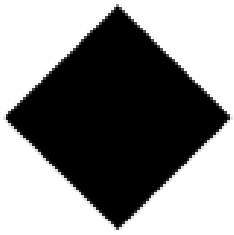


Software Masters

Software Design Document

July 25, 2016

Team

Rachel Glockenmeier

Marc Julian

Rob Miceli

John Parrish

Melisa Satili

Heather Zellner

Software Masters

Software Design Document

Contents

---

1. Class Diagram

The UML class diagram is shown in Figure 1. It contains the class associations, aggregations, and specializations for the Wheel of Jeopardy design. A larger version of this image is on the next page.

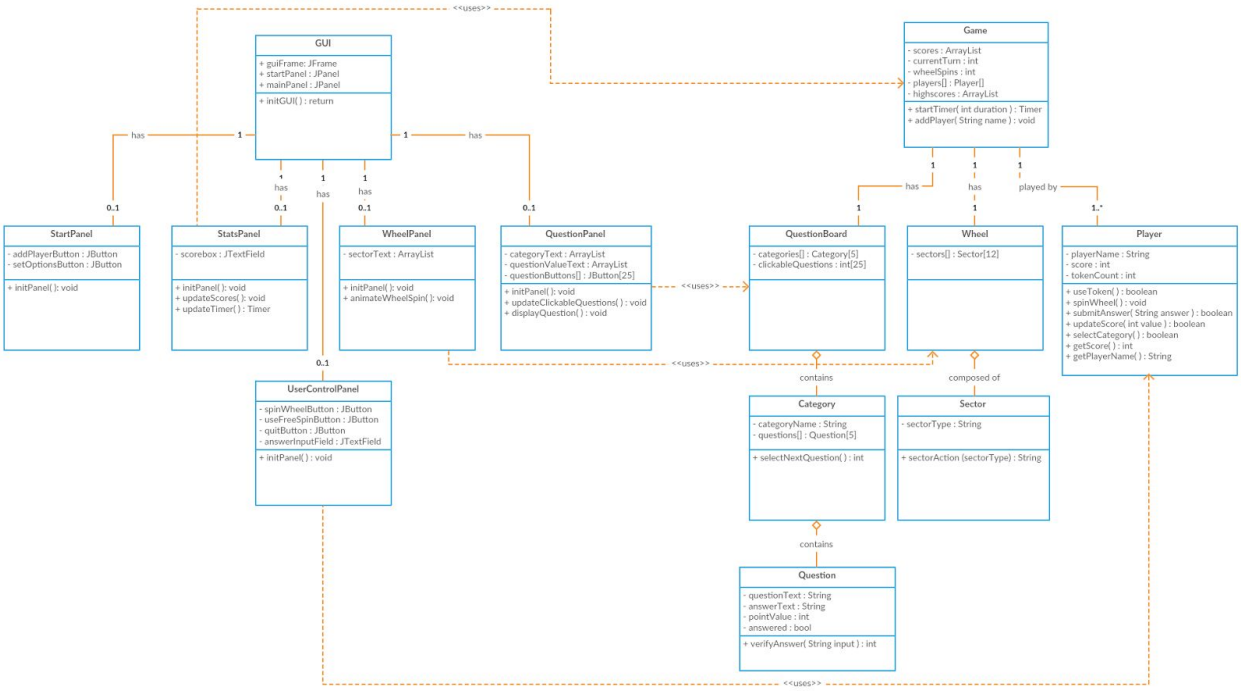
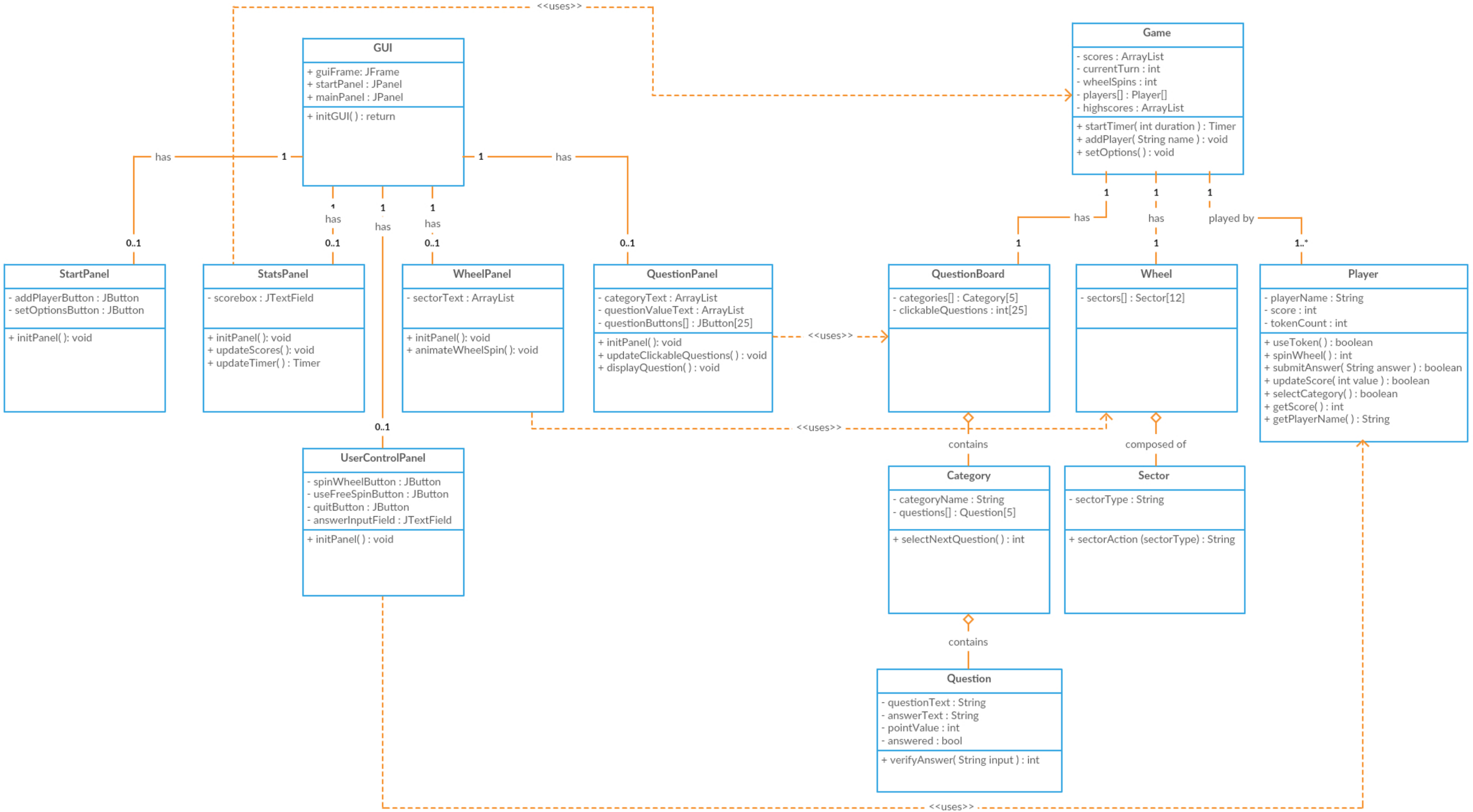


