**PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO**

**Mini MRPG 2D usando Unity/C# para Quests Dinâmicos**

**Lucas Pinheiro Nepomuceno**

**[INF1951] PROJETO FINAL II DE GRADUAÇÃO**

**CENTRO TÉCNICO CIENTÍFICO - CTC**

**DEPARTAMENTO DE INFORMÁTICA**

CURSO DE GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Rio de Janeiro,

Junho de 2019



Lucas Pinheiro Nepomuceno

Mini MRPG 2D usando Unity/C# para Quests Dinâmicas

Projeto final apresentado ao Curso Ciência da Computação como

requisito parcial para a obtenção diploma em Ciência da Computação.

Orientador: Bruno Feijó

## A System for Adaptive Multiplayer RPG based on Psychological Traits and Player Interaction

**Lucas Pinheiro Nepomuceno, Bruno Feijó**

Pontifical Catholic University of Rio de Janeiro (PUC-Rio), Department of Informatics, Rio de Janeiro, RJ, Brasil

ARTICLE INFO

Keywords:

Behaviour

Personality  
Quest Generation  
Gaming

Audience Model

Adaptive Storytelling  
Player Interaction

ABSTRACT

Most current video games base its storyline on a fixed, pre-programmed storyline, with waypoints where the story branch off based on players choices. This technique, which is a evolution of the previous linear storytelling technique used in older games, gave birth to a whole new genre of games where player decisions influence the outcome of the narrative.

However, the level of adaptability of a storyline can be improved. This project intends to develop a game that, based on the players’ psychological profile and the relationship between players, decides the best quest, from a pool of available quests. To do that, we utilize a combination of questionnaires meant to access the players’ gaming interests, as well as the players’ psychological and behavioural characteristics, based on the well-known Big Five model.

With these pieces of informations, along with information about the relationship between players, we choose which available next-quest provides the best gaming experience to that particular group of players.

## Introduction

**1.1 Motivation**

The history of gaming has evolved a lot through the years, influenced by technological advancements, a shift on the philosophy of game developing companies and player tendencies. In the early days of gaming, games didn’t need a narrative, designed *lore*, to engross the players. Video-games were treated like traditional games, like solitaire or baseball. There was no need to engage the players on the deeper meaning of a game.

As the games evolved, so did the need to engage players on the meaning of a game, or to give a satisfying goal for the player. That need created the concept of lore for a game. The lore explains the reason why a player would want to go on adventures and fight monsters. Some games don’t need lore to be enticing to players, while others may have a subpar gameplay experience, but its incredible lore make it a success. Lore is explained or presented in various ways, be it through items descriptions, dialogue or quest lines. Here we will focus on the latter one.

The way quest lines were presented in games has evolved along with the games themselves. We used to have a single line of progression, a player is sent to a place, do some action would trigger an event, sending him to another place, requiring some other action, and so forth and so on.

With the rising popularity of RPGs (Role Playing Games), players were presented with new kinds of story lines. Quests which influenced later stages of the game, or multiple paths of the story. Storytelling became more sophisticated for the entertainment of the players. The concept of *world-building* was created. Giving depth and meaning to characters, creating a backstory of events for the game world increased the players commitment to the game.

The kind of quest lines developed by the RPGs is designated nonlinear storytelling, that is, waypoints and branches in the quest line (Figure 1). That model of storytelling swept the market, because it extended the lifetime of a game, giving reasons for the players to replay a game, discover more information about the stories, experiencing other points of view. This form of narrative was very well received by the player base and spanned through years.

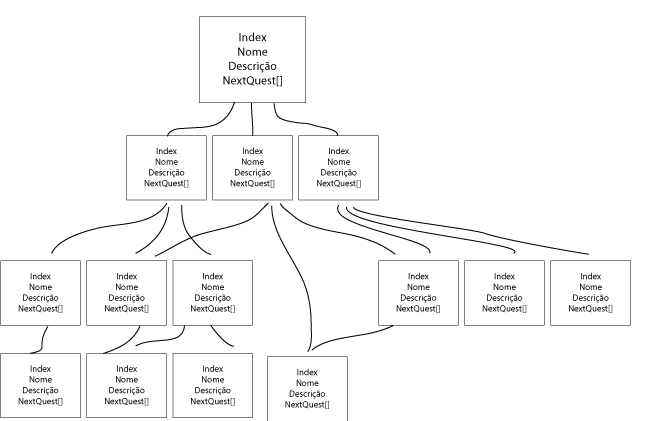


Figure 1 - Branching Representation

Recently, the philosophy of gamers and game developers has been shifting to a new form of nonlinear gameplay. Where once an action was required to generate new directions in the narrative, today some developers have decided to forego of the branching system in its entirety. Some games now use what’s called an open-world system. As soon as the player enters the gaming world, he’s free to do whatever and go wherever he wants. Some areas may be outside the players capacities, which creates a “soft” throughline for the game but the player is not impeded from following that path.

What all these games had in common, is that they were developed to please the average gamer, that is to say, they didn’t have any customization in its narrative that molded them to the type of player that was experiencing them. This creates a small disconnection to the player that may break the immersion that was intended. For example, consider the game Megaman x6 (Capcom) (Figure 2). The game is a platformer 2D that ultimately has one goal, acquire enough power ups to face and defeat the last boss Sigma. Those power ups are obtained through defeating several enemies who get more and more powerful with each item the player acquires. The game is presented as a monster slaying game, it builds that expectation. However, one quest in particular asks that a player solve a complicated puzzle that requires the player to wait several minutes to find the answer. Not only is that quest so out of place in a game design perspective, it also feels like it’s aimed for a completely different kind of player than the ones that would enjoy the kind of game that Megaman is.



Figure 2 - Megaman x6 Treasure Hunting on a Monster Slaying game

**1.2 - System Purpose**

This system intents to change the current model of multi-branch storytelling by creating a more personalized storyline for a particular group of players. To do that we plan on adapting two previous works developed at PUC-Rio. The first work [1] is the main basis and inspiration for this project. It defines a method for creating stories based on a single player psychological model and behaviour.

This system uses the player psychology and behavioural monitoring system proposed in [1] and tries to adapt it to a multiplayer setting. Additionally, this system tries to adapt Baffa et al.’s work [2]. The goal is to apply the mathematical model proposed by [2] to the algorithm proposed by [1], in order to find the most satisfying quest line for a particular group of players.

In order to do that, the system will need to access the players preferences in game types, the players personalities and their relationship with one another. Given the limited resources, the system was trimmed down to only obtain the players preferences in game types, and player personalities. The relationship between each couple of players was established by the system, but with plans on implementing a module to obtain that information in real time.

What will be presented is a 2D, top-down style game where the players will be presented with several quests to complete. These quests are defined by three characteristics:

* Level of Aggression - How many enemies do the players need to kill on this quest?
* Level of Greed - How many objectives do the players need to collect before finishing this quest?
* Level of Rescuer - How many victims need to be saved in this quest?

The system guarantees that only the kinds of quests that satisfy this particular group of players will be presented to them. It does that by electing a leader for this group. A leader is a player who has the highest overall value of empathy with every other player. It is assumed that by making the most respected player satisfied, the group will be satisfied by as well.

## Theoretical Background

This project utilizes a model highly recognized and utilized in psychology called the Big Five model, Five Factor model or OCEAN model [3]. This model surveys the users and utilizes common words to analyze a person’s personality and quantify each aspect it deems relevant. These traits are then quantified on a normalized scale [0..1]. The five factors utilized in this model are (definitions acquired from references [3] and [4]):

* + 1. **Openness (to Experiences)**

Openness is a general appreciation for art, emotion, adventure, unusual ideas, imagination, curiosity, and variety of experience. People who are open to experience are intellectually curious, open to emotion, sensitive to beauty and willing to try new things. They tend to be, when compared to closed people, more creative and more aware of their feelings. They are also more likely to hold unconventional beliefs. High openness can be perceived as unpredictability or lack of focus, and more likely to engage in risky behaviour or drug taking.Moreover, individuals with high openness are said to pursue self-actualization specifically by seeking out intense, euphoric experiences. Conversely, those with low openness seek to gain fulfillment through perseverance and are characterized as pragmatic and data-driven—sometimes even perceived to be dogmatic and closed-minded. Some disagreement remains about how to interpret and contextualize the openness factor.

* + 1. **Conscientiousness**

Conscientiousness is a tendency to display self-discipline, act dutifully, and strive for achievement against measures or outside expectations. It is related to the way in which people control, regulate, and direct their impulses. High conscientiousness is often perceived as being stubborn and focused. Low conscientiousness is associated with flexibility and spontaneity, but can also appear as sloppiness and lack of reliability. High scores on conscientiousness indicate a preference for planned rather than spontaneous behavior. The average level of conscientiousness rises among young adults and then declines among older adults.

