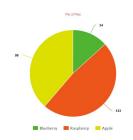
## Framework for Entertainment Data Collection and Visualization

Jiachen Liu, Julia Kim, Sabyasachi Mohanty







## Framework Domain

Any media that has a title, videos and reviews/comments such as movies, tv shows, video games and music.

- Example data plugins:
  - Movie information scraped from imdb.com.
  - Video games with twitch gameplay and ign reviews.
  - Music reviews from Last.FM with a music video.

- Example visualization plugins:
  - **Web page** that introduces the media with a video and other information.
  - **Wordcloud** of most frequent words in the review/comment.
  - Pie chart with sentiment analysis.

## Generality and Specificity of Framework

#### Abstractions:

- The data and visualization plugins are not coupled; the framework and framework GUI controls the flow of the project
- The data is represented as an abstract class (MediaInfo)

#### Reusability:

- Framework can be reused for a variety of data and visualization plugins
- Sentiment analysis can be conducted for all media

#### Flexibility of plugins:

- Data plugins can take in data from any source, given that the data represents some form of entertainment with a title, reviews/comments, and an associated video
  - For instance, a song typically has a title and a music video with comments
- Visualization plugins can visualize any data processed by the data plugin

## **Project Structure**

#### Main class

#### Core

- MediaFramework
- MediaInterface

#### GUI

FrameworkGUI

#### Plugins

- Data Plugins (extend Media Plugin Interface):
  - MovielMDBPlugin
  - MovieRottenTomatoPlugin
  - VideoGameIGNPlugin

#### Plugins

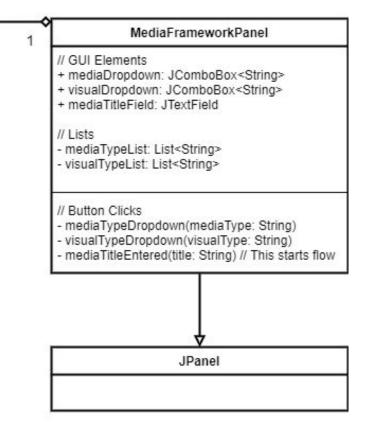
- Visual Plugins (extend VisualPluginInterface):
  - SentimentPlugin
  - WebpagePlugin
  - WordCloudPlugin

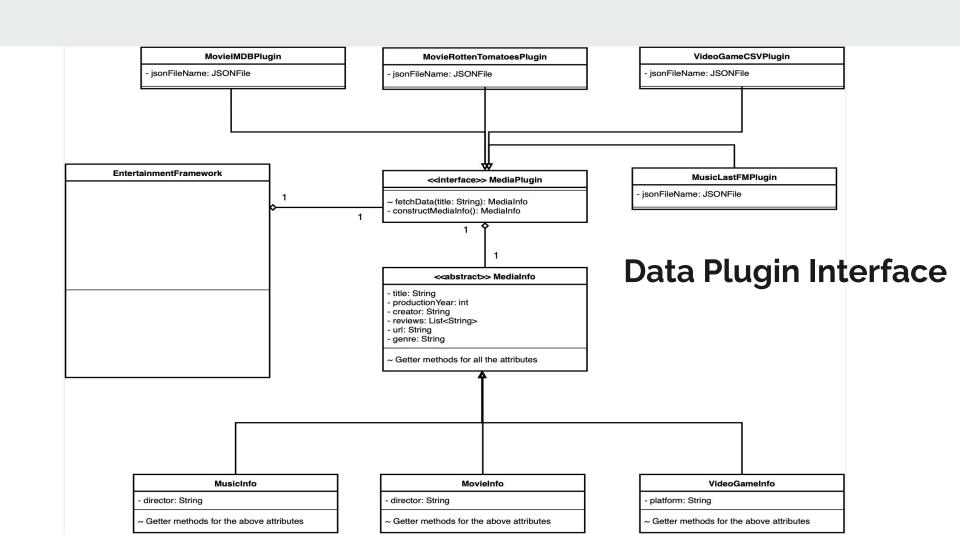
#### Test

- FrameworkTest
- PluginTest

### <<interface>> MediaFrameworkInterface // Button Clicks ~ setMediaType(mediaType: String) ~ setVisualType(visualType: String) ~ setMediaTitle(title: String) MediaFramework // Attributes - mediaPlugin: MediaPlugin visualPlugin: VisualPlugin mediaTitle: String medialnfo: Medialnfo // Maps mediaTypeToMediaPluginMap: Map<String, MediaPlugin> mediaTypeToVisualPluginMap: Map<String, VisualPlugin> // Internal initFlow() // This starts the flow // Third party API call extractSentiment(media: MediaInfo)

## Framework UML





# Visualization Plugin