Figure 1: UML class diagram for the Wheel of Jeopardy


GUI
+ guiFrame: JFrame
+ startPanel : JPanel
+ mainPanel : JPanel
+ initGUI( ) : return
Game
- scores : ArrayList
- currentTurn : int
- wheelSpins : int
- players[] : Player[]
- highscores : ArrayList
+ startTimer(int duration) : Timer
+ addPlayer(String name) : void
+ setOptions( ) : void
<<uses>>
has
played by
1
0..1
1..*
StartPanel
- addPlayerButton : JButton
- setOptionsButton : JButton
+ initPanel( ): void
StatsPanel
- scorebox : JTextField
+ initPanel( ): void
+ updateScores( ): void
+ updateTimer( ) : Timer
WheelPanel
- sectorText : ArrayList
+ initPanel( ): void
+ animateWheelSpin( ): void
QuestionPanel
- categoryText : ArrayList
- questionValueText : ArrayList
- questionButtons[] : JButton[25]
+ initPanel( ): void
+ updateClickableQuestions( ) : void
+ displayQuestion( ) : void
QuestionBoard
- categories[] : Category[5]
- clickableQuestions : int[25]
Wheel
- sectors[] : Sector[12]
Player
- playerName : String
- score : int
- tokenCount : int
+ useToken( ) : boolean
+ spinWheel( ) : int
+ submitAnswer(String answer) : boolean
+ updateScore(int value) : boolean
+ selectCategory( ) : boolean
+ getScore( ) : int
+ getPlayerName( ) : String
UserControlPanel
- spinWheelButton : JButton
- useFreeSpinButton : JButton
- quitButton : JButton
- answerInputField : JTextField
+ initPanel( ) : void
contains
composed of
Category
- categoryName : String
- questions[] : Question[5]
+ selectNextQuestion( ) : int
Sector
- sectorType : String
+ sectorAction (sectorType) : String
Question
- questionText : String
- answerText : String
- pointValue : int
- answered : bool
+ verifyAnswer(String input) : int

2. Class Design

GUI Classes

GUI

Attributes	
guiFrame	A JFrame variable that encapsulates the entire GUI
startPanel	A JPanel variable that contains the start menu. This is where players will setup the game and start game play.
mainPanel	A JPanel variable that contains the main game panel. This is where the players will interact and play the game.
Operations	
initGUI( )	Method that initializes the GUI.
Connections	
Has a StartPanel, StatsPanel, WheelPanel, QuestionPanel, and UserControlPanel	