* + 1. **Extraversion**

Extraversion is characterized by breadth of activities (as opposed to depth), surgency from external activity/situations, and energy creation from external means.The trait is marked by pronounced engagement with the external world. Extraverts enjoy interacting with people, and are often perceived as full of energy. They tend to be enthusiastic, action-oriented individuals. They possess high group visibility, like to talk, and assert themselves.Extroverted people may appear more dominant in social settings, as opposed to introverted people in this setting.

Introverts have lower social engagement and energy levels than extraverts. They tend to seem quiet, low-key, deliberate, and less involved in the social world. Their lack of social involvement should not be interpreted as shyness or depression; instead they are more independent of their social world than extraverts. Introverts need less stimulation, and more time alone than extraverts. This does not mean that they are unfriendly or antisocial; rather, they are reserved in social situations.

* + 1. **Agreeableness**

The agreeableness trait reflects individual differences in general concern for social harmony. Agreeable individuals value getting along with others. They are generally considerate, kind, generous, trusting and trustworthy, helpful, and willing to compromise their interests with others. Agreeable people also have an optimistic view of human nature.

Disagreeable individuals place self-interest above getting along with others. They are generally unconcerned with others' well-being, and are less likely to extend themselves for other people. Sometimes their skepticism about others' motives causes them to be suspicious, unfriendly, and uncooperative. Low agreeableness personalities are often competitive or challenging people, which can be seen as argumentative or untrustworthy.

Because agreeableness is a social trait, research has shown that one's agreeableness positively correlates with the quality of relationships with one's team members.

* + 1. **Neuroticism**

Neuroticism is the tendency to experience negative emotions, such as anger, anxiety, or depression. It is sometimes called emotional instability, or is reversed and referred to as emotional stability. Those who score high in neuroticism are emotionally reactive and vulnerable to stress, they also tend to be flippant in the way they express emotion. They are more likely to interpret ordinary situations as threatening, and minor frustrations as hopelessly difficult. Their negative emotional reactions tend to persist for unusually long periods of time, which means they are often in a bad mood. Moreover, individuals high in neuroticism tend to experience more negative life events, but neuroticism also changes in response to positive and negative life experiences. Also, individuals with higher levels of neuroticism tend to have worse psychological well being.

At the other end of the scale, individuals who score low in neuroticism are less easily upset and are less emotionally reactive. They tend to be calm, emotionally stable, and free from persistent negative feelings. Freedom from negative feelings does not mean that low-scorers experience a lot of positive feelings.

Neuroticism is similar but not identical to being neurotic in the Freudian sense (i.e., neurosis.) Some psychologists prefer to call neuroticism by the term emotional instability to differentiate it from the term neurotic in a career test.

These traits often vary along a person’s life but tend to stabilize once they reach adulthood. They also are mostly valid regardless of culture or locations, with some adjustments required to fit the model to some cultures (mostly cultures that are not so similar to western culture)

## Requirements

In order for the system to work as intended, it is necessary at least 2 players. Once the game starts, the players are presented with a questionnaire. The first questionnaire is meant to access each players’ preferences between types of games (violent games, rescuing games and collecting games). This questionnaire provides the necessary information about which kinds of quests should be presented to that particular group of players.

After the player type questionnaire, the players are asked 10 questions about their views on several scenarios. These questions are the same from [1] and are meant to mirror the Big Five Inventory - 10 [2](#_2vprib2kw8te) which is a tried and true questionnaire to obtain a person’s [OCEAN](#_xk0tmk71jm61) values. These answers, when applied to a gaming evaluation of a player’s psychology is denoted BFGI, Big Five Gaming Inventory.

With both values, BFGI and Player Type, the system assigns random empathy values to each pair of players. These empathy values should be obtained through observation of player interaction and refined along the gameplay but for testing and demonstration purposes they’re auto generated.

These are all the requirements for the system to work:

* [REQ001] The values for each player about their player type and game preferences
* [REQ002] The OCEAN values for each player
* [REQ003] The empathy levels between each pair of players

Some assumptions were made about the validity of these requirements, for example, we assume the questions for the BFGI questionnaire is a good adaptation for the BFI questions. We also assume that player empathy can be measured and interpreted in a way with no nuances.

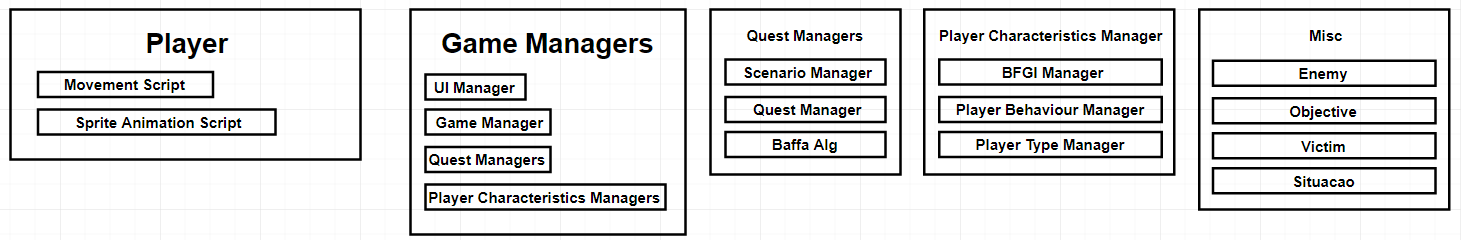
## Modelling

* 1. **System Architecture**

The unity project is comprised of a main game manager, GameManager. This manager controls the main aspects of the game (enemy velocity, power up duration, etc). It also possesses the models (prefabs) for the players, enemies, victims, rescue zones, objectives and items.

Besides the main GameManager, the project is comprised also of:

* UI Manager - Controls all the information that are displayed on the screen (player empathy, ocean values, etc)
* BFGI Manager - Big Five Game Inventory Manager, controls all the information regarding the players OCEAN values.
* Player Behaviour Manager - Keeps statistics of each player (average enemy kill time, average time between attacks, etc). Also keeps the players empathy information.
* Scene Manager - Controls the creation, processing and clean up of the scenarios and quests. Instantiates enemies, objectives, victims, etc.
* Quest Manager - Holds the list of existing quests, as well as the current quest being shrubles
* Baffa Alg - Called once the current quest is complete. Runs through the list of following quests, and finds the quest that best pleases all players.
* Player Type Manager - Handles the questionnaire and store the values of the player types (preference for violent games, rescuer games or collector games).

Figure 2 - System’s main modules

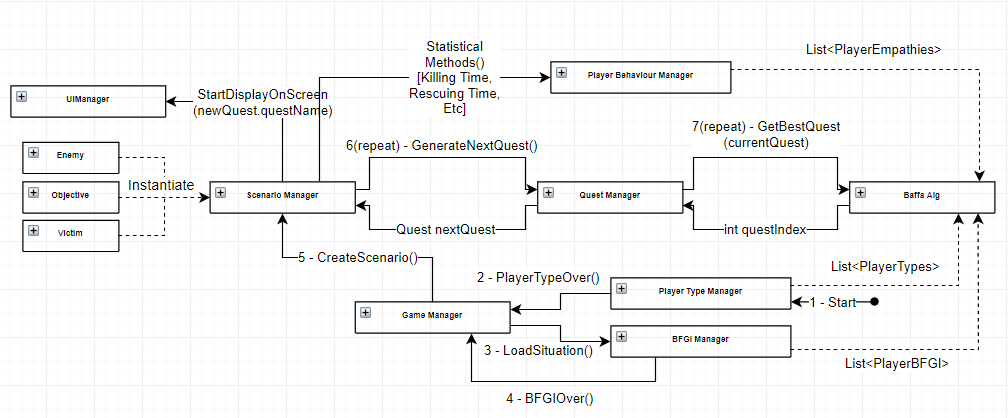


Figure 3 - Class Relation Diagram

* 1. **Project Sequence**

Once the game is started, the GameManager call the methods from Player Type Manager to create a questionnaire to access the Player Type. As long as no answer is given, the GameManager checks player inputs for a start command to add a new player (up to 5).

After the number of players is defined (once the first answer of the questionnaire is given), the players fill the questionnaire to define their Player Type. After having finished the questions (3 questions), the players are taken to the BFGI questionnaire. These questions are meant to access the player’s Openness, Consciousness, Extraversion, Agreeableness and Neuroticism values.

After finishing the BFGI, random values are assigned for each pair of players empathy levels (future versions are planned to have ways to access these empathy levels). For testing purposes, ways to control these empathy levels were implemented.

After obtaining these three requirements, the gameManager cedes control to the ScenarioManager, which loads the quest line determined at the creation of the project.

