**Project name**: GamerSight  
  
**Project Description:**

The project is about providing tools to choose a video game a user would desire. Two main features of it are searching games by categories and providing helpful materials about the games. The categories to choose from are genre, year of release, critic review scores etc. There will be an option of combining and excluding them. When the player decides to view a certain game from the list, he will be given diverse information about the game. It will include official game trailers, critic comments, gameplays and recommended hardware. In addition, the GamerSight will show the price discounts if those are currently active.

As a person who likes to play video games myself, I always like to research before choosing to buy and install a game. In order to pick one I would enjoy and can afford, it is very helpful to look through videos about it on YouTube and read reviews at various gaming websites such as Steam, IGN, GameSpot. With the GamerSight I will be able to search for my game of choice easier. Plus, all the information about it, which is usually scattered throughout the internet, will be in one place.

**Project Planning:**

First, I will implement searching games by categories. This will be executed by taking the list of games from sites like Steam, GOG, and Wikipedia using BeautifulSoup. Their genres, year of release and other profile information will be taken from those sites too. Each game’s information will be handled using classes.

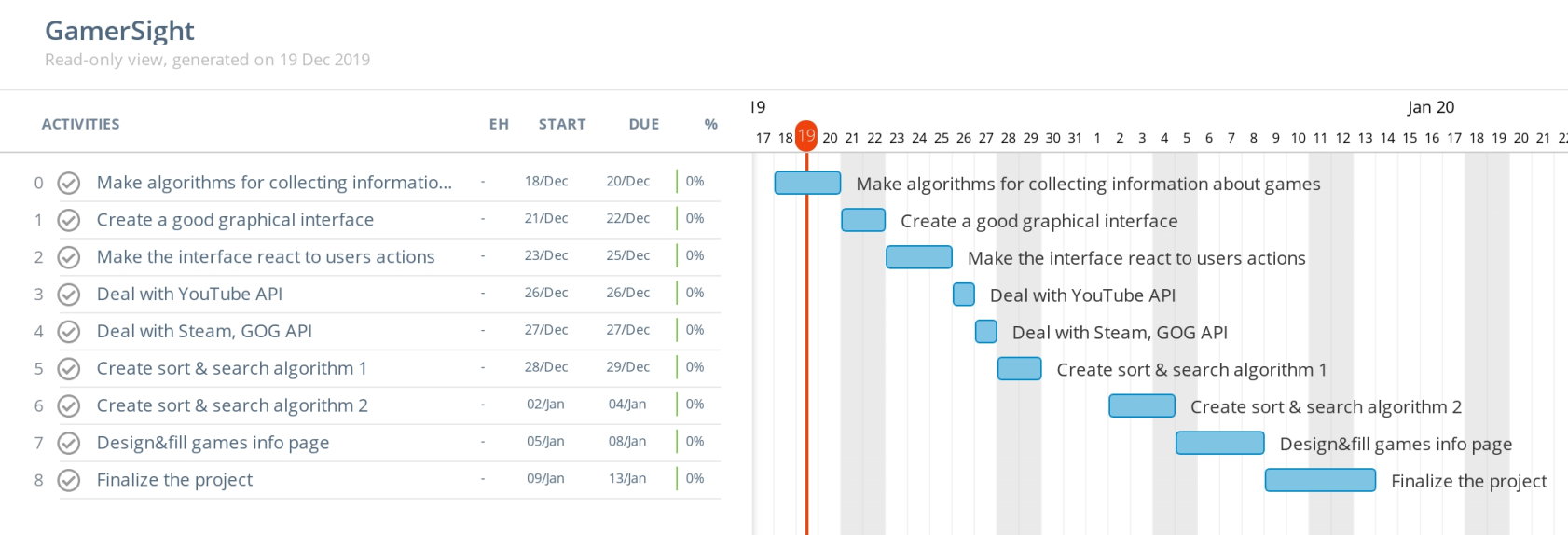
Then there is a problem of getting critic review scores, ratings, and awards. This will also be solved by using BeautifulSoup. The sites that will be used this time are IGN, Metacritic, PCGamer.

The collected information will serve as a means of searching games by categories.

Secondly, I will design a game view page layout and will fill it with game’s basic information, critic comments, awards, ratings, available discounts/special offers, official game trailers and optionally selected gameplays. YouTube API will be used to fetch trailers and gameplays. Steam’s and GOG’s APIs will be utilized to get game news, from which discounts and offers will be extracted.

