

July 23, 2018

605.601.83.SU18

Foundations of Software Engineering

Software Architecture Addendum

Kevin HIcks;Fiona Lyons;Jennifer Walker

# Subsystems Summary

In Wheel of Jeopardy, the subsystems are as follows, and will each be described in detail:

* Game
* Player
* Wheel
* Slice
* Board
* Category
* Question

## Game Subsystem

The Game subsystem represents the game of Wheel of Jeopardy itself. It is responsible for the overall flow of the game, such as taking turns and awarding points, and for displaying information about the game. This is the main controller for the game.

|  |  |
| --- | --- |
| Information | Interfaces |
| * How many players are in the game * Whose turn it is * Which round it is * How many turns are left in the round | * Display player information * Display turn information * Display round information * Switch rounds * Take turns * Display Wheel * Display Board * Declare winner |

## Player Subsystem

The Player subsystem is responsible for maintaining and modifying data about a player in the game. This includes maintaining and changing their score, and keeping track of tokens.

|  |  |
| --- | --- |
| Information | Interfaces |
| * Round score * Total score * Tokens | * Award points * Decrement points * Award token * Double your score * Go bankrupt |

## Wheel Subsystem

The Wheel subsystem represents the wheel in Wheel of Jeopardy. The wheel can be spun, and according to where it lands, different information will be passed to other subsystems.

|  |  |
| --- | --- |
| Information | Interfaces |
| * Slices of wheel | * Spin the wheel * Pass information on selected slice |

## Slice Subsystem

The Slice subsystem represents a slice on the Wheel subsystem. The slice can contain different data depending on the type, such as a bankruptcy or free turn slice.

|  |  |
| --- | --- |
| Information | Interfaces |
| * Type of slice | * Pass type information |

## Board Subsystem

The Board subsystem represents the Jeopardy! board in the Wheel of Jeopardy game. The board is responsible for displaying questions and their values, and keeping track of which questions have already been answered. If all questions have been answered, the board needs to notify the Game that the round needs to be switched.

|  |  |
| --- | --- |
| Information | Interfaces |
| * Categories on board | * Get categories * Get questions * Answer question * Pass points information * Is empty |

## Category Subsystem

The Category subsystem represents a category on the Board in the game. The Category contains the Questions within that category, and must keep track of them.

|  |  |
| --- | --- |
| Information | Interfaces |
| * Category name * Questions in category | * Get category name * Get questions * Get current question |