The quest sequence is planned in the following way:

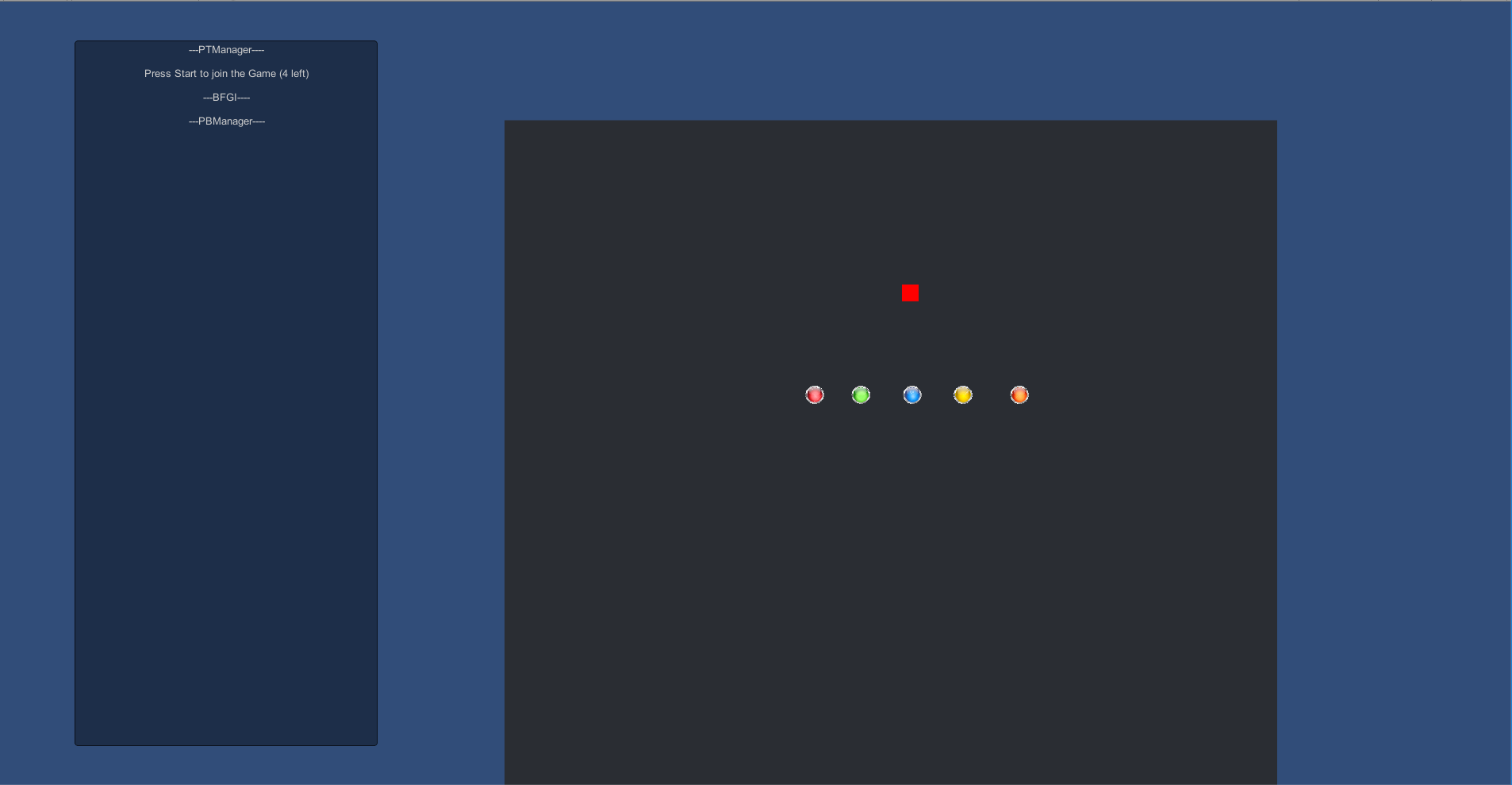
1 - Starting Quest - The first quest, meant to establish the scenario of the game (lore)

2 - Empathy Check - Will serve to access the players empathy levels (not yet implemented)

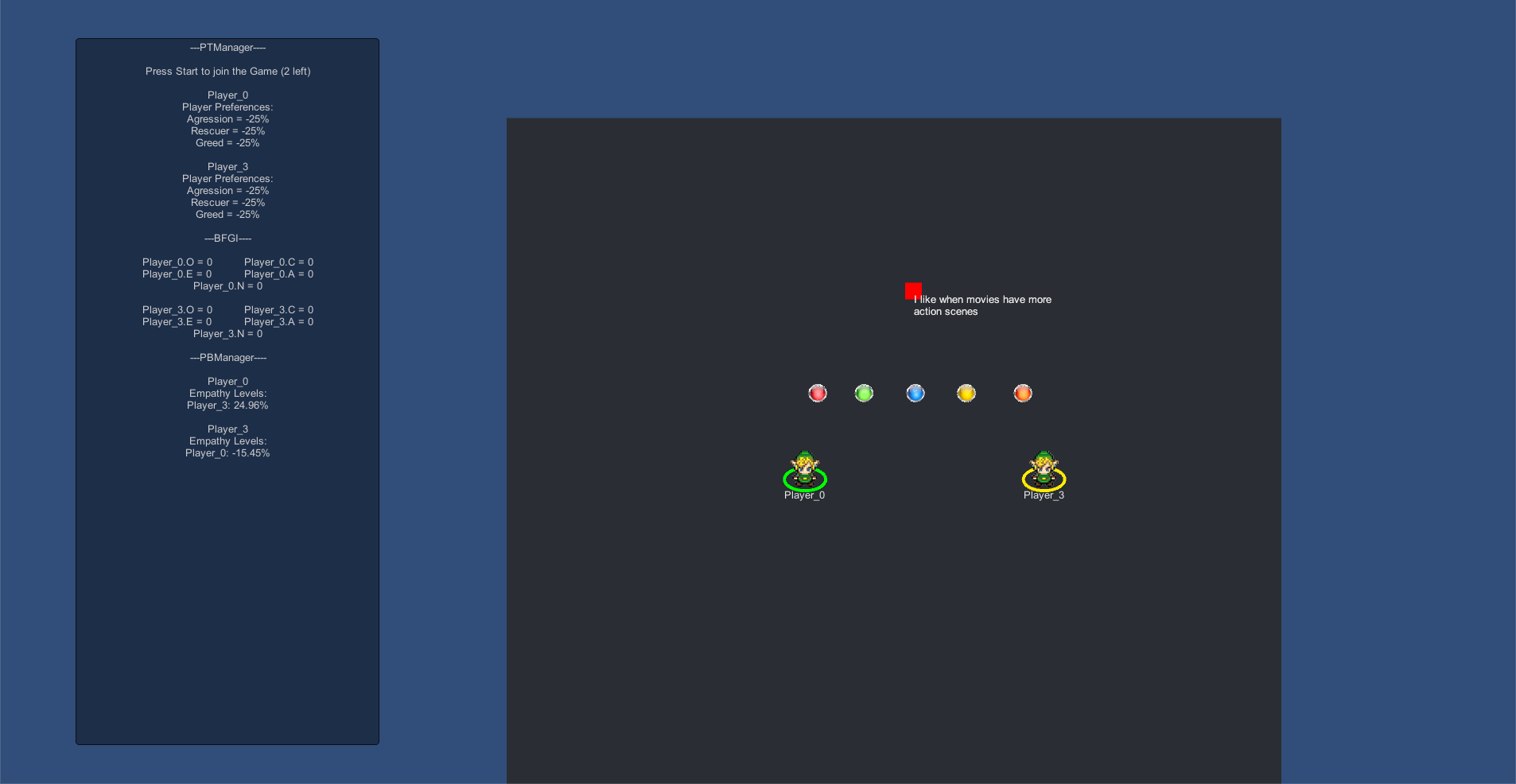
3 - CheckPoint - Quest that is called after each completed quest. Equivalent to a travel to the central hub, or main city of most games, where the players can rest, buy items etc.

[4-31] - Main Quests - Quests with varying degrees of intensity for Aggression, Rescuing and Gathering. The quest that best meets the players expectations is chosen. Currently, the player with the highest overall empathy with every other player is considered the leader of the group, and the code tries to please the leader, assuming the other players will be satisfied as well.