The user interaction will be achieved through Python’s graphical interface. Search by categories, looking for a game will be handled in that way too. There will be fields indicating categories, which would be a specific color depending on whether they are excluded/ included/ not selected. Searching for a specific game will be possible by entering the name of the game in a special blank space.

These errors will most certainly need handling: Connection Lost, Incorrect User Input, and Unavailability of data on a web-page.

**Timeline:**

**Update 1:**

* What you have done:

I have created a search for a game by keywords. User inputs some words and the program searches the game database for best matching results and outputs first 10. After printing out the names of those games, the program asks if the user wants more results. User has to print “yes” for 10 more games to show up or “no” if he is satisfied with the games he/she sees (it doesn’t matter if some or all letters of “yes” and “no” are written in caps). If the program receives something besides the desired input, it prompts the user to input again.

After the list of games has been shown and user inputted “no”, the program asks for a number of the game the user wants to know more information about. Users inputs a number (incorrect input is handled) and such information about a game is shown: name, genres, release date, platforms, on which the game is available, developers, themes, game age rating, overview, minimum and recommended hardware requirements, as well as, the link to the web-site from which the information was taken.

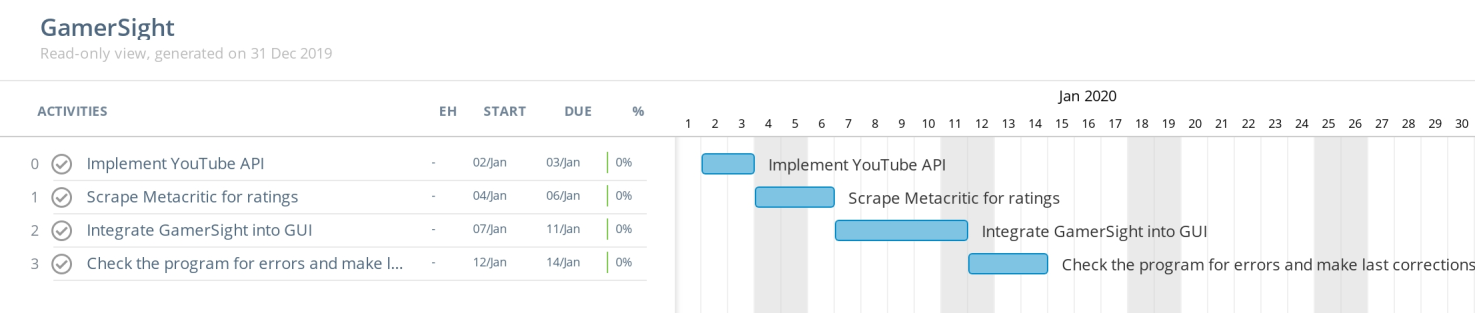
* Any changes on the final project plan, and why:

I didn’t use Wikipedia. Scraping Wikipedia for the list of games was a bad idea because the names of games were spread out on different pages with various formatting. The various formatting required scraping every page with different approach. It was not worth to make it work, because it would take up too much time.

Instead of Wikipedia I tried using API of the web-sites that provide databases of games (Steam, IGN, RAWG, GOG, IGDB, Giant Bomb). Steam does not have API that allows extracting information about games; IGN, RAWG, GOG and IGDB have complicated procedures for using their API. But Giant Bomb has an easy to use API, which allows extracting a lot of information. That is why I ended up choosing Giant Bomb as one of web-sites I use API from.

I also decided not to make the search by categories, because Giant Bomb’s API doesn’t provide sort function and it also doesn’t have an option to extract the whole database of games, for me to sort through them late­­­­r.

* Timeline for the rest:



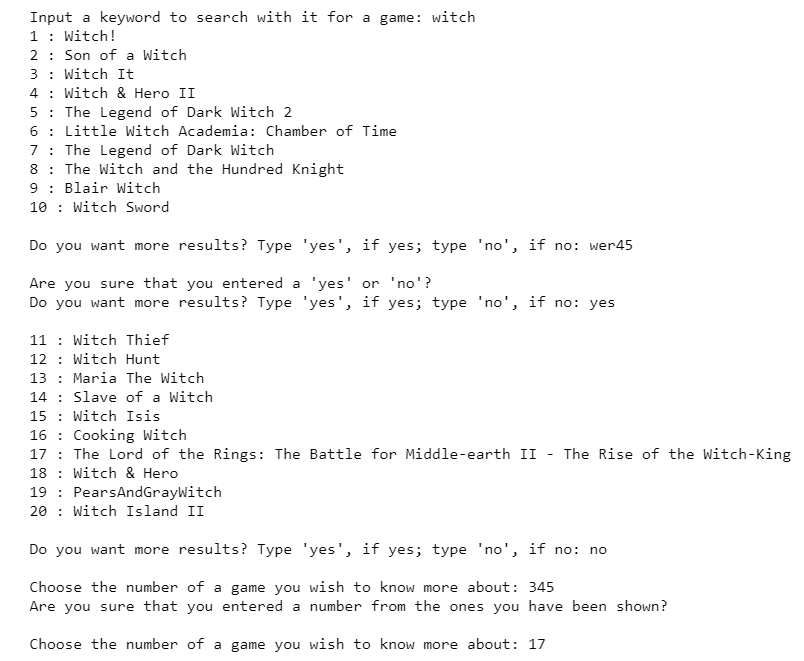
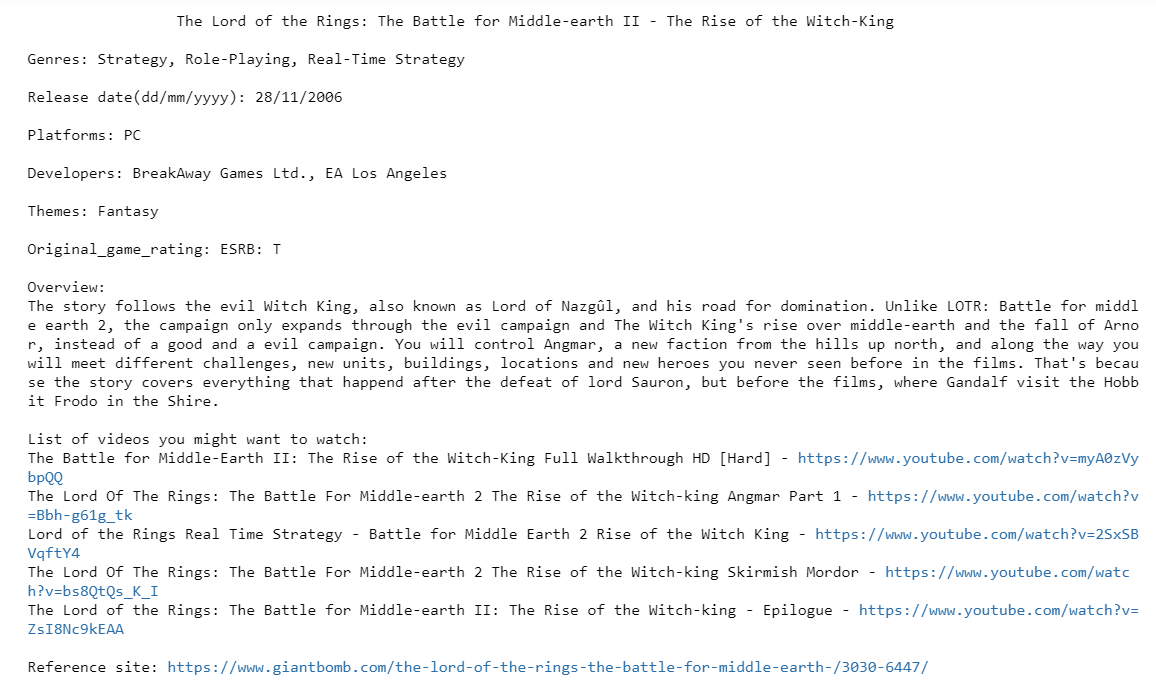
**Update 2:**

Since the last Update, I integrated YouTube API into GamerSight, made use of BeautifulSoup by scraping IGN, fixed some of the bugs, and finalized handling of the incorrect user input. Unfortunately, I also cancelled some of my previous goals: making GUI and scraping Metacritic for ratings.

YouTube: GamerSight searches YouTube for the name of a game and shows names and links to the top 5 results.

IGN: if a game has appeared in IGN’s top 100 games of all time, GamerSight notifies about that.

**Example of interaction with GamerSight:**

****

**RUN:**

No images, music files etc. are needed to run GamerSight.

The program uses 4 libraries: google-api-python-client, urllib3, beautifulsoup4, json.

To install libraries:

1. If using Anaconda. In Conda terminal write: **conda install** ***name of the library***.
2. If using Python with pip. In the OS terminal, set the path to the folder where pip is located and write: **pip install *name of the library***.

I have provided 2 files with extensions .py and .ipynb. You can run .py in Python 3 or any other environment that supports .py. You can run .ipynb in Jupyter Notebook.