# Game_Knight System

We now present a study of using ADD 3.0 for a brownfield system based on a game database system. This study examines the initial design of the game database. The study is composed of three iterations and is tied to real-world circumstances. Initially the study presents the business case, and then a summary of the requirements for the database system. A summary of the activities that are performed during the ADD iterations is done iteration by iteration and steps are included within those.

.

## 1.1 Business Case

A movie database corporation wanted to extend their reach into the game verse to achieve this they decided to investigate their competitors and decide what is necessary when building a system that focuses on games. As with their movie database they found it necessary to include an API that adds tuples of game related data in the respective tables. A poor API will greatly affect

the performance of the system as the corporation intends to connect its API to its database program. The featured tables in the database include:

- Game (game_id, rawg_game_id, fk_dev_id, name, release_date, game_img, description, metacritice_score)
- Developers (dev_id, rawg_dev_id, dev_name, dev_img)
- Genre (genre_id, rawg_genre_id, genre_name)
- Platform (plat_id, rawg_platform_id, name)
- Genre_Associated (ga_id, fk_game_id, fk_genre_id)
- Has_Tags (ht_id, rawg_game_id, game_id, tag)
- Played_On (played_on_id, fk_plat_id, fk_game_id, release_date)

To achieve the goals of the corporation, in making a game database relevant to the public functions must be developed. The model system was named Game_Knight and the functions include:

- *Sort function*. The purpose of the Sort function is to narrow the field of games the end-user views based on the prompt they have selected provided by a drop select list containing genre, rating, tag, platform, and developer name. the system is altered based on the prompt selected.
- *Expand and Collapse function*. The purpose of the Expand and Collapse function is to allow the end-user to view more information about the game they have selected and close that information when they decide to, returning to the home screen. The function is accompanied by a hover highlight to signify expandability.
- *Recommend Game function*. This function allows the end-user to receive a recommended game either based on a relation to another game, relation to a genre, or a random game.
- *Search function*. The search function allows the end-user to search a specific game title and provide a form with games either matching the search or games that are closely related. This function also allows the end-user to use all other functions within the form.
- *Display platform associated*. The function allows the user to view the platform associated to the game the end-user is viewing. This function also displays an icon when the game is in its collapse state.
- *Game stats analysis*. The function displays game information based on the prompt the user selects for example, games with the highest Metacritic score which shows a form of the top ten games with the highest Metacritic score.
- *Fault analysis function*. The goal of fault analysis function is to show, examine, and log faults that occur in the system. The system detects whether a function responds the way it is intended for example, the expand and collapse function within the system intends to show less or more information whether the function operates the server is expected to log this information.