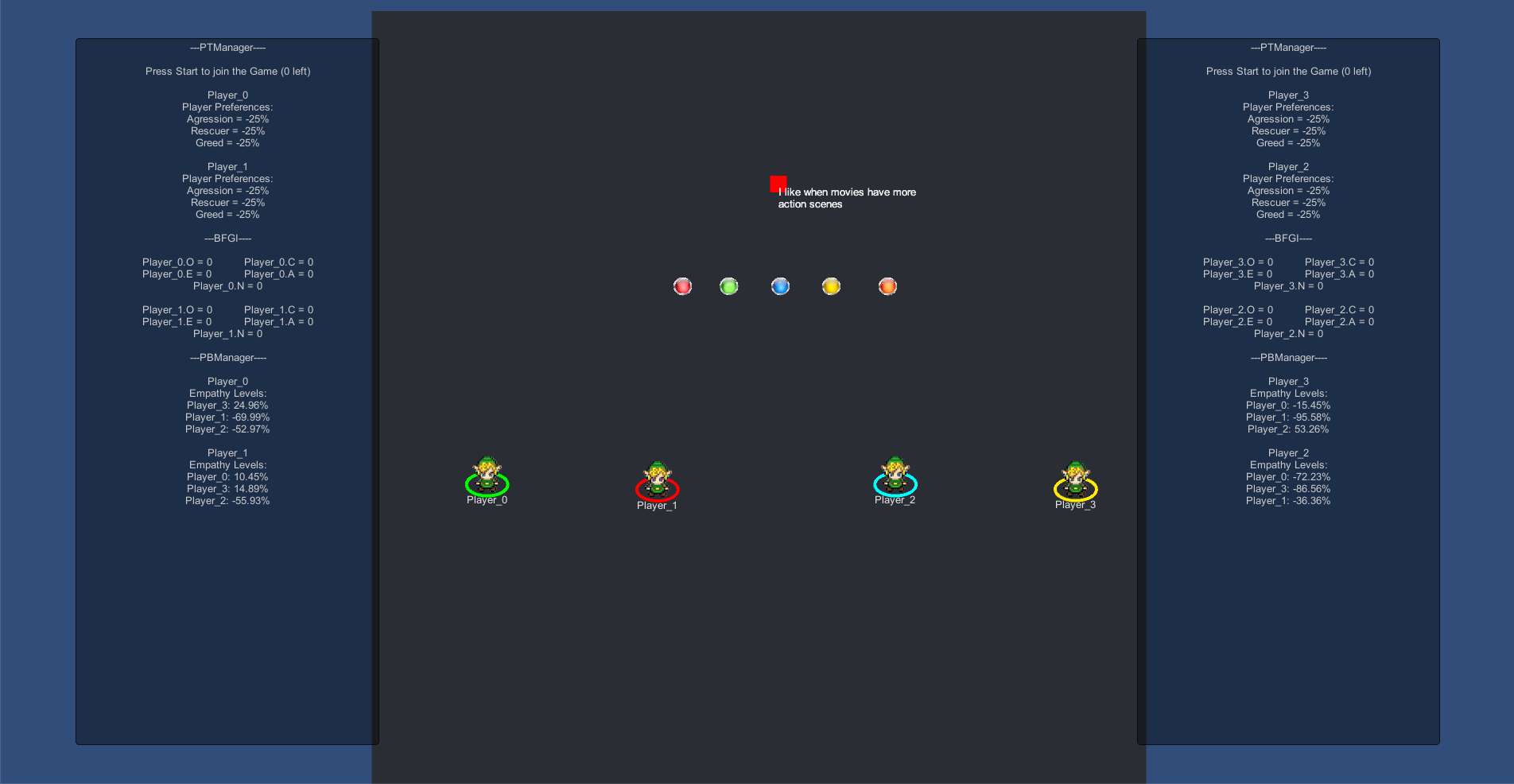
99 - EndGame - Final quest of the game, can only be reached once the players have reached the checkpoint at least 10 times.



