsScene	Congratulations scene shown when the player cleared the game. Have a ‘return to main menu’ button.
- NavButton button_Tutorial	Once clicked, show the tutorialScene
- NavButton button_Play	Once clicked, show the boardScene
- NavButton button_Credits	Once clicked, show the creditsScene
+ Vbox root2	The main container for the boardScene. It will holds the menuPane and the CardBoardPane

4.1.2. Methods

+ void start (Stage stage)	Set the window to the stage. Initiliaze the game level 1. Setup all the scenes. Setup all the 3 NavButtons. Link the 3 NavButtons to each scenes. Play the background Animation Play the main menu music. Set the scene to be the menuScene Show the stage.
+ Scene continueSceneSetUp()	Return a set up continue scene with 2 NavButtons: QUIT and CONTINUE.
+ Scene congratsSceneSetUp()	Return a set up congrats scene with a NavButton: BACK TO MENU Stop the mainmenu song. Play end game song.
+ Scene creditsSceneSetUp()	Return a set up creditsScene. Set up a BACK TO MENU NavButton. Set the background to creditPage.png image from res folder.
+ Scene tutorialSceneSetUp()	Return a set up tutorialScene. Set up a BACK TO MENU NavButton. Set the background to rules.png image from res folder.
+ void main (String[] args)	Launch