StartPanel

Attributes	
addPlayerButton	A JButton that will add a new player to the game. This will trigger the addPlayer() method in the Game class
setOptionsButton	A JButton that will allow the player to edit the game options.
Operations	
initPanel( )	Method that initializes the start panel on the GUI.
Connections	
Part of a GUI	

StatsPanel

Attributes	
scorebox	JTextField that contains the scores of each player, the players' names, the question timer, the number of wheel spins
Operations	
initPanel( )	Method that initializes the stats panel on the GUI.
updateScores( )	Method to update the scorebox. It polls the Game class to obtain the relevant variables.
startTimer( )	Method that causes the timer in Game class to begin.
Connections	
Part of a GUI. Uses the Game class	

UserControlPanel

Attributes	
spinWheelButton	JButton that results in the wheel spinning when pressed. It will trigger the animateWheelSpin() and spinWheel() methods.
useFreeSpinButton	JButton that allows the player to use their free spin token. It will trigger the useToken() method.
quitButton	JButton that allows the player to quit the game. It will trigger a pop-up dialog that confirms that the player wants to quit.
answerInputField	JTextField that accepts input for each question. Will only be active when the player is answering a question.
Operations	
initPanel( )	Method that initializes the user control panel on the GUI.
Connections	
Part of a GUI. Uses the Player class	

WheelPanel

Attributes	
sectorText	An ArrayList of strings representing the text for each sector
Operations	
initPanel( )	Method that initializes the wheel panel on the GUI.
animateWheelSpin( )	Method that animates the wheel to spin when a player clicks to spin it.
Connections	
Part of a GUI. Uses the Wheel class	

QuestionPanel

Attributes	
categoryText	String variable for category names
questionValueText	String variable for question point value
questionButtons	JButton to select a question from the board
Operations	
initPanel( )	Method that initializes the question panel on the GUI
updateClickableQuestions( )	Method that updates the question panel to reflect which questions have yet to be answered.
Connections	
Part of a GUI. Uses the QuestionBoard class	

Game Classes

Game

Attributes	
scores	An ArrayList containing the scores of each player. The index of the array will correspond to the player number.
currentTurn	An integer variable that corresponds to the current player's turn
round	An integer corresponding to the round number. 1 is for round 1
wheelSpins	An integer variable to keep track of the number of wheel spins that have occurred so far in the game. The max number of wheel spins in a round is 50.
players[]	An array of Player objects that keeps track of all the players currently in the game.
highscores	An ArrayList that contains strings of player names and their score. The variables are sorted from highest to lowest scores. Contains a maximum of 10 variables.
Operations	
startTimer(int duration)	Method to start a Timer object
addPlayer(String name)	Method to create a new Player instance
Connections	
Has a QuestionBoard, Wheel, and Players. Used by StatsPanel	

QuestionBoard

Attributes	
categories[]	An array of six Category objects. This class maintains pointers to each of the Category objects created to represent the categories on the question board.

Operations
None
Connections
Contains an aggregation of 5 Categories. Child class of Game. Used by QuestionPanel

Category

Attributes	
categoryName	A string variable that holds the name of the category represented by this object.
questions[]	An array of six Question objects. This class maintains pointers to each of the Question objects associated with this category.
Operations	
selectNextQuestion()	Method that selects the next unanswered question from this category.
Connections	
Contains an aggregation of 5 Questions. Aggregated to form a QuestionBoard	

Question

Attributes	
questionText	A string variable that holds the question text.
answerText	A string variable that holds the answer text for this question.
pointValue	An integer variable that holds the point value associated with this question.
answered	A boolean variable describing whether the question has been answered yet or not.
Operations	

verifyAnswer(String input)	Method that checks the player's answer to the question against the correct answer.
Connections	
Aggregated to form a Category	

Wheel

Attributes	
sectors[]	An array of twelve Sector objects. This class maintains pointers to each of the Sector objects associated with the wheel.
Operations	
None	
Connections	
Aggregation of 12 Sectors. Used by WheelPanel	

Sector

Attributes	
sectorType	A string variable that holds the type of sector from the wheel represented by this object. Sectors include: six board categories, bankrupt, player's choice, opponents' choice, lose a turn, free spin, and spin again.
Operations	
sectorAction(sectorType)	Method that performs the associated action for the input variable sectorType.
Connections	
Aggregated to form a Wheel	