Press Start Screen



Two Players UI Screen



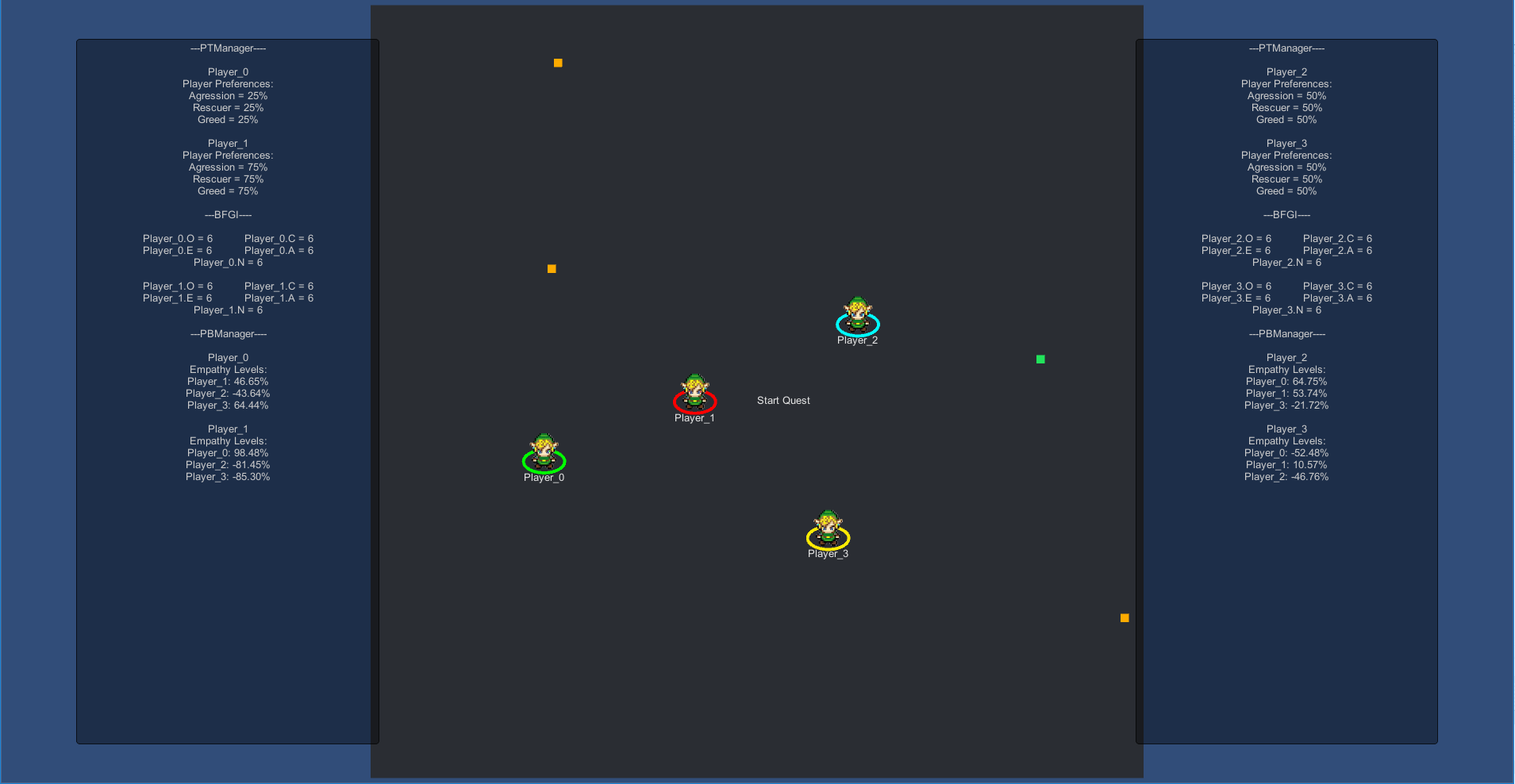
3+ Players UI Screen



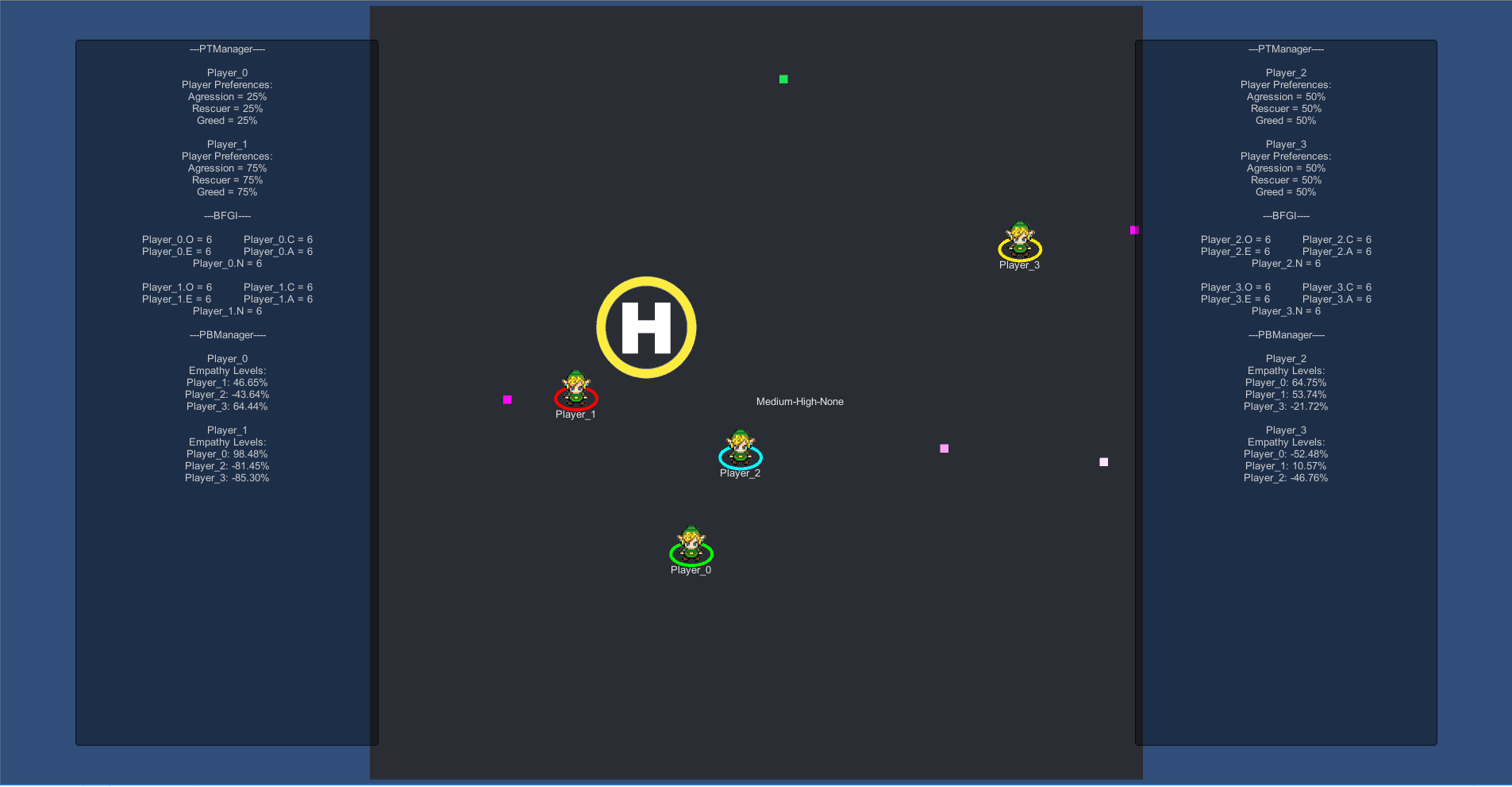
Answering Player Type Questionnaire



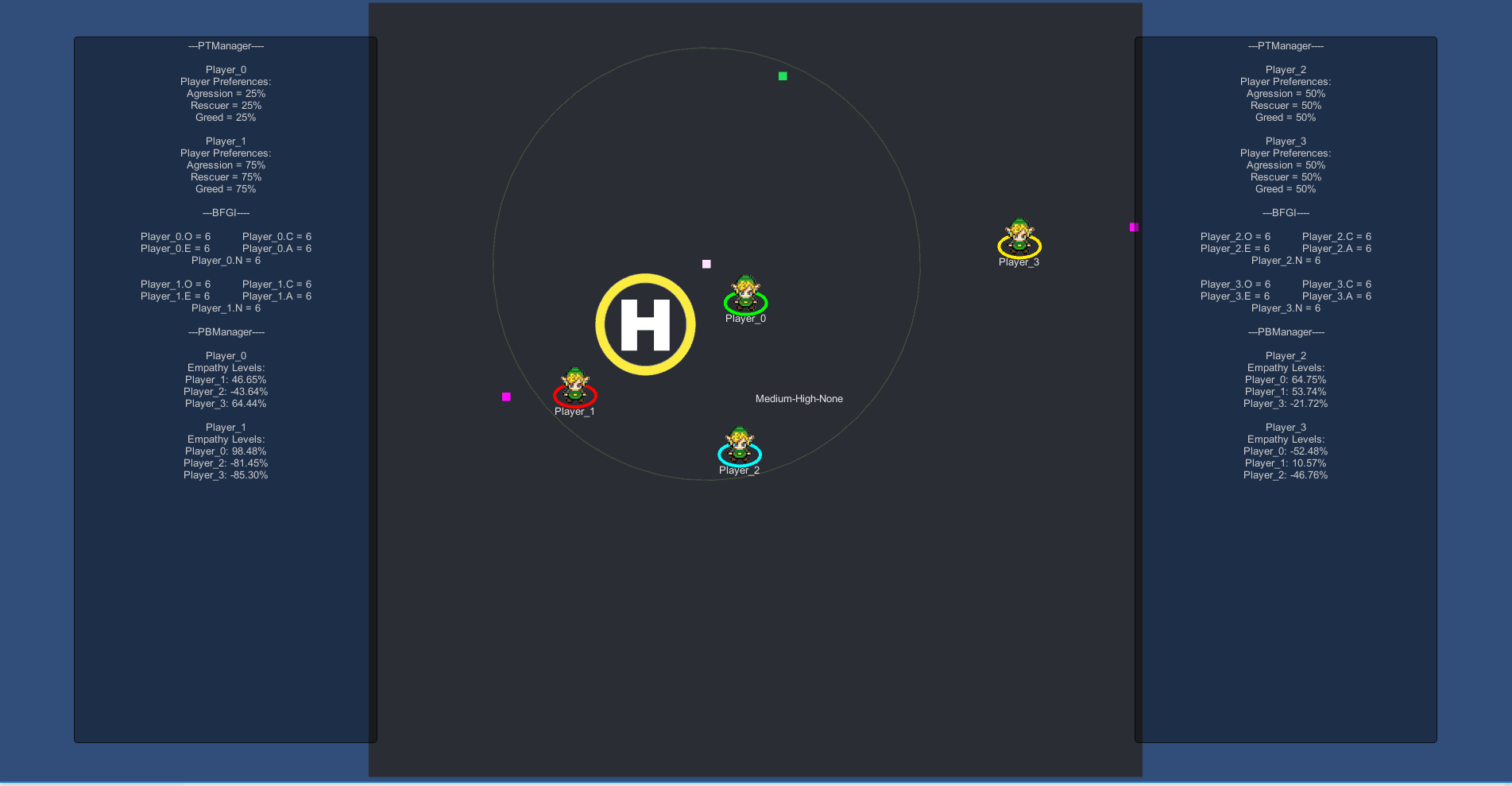
Answering BFGI Questionnaire



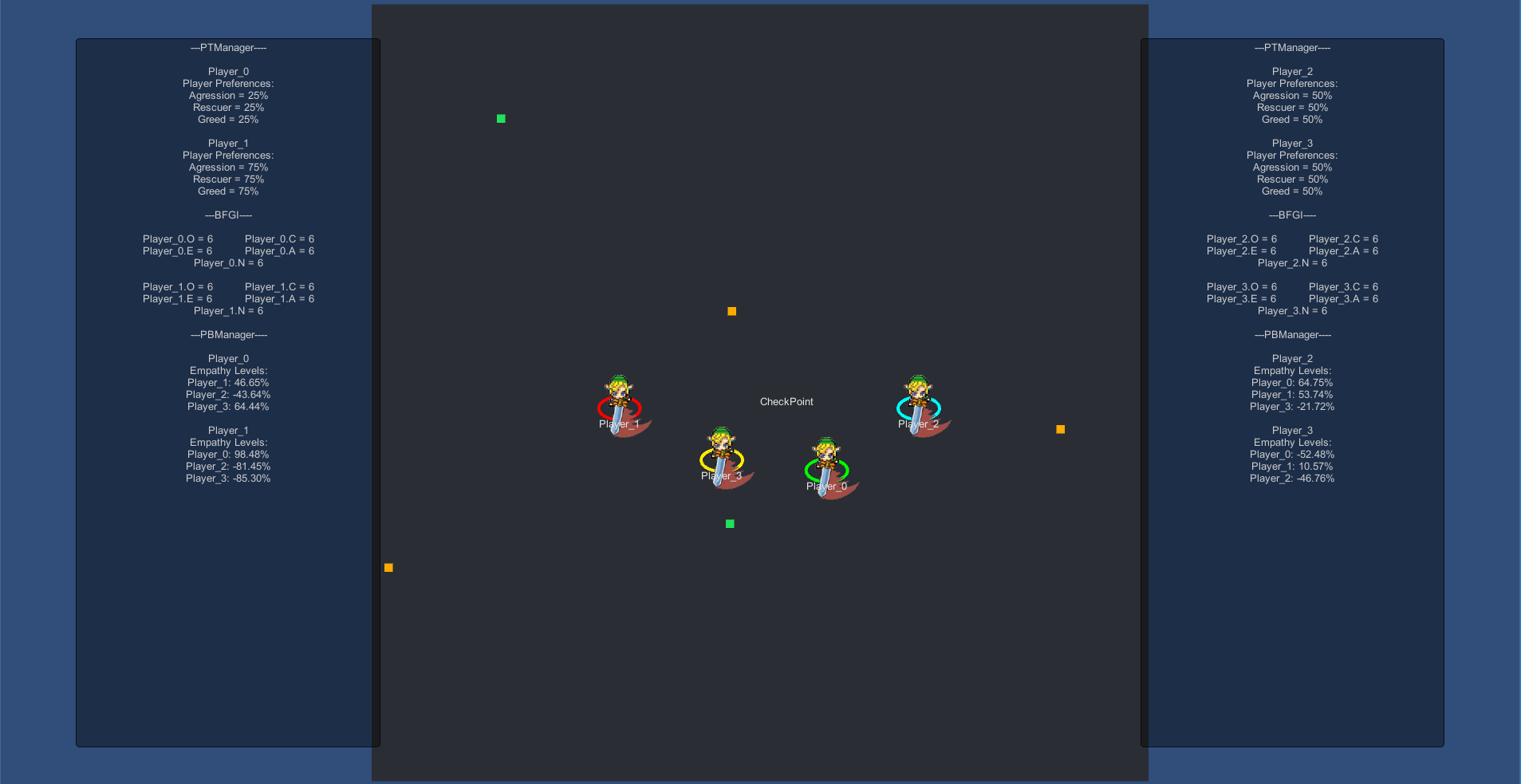
Start Quest



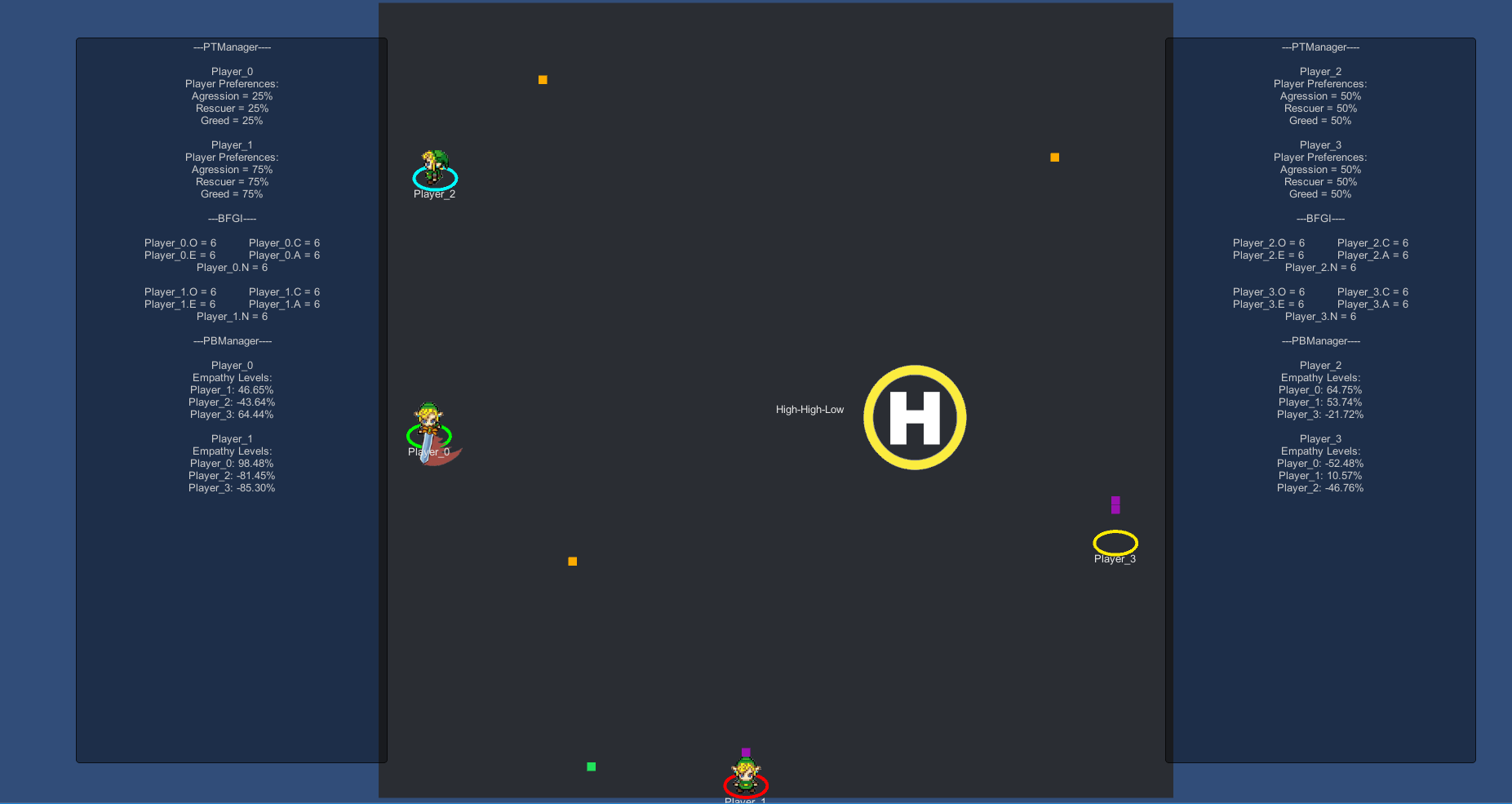
Quest Example - Medium Aggression, High Rescuing, None Collector



Rescuing a Victim



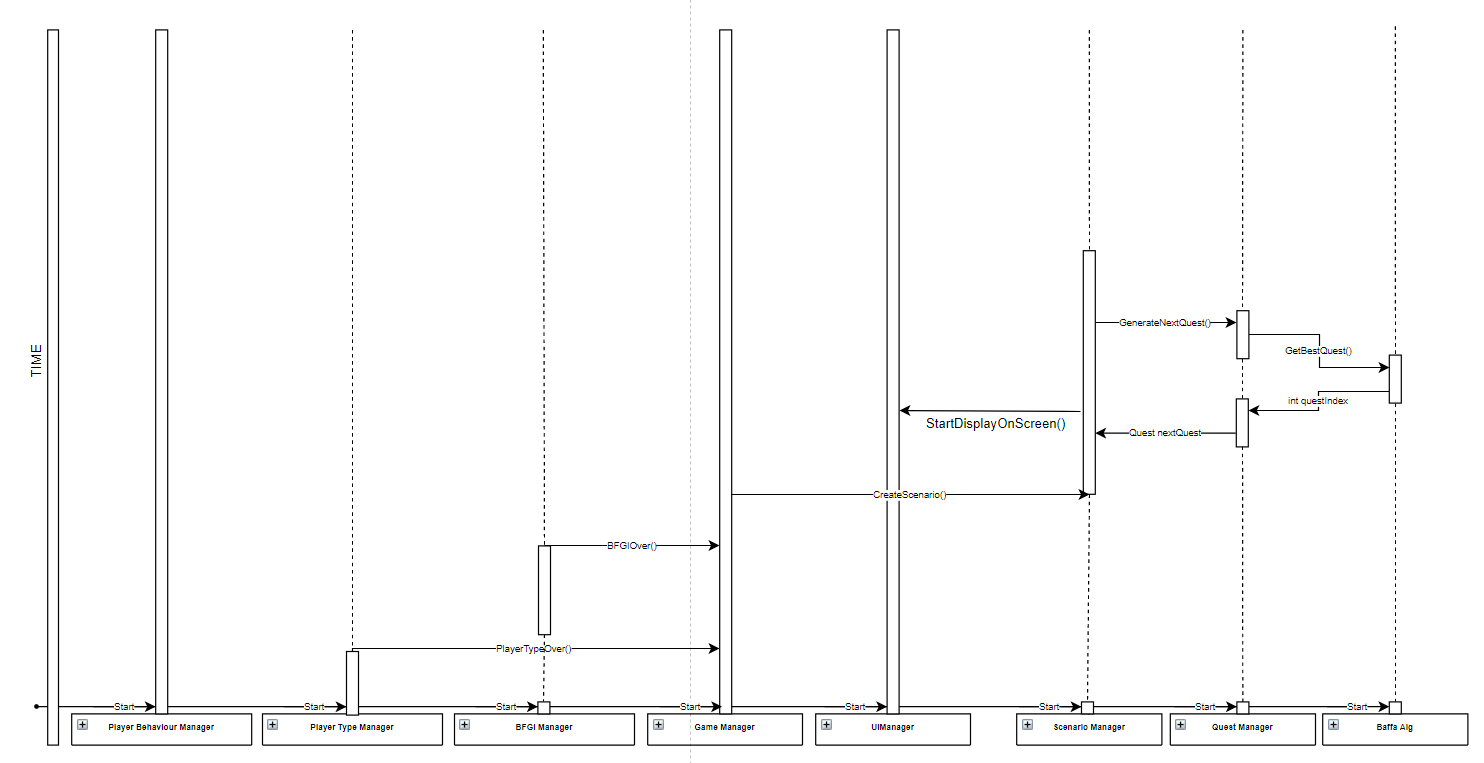
Checkpoint



Player 3 Damaged Player 1 Attacking



EndGame

Figure 4 - Sequence Diagram

## Coding

The coding of the project was made as modular as possible, which causes some redundancies when calling methods from classes, but keeps the documentation and functionality concise (high cohesion and low coupling). All of the game Managers (GameManager, UIManager, ScenarioManager, BFGIManager, PlayerTypeManager, BaffaAlg, QuestManager) were made using the Singleton design pattern, that is, static classes with static methods for calling and accessing them. That eliminates the need to declare a local object on start and allowing to only access the class on demand.