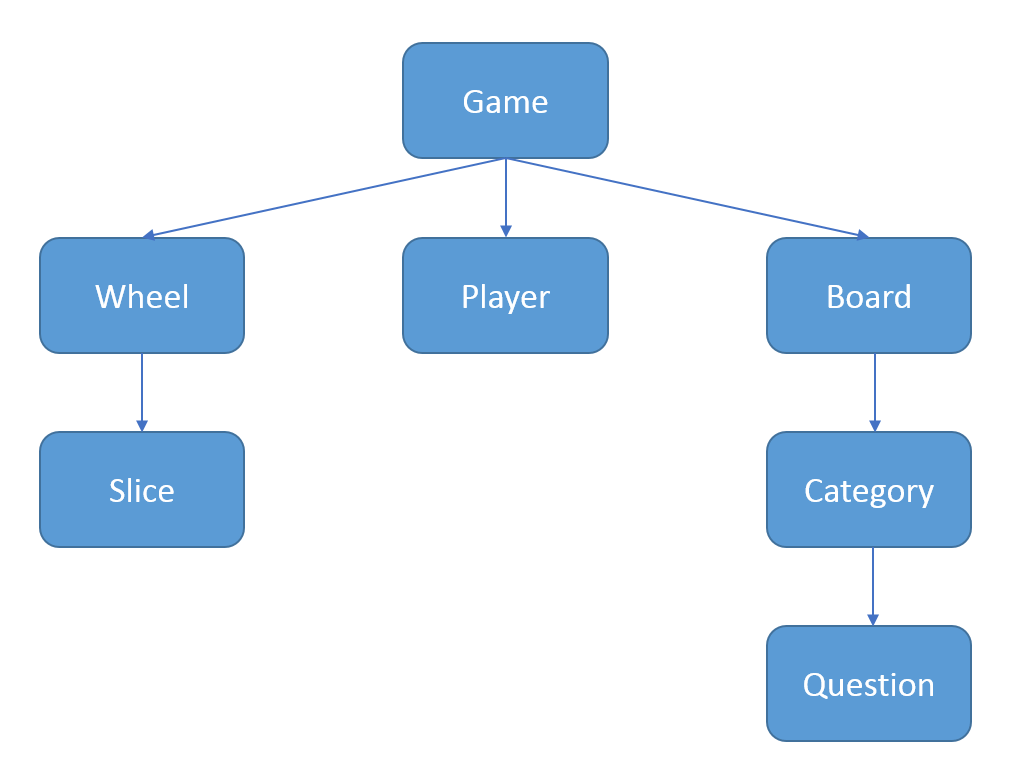
## Question Subsystem

The Question subsystem represents a question within a Category within a Board. The question must keep track of whether it has been answered or not, and how much that answer is worth.

|  |  |
| --- | --- |
| Information | Interfaces |
| * Question * Answer * Has been answered * Point value | * Get question * Get answer * Get point value * Answer question |

# Structural Model

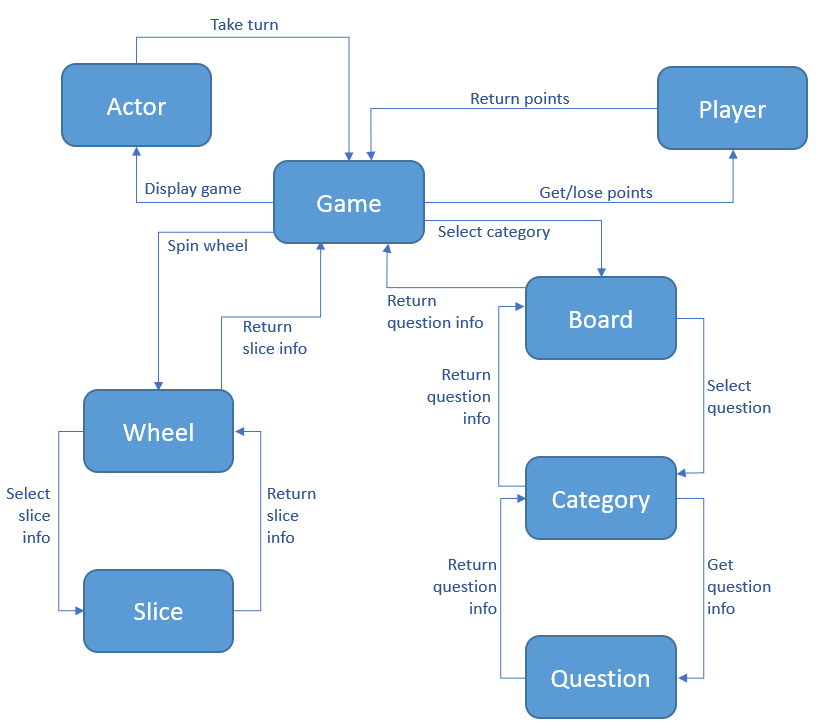
The structure of the Wheel of Jeopardy game is hierarchical, and is graphically represented as follows.



Each subsystem in the tree is dependent on the subsystem above it, until the Game subsystem, which the player actor interacts with directly. The subsystems are partitioned to control who has access to what data. For example, the Board subsystem has no need to directly access the information in the Wheel or any of the Slices, and instead the Game controls any information flow between the two.

# Information Model

To further enforce the separation of roles and responsibilities of each of the subsystems, the following information model is provided.



This model is non-exhaustive and includes the major pieces of information passed between the subsystems. This model also includes an Actor, who interacts with the system; this is a user of the game.

Like in the structural model, subsystems that are not connected to not directly share information. This way, the passing and modification of information is very carefully managed and there is low coupling within the system as a whole.