Player

Attributes	
playerName	A string variable that holds the player's name.
score	An integer variable that keeps track of a player's current score.
tokenCount	An integer variable that keeps track of how many free spin tokens a player has.
Operations	
useToken( )	Method that performs the associated actions for using a free spin token, such as decrementing the player's token count and allowing the player to spin the wheel again.
spinWheel( )	Method to generate a random number between 1 and 12 inclusive. The method will return an integer corresponding to the sector number.
submitAnswer(String answer)	Method to submit and verify an answer. Returns true if the input string matches the question answer. Otherwise, it returns false. The java method contains() will be used to check if the player answer matches the question answer.
updateScore( )	Method that updates the player's score based on whether they answered a question correctly or incorrectly.
selectCategory( )	Method that allows the player to select a category, when the wheel lands on "Player select" or "Opponent select"
getScore( )	Method that returns the player's score via the score variable.
getPlayerName( )	Method that returns the player's name via the playerName variable.
Connections	
Child class of Game. Used by UserControlPanel	

3. Activity Diagram

The activity diagram in Figure 2 below shows the major scenarios of the Wheel of Jeopardy system.

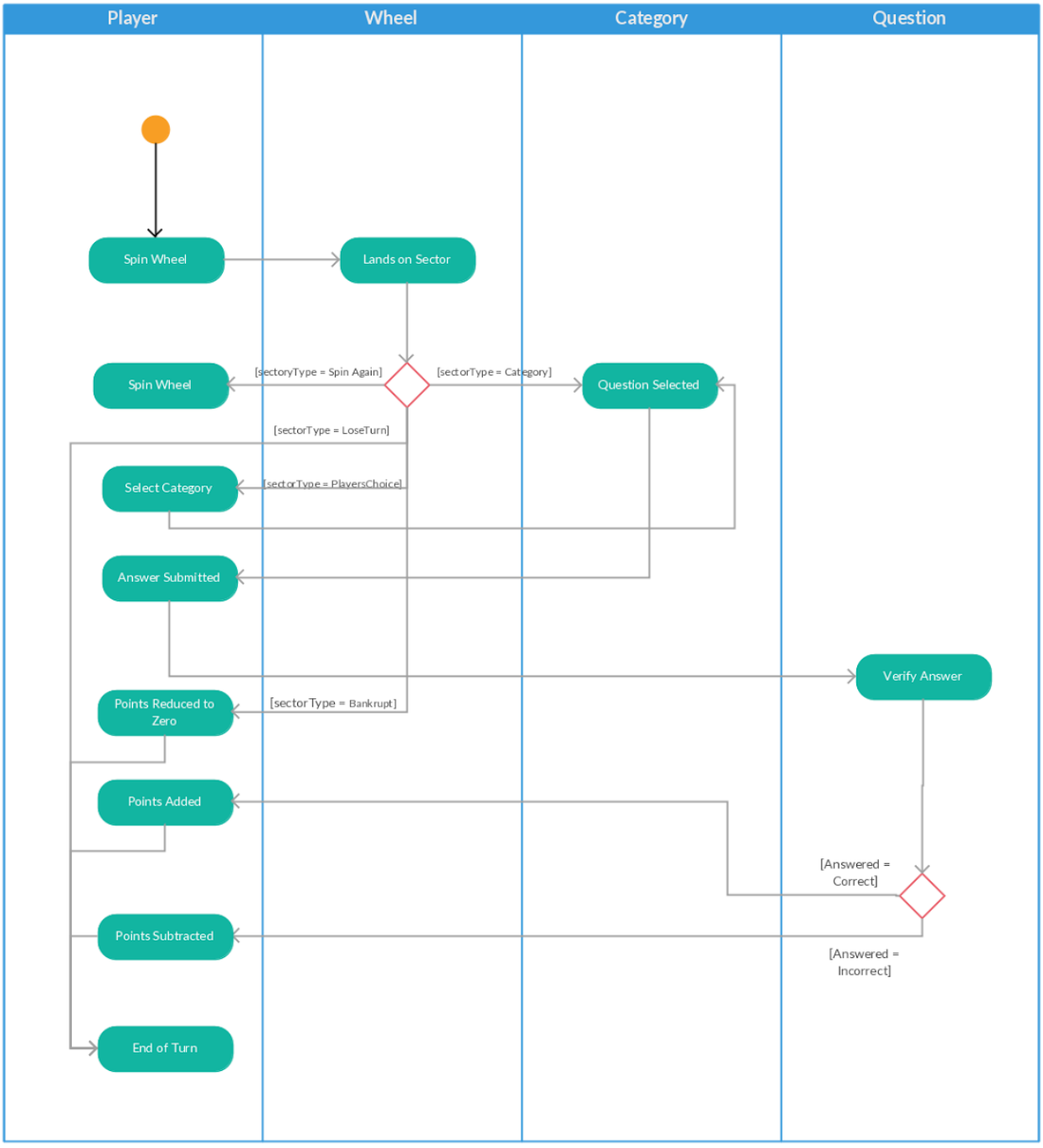


Figure 2: Activity Diagram for the Wheel of Jeopardy