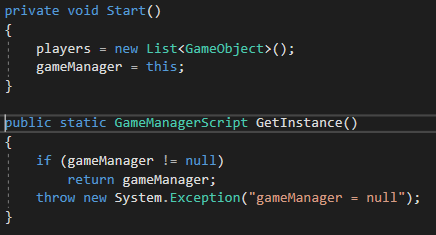


Figure 5 - Static Method Example

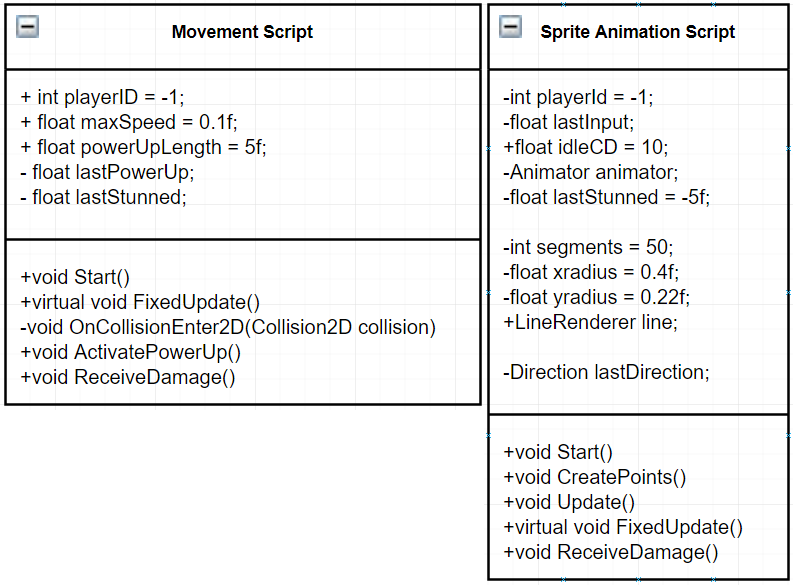
The project has very low coupling meaning changes in a module shouldn’t have much impact in the workings of the other modules, if the necessary precautions are made. Measures were also taken to make sure the modules would be as cohesive as possible, meaning the elements in each module try to be as particular to that module as possible. New elements can be added to the modules but should be kept so as to not increase coupling between modules.

The project was developed using cloud storage to increase backup safety and remote development. No version control was implemented, and version and development software (ie. github) was recommended but not utilized.

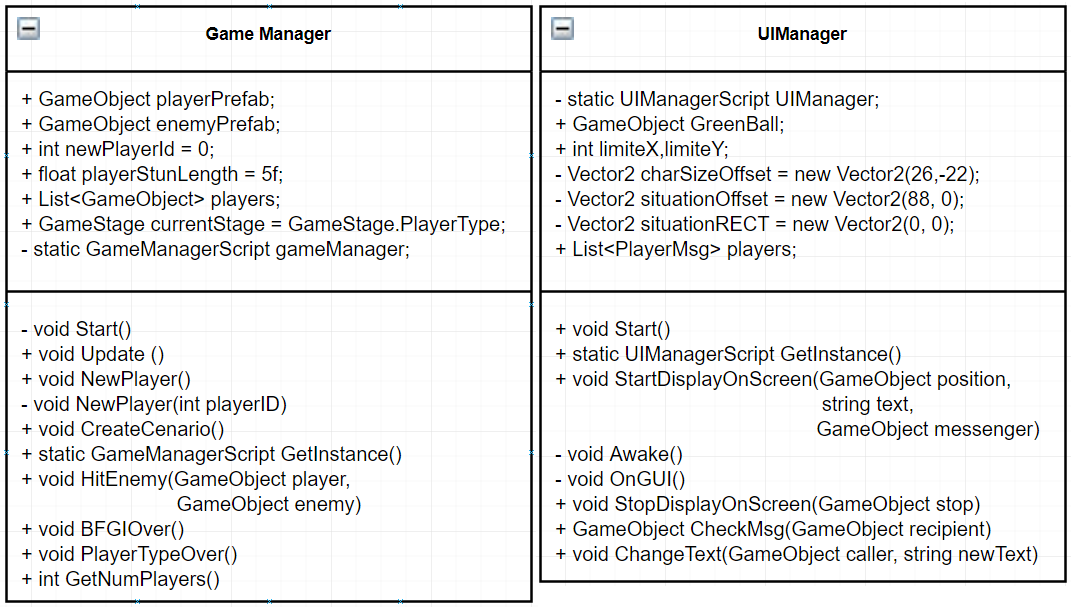
Some documentation was published on each module, but only for the methods relevant to the implementation of the quest selection system and the BFGI methods. Documentation about the game itself (ie. enemies, adding new players, generation of scenarios etc) was not provided.

The classes contained in the project are listed in the Figures below with each attribute and method. They expand on the main modules presented in Figure 2 - System’s main modules.

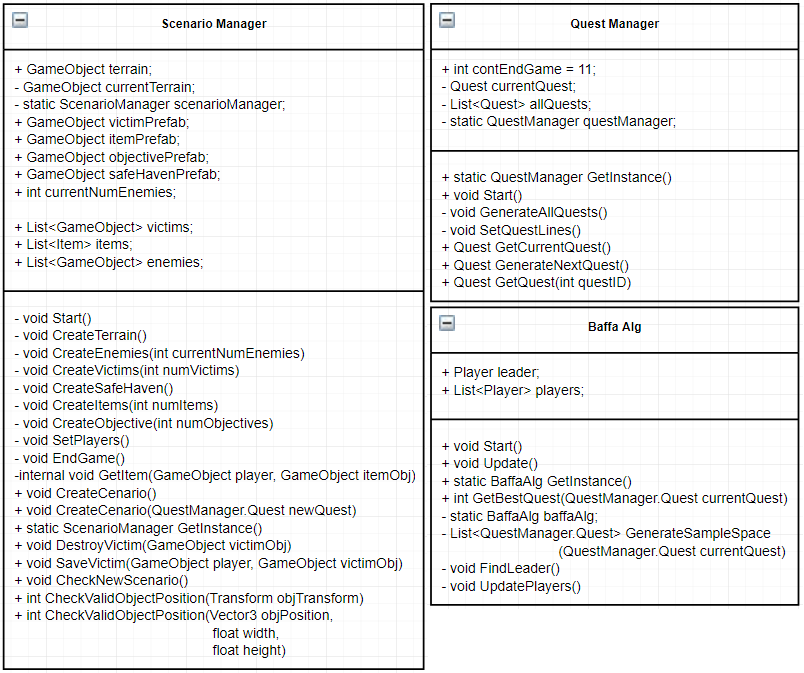
The entire unity project and code solution can be found at [[https://bit.ly/2X34WFJ](https://www.dropbox.com/sh/k0t8etvguroaaxk/AACaqu5S8NvezN6Uepux8a-4a?dl=0)].



