|  |  |
| --- | --- |
|  |  |
| FINAL PROJECT PLANT VS ZOMBIE GAME |  |
|  |  |
|  | 2/1/2020OBJECT ORIENTED PROGRAMMING |
|  | DR. TRAN THANH TUNGGROUP’S NAME |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  | ABSTRACTLorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. | |  |
|  | ACKNOWLEDGEMENT Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.  Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.  Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. | Image |  |

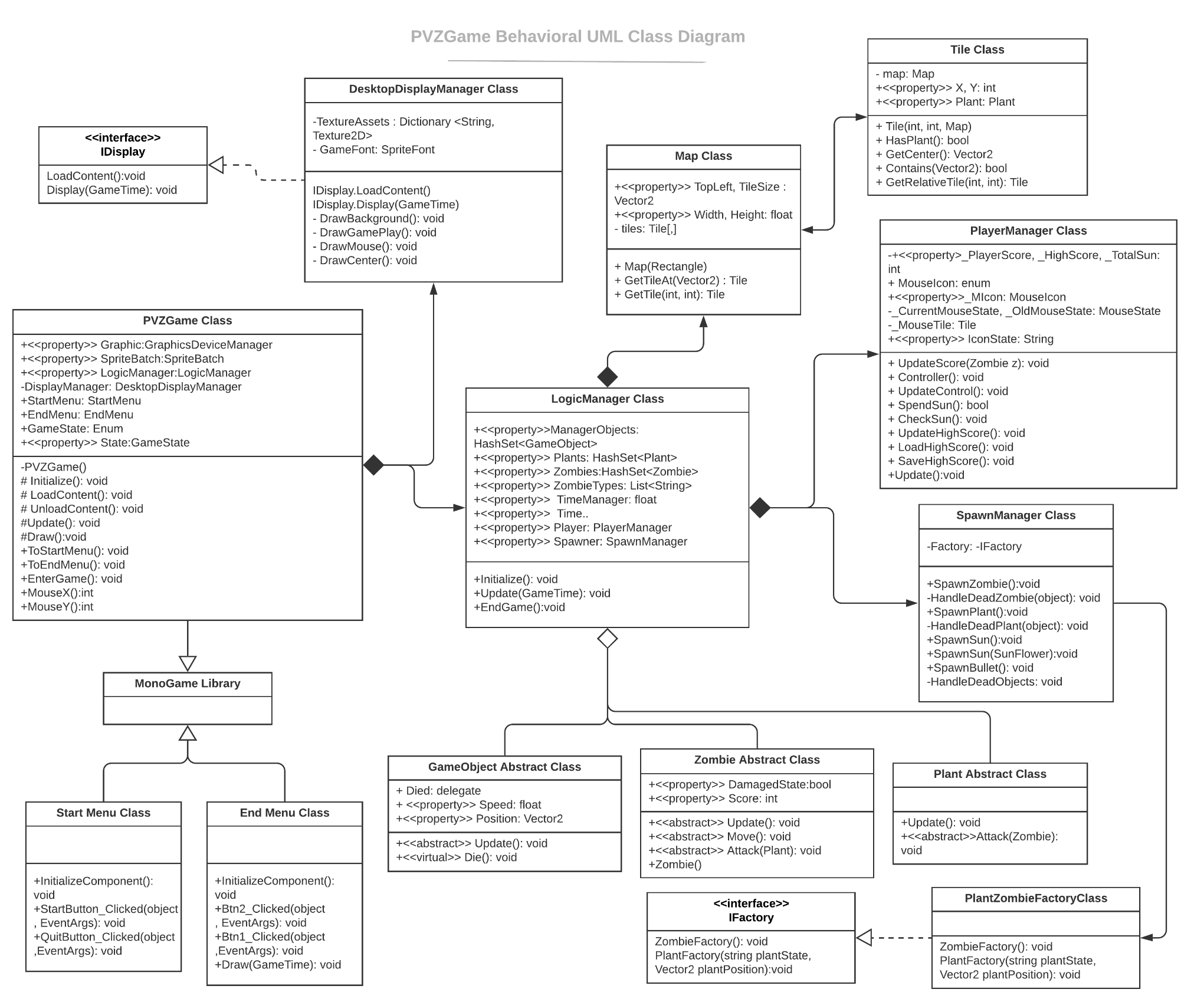
|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  | GAME CONTENT – NEEDS EDITING |  |  |
|  | HEADING Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. | |  |

### ARCHITECTURE AND DESIGN

#### In this section, the architecture and design of the game is discussed and reasoned to show the compromise of the team. This section is divided into three parts: behavioral architecture, structural architecture and design patterns used in the game.

## **Behavioral architecture:**

The main efforts while trying to build this behavioral side of architecture was to avoid passing arguments and making bi-directional association between two classes. Such efforts bring our team to the architecture as in the UML diagram below:



PVZGame Class extends from the class Game of MonoGame framework. Since this class must obey the structure required by the framework, it contains both game logical functions such as Initialize function, Load and Unload Content functions, Update function and draw function. As the team trying to compromise with Single Responsibility Principle, we are unable to separate these functions since they are protected. Hence, the team came to the solution of creating two Managers: Logic and Display. These Managers are the components of the PVZGame class. Then, PVZGame will only control the properties that are related to the game state and MonoGame frameworks.

* DisplayManager, or DesktopDisplayManager implements the IDisplay interface and manage the drawing methods of the game. This implementation not only obeys the Single Responsibility Principle, but also provides an openness for extension in other displayers (webs, phones, etc.).   
  Furthermore, reflection by using TextureAssets Dictionary from a String value to a Texture2D value and GetType methods is applied in order to draw the textures of the object without having to create concrete Draw classes or violating Open/Close Principle.
* LogicManager control the game loop and be the composite of most of the game objects. Other objects try to “know” each other by this Manager.

It is worthwhile to mention the SpawnManger class. This class uses a concrete class which implements from the IFactory interface. This step follows the Factory Method design patterns, this pattern will be discussed in detail in the Design Patterns used in the game.   
The PlayerManager class shown in the diagram controls all features related to the players, namely: score, sun (to plant plants), mouse and keyboard controls. This class although has to control many aspects of the player, since these aspects are unlikely to extend, it is compromised to put all of these aspects in to one class for the reduction of complexity.

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Lorem Ipsum is simply dummy text of the printing and typesetting industry. |
|  |