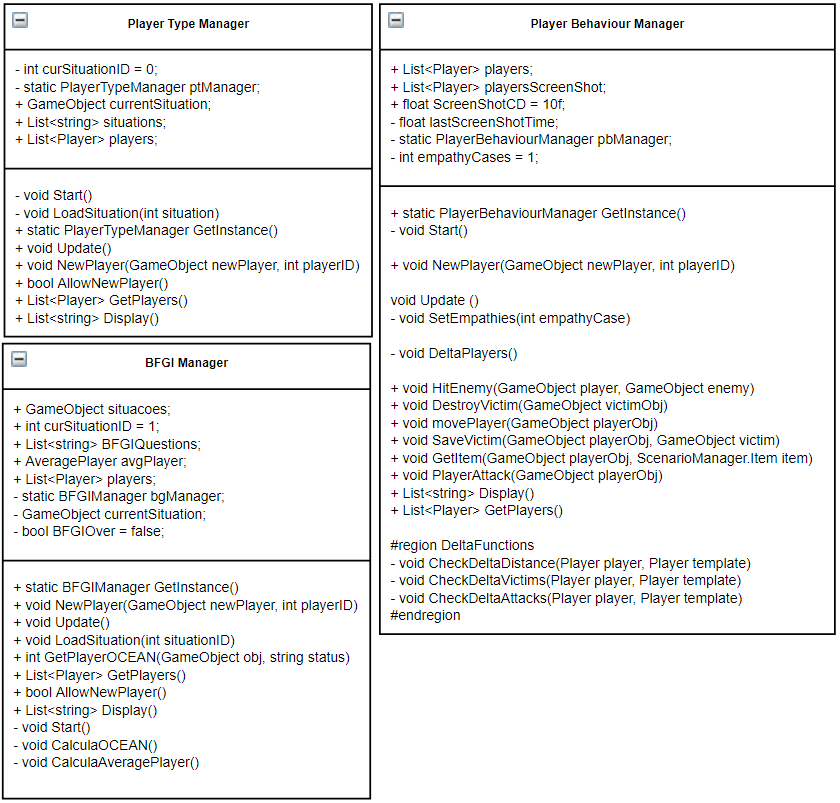
Player Scripts



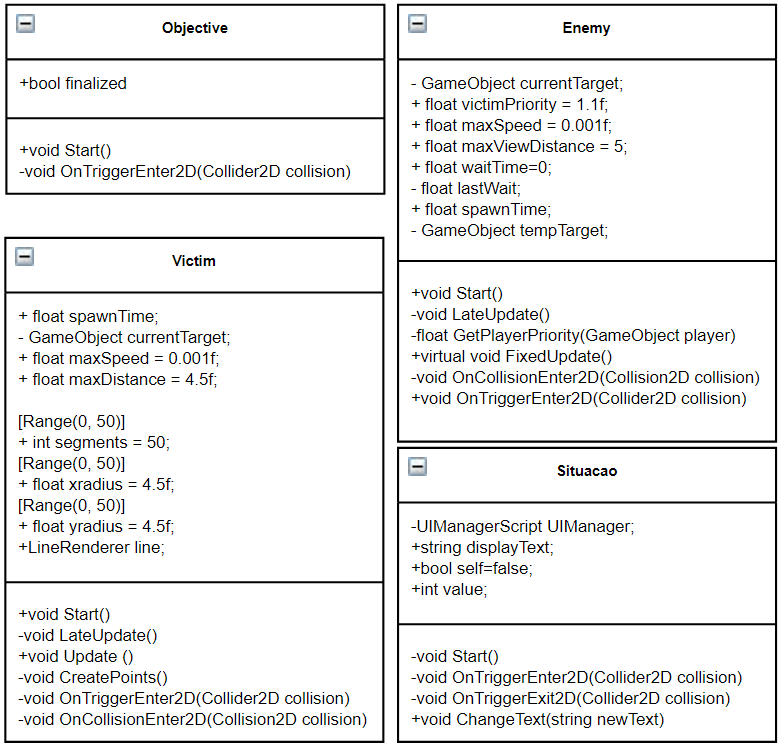
Game Managers



Quest Managers



Player Characteristics Managers



Misc Classes

## Testing and Maintenance

**6.1 Testing**

Testing was made in-house during development. No focus group was used to test the validity of the BFGI model, but extensive test was made on the functioning of the quest selection system, the analysis for the player type and OCEAN values.

More time should be spent in asserting the BFGI model, but the project can be expanded on future works. The project provides a good starting point for future implementation

After each questionnaire is complete, the corresponding result for each player is displayed on screen (Figure 6). On each side of the screen, it is possible to check the output from the PlayerType Manager for each player (Player\_0 [on the left side] to Player\_3 [on the right side]) detailing the preference to each type of game. The same is true for the BFGI Manager that details the magnitude of each OCEAN value from each player, and lastly the PlayerBehaviour Manager details the Empathy Levels that each player feels towards each other player.

This, plus the random assigned values for empathy, allowed us to test the capturing of the values and fulfil requirements [REQ001] through [REQ003].

All the small tests for the basic functionality of the system (player movement, attack input, enemy actions, level generation, etc) were tested on implementation,

The only other test necessary for the correct functioning of the system is the quest selection algorithm. For that, we implemented a function that allowed us to randomize the empathy values of each player on demand, so that we can create a leader with OCEAN O value of <4. With that setup, it is possible to test the return of the quest selection system and guarantee that only the quests that fit the leader’s player type gets returned on the quest sample space.



Figure 6 - Questionnaire Results

**6.2 Maintenance**

Maintaining the project should come in the form of adapting it to the local machine, making sure it runs properly, but implementation shouldn’t vary between machines. Once built and deployed, the project should run in any windows machine with minimal specs.

## Conclusion and Future Projects

**7.1 Conclusion**

In this project, the goal was to develop a Role Playing Game that adapted itself to the players preferences, in an attempt to shift away from the branching system of old video games, and the new open world concept of the newer games. The method chosen was to study the player preferences and relationships to find out what the players liked and disliked as a group.

That goal was only partially reached, and the end product is a system that studies the current player base, elects a leader among the players (the influencer) and, if the leader is not open to new experiences (OCEAN value O < 40%), the system filters the pool of available quests to only include the kinds of quests that that leader likes.

The system is simple in its development so much work can still be made upon to improve its market availability, however its theoretical background seems sound. Testing was made in-house and the the system implements what was proposed, however further testing should be done with a live audience to see its appeal and acceptance.

**7.2 Future Projects**

The system is still in its infancy. The system still used some assumptions, and only implements part of its functionalities.

Further development should be made to study player relations. The current project assigns random values for the empathy levels between each pair of players. A system that analyses player interactions with one another and the game environment would increase the system’s reliability to create a more pleasant experience for the entire group, and not assume that pleasing the leader would assume the whole group. No treatment was made, for example, about what to do if one person of the group had negative empathy levels with the elected leader. That would mean pleasing the leader could displease the player. Deliberations like those should be made and treated.

Other future developments also include expanding the treatment of the criteria used to choose quests. Currently the system only enter in effect if the elected leader OCEAN’s Openness value is < 40%, that is, the leader doesn’t deviate from their preferred game types. If the leader Openness value is higher than 40%, all quests become available for the whole group. New criteria should be studied and implemented to make the custom quest choosing system more dynamic and cover a wider array of player groups.

Other decision systems could also be implemented that doesn’t involve electing a leader, or consider smaller subsets of those players (ie. two good friends and 3 not-as-good friends).

Given the system low cohesion, new forms of quest selection could be implemented with minimal work, requiring only the change of a pair of methods in the baffaAlg called GetBestQuest() and GenerateSampleSpace().

Documentation should also be worked upon, to better explain the functionalities of each module and the project as a whole. Versioning should be expanded upon, by utilizing third party softwares and techniques for version control.

## References

[1] Lima ES, Feijó B, Furtado AL. Player behavior and personality modeling for interactive storytelling in games. *Entertainment Computing*. 2018; 28: 32-48. Available from doi:10.1016/j.entcom.2018.08.003.

[2] Baffa A, Poggi M, Feijó B. Adaptive Automated Storytelling Based on Audience Response. In: Chorianopoulos K, Divitini M, Hauge JB, Jaccheri L., Malaka R. (eds.). *Lecture Notes in Computer Science.* 1ed. Cham: Springer International Publishing. 2015; 9353: 45-58. Available from doi: https://doi.org/10.1007/978-3-319-24589-8\_4.

[3] Goldberg LR. An alternative “description of personality”: the Big-Five factor

structure. *J. Pers. Soc. Psychol.* 1990; 59 (6): 1216–1229, 1990.

[4] Wikipedia. The Big Five Personality Traits. Available from: <https://en.wikipedia.org/wiki/Big_Five_personality_traits>. [Accessed 19th May 2019]

1 - [Big Five Game Questionnaire](http://www.icad.puc-rio.br/~logtell/interactive-quests/bfgi-10.pdf)

2 - [Big Five Questionnaire](https://www.gesis.org/fileadmin/upload/SDMwiki/BFI-10/BFI-10_English_Items.pdf)

3 - [Big Five Personality Traits (Wikipedia)](https://en.wikipedia.org/wiki/Big_Five_personality_traits)

4 - [Video Demonstration](https://bit.ly/2ZYVDsx)