Object Manager

This module serves as the main object factory as well as the module to update and access any objects it has currently. This module will take care of all creation of all objects during its initialization based off of the max number of each type of object, recycling them when a new object is needed. In order to get access to any object that the programmer does not have a pointer to, he must search for it through a vector based on type of object. This will also check for and handle all collisions for all objects.

**Dependencies**

* Access to the following:
  + Collision library
  + Base Object
  + Player Object
  + Projectile Object
  + Enemy Object
  + Fire Object
  + Ice Object
  + Ranged Object
  + Melee Object
  + Explosive Object
  + Wizard Object
* Accessed by the following:
  + State System
  + Enemy Behavior
  + Fire Beast Behavior
  + Ice Beast Behavior
  + Ranged Enemy Behavior
  + Melee Enemy Behavior
  + Explosive Enemy Behavior
  + Wizard Enemy Behavior

**Public Interface**

|  |  |  |  |
| --- | --- | --- | --- |
| **Return** | **Name** | **Parameters** | **Description** |
| Bool | Initialize | Int numMelee  Int numRanged  Int numExposive  Int numIce  Int numFire | Initializes the object factory and all of the objects based on the max amounts of each type of object passed in. |
| Player\* | GetPlayer | void | Returns a pointer to the character to be used for reading or writing to. |
| BaseObject\* | GetObjects | Int objectType  //based off of the m\_eObjectTypes enum | Returns a pointer to the entire array of the specified type of objects. |
| BaseObject\* | NewObject | Int objectType  //based off of the m\_eObjectTypes enum | Returns a pointer to a reinitialized object of the specified type. |
| Void | Update | Float fElapsedTime  // time elapsed since last update | Updates all objects and calls its private function to check all collisions of all active objects in the game world |
| Void | ExpandObjectList | Int objectType  //based off of the m\_eObjectTypes enum  Int newNum | Resizes the size of an array of a specified type to the new size. If less than the original amount, will return out without resizing. |
| Void | Shutdown |  | Deletes all objects and shuts down the object manager. |

**Time to Complete Estimate**

* Object Manager estimate – 1 day
  + Once the object hierarchy is done with all of their classes, this should mostly be just calling their function for approximately half a day and integrating the collision library into the behind the scenes functions to check all collisions.

**Module Author(s)**

* Charles Meade

Base Object

This module is the base class for all of the objects in the game world. It will be an abstract base class so that there can be no instances of this in the game and only the derived types. It will have the basic functions and variables that all objects in the world will need.

**Dependencies**

* Access to the following:
  + View
  + Base Behavior
  + Behavior Manager
* Accessed by the following:
  + Base Behavior
  + Behavior Manager
  + Game State System
  + View
  + Object Manager

**Public Interface**

|  |  |  |  |
| --- | --- | --- | --- |
| **Return** | **Name** | **Parameters** | **Description** |
| Void | Initialize | Int Type  // based off of the of the m\_eObjectTypes enum | Virtual function that needs to be overridden. The overridden function sets the initial starting variables. |
| Void | Reinitialize |  | Virtual function that needs to be overwritten. The overridden function resets the variables to their starting values. Used for object recycling. |
| Void | Update | Float fElapsedTime | Updates all of the variables based off of its update logic. |
| Int | GetType |  | Returns the type of object this instance is. |

**Time to Complete Estimate**

* Base Object– less than one day

**Module Author(s)**

* Charles Meade

Player Object

This module stores all of the data and functionality for the character that the player controls throughout the game. This derives off of base object and overrides all of the virtual functions of the base. There should only ever be one instance of this class in the game at one time.

**Dependencies**

* Access to the following:
  + Object Manager
  + State System
  + Base Object
  + View
  + Base Behavior
  + Behavior Manager
* Accessed by the following:
  + Object Manager
  + Base Object
  + State System
  + Behavior system
  + Base Behavior
  + View

**Public Interface**

|  |  |  |  |
| --- | --- | --- | --- |
| **Return** | **Name** | **Parameters** | **Description** |
| Void | Initialize | Int Type  // based off of the of the m\_eObjectTypes enum | Function that sets the initial starting variables. |
| Void | Reinitialize |  | Function that resets the variables to their starting values. Used for object recycling. |
| Void | Update | Float fElapsedTime | Updates all of the variables based off of its update logic. |
| Int | GetType |  | Returns the type of object this instance is. |
| Void | UpdateInput | Char inputBits | Takes in a char full of bits to show if each input is currently down and stores them for update. This should be called before calling this object’s update. |

**Time to Complete Estimate**

* + Player object – less than one day

**Module Author(s)**

* Charles Meade

Environment Object

This module derives off of base object and has all base data and functionality that each environment object must have.

**Dependencies**

* Access to the following:
  + View
* Accessed by the following:
  + Object Manager
  + Base Object
  + State System
  + Behavior system
  + Base Behavior
  + View

**Public Interface**

|  |  |  |  |
| --- | --- | --- | --- |
| **Return** | **Name** | **Parameters** | **Description** |
| Void | Initialize | Int Type  // based off of the of the m\_eObjectTypes enum | Function that sets the initial starting variables. |
| Void | Reinitialize |  | Function that resets the variables to their starting values. Used for object recycling. |
| Void | Update | Float fElapsedTime | Updates all of the variables based off of its update logic. This class will not have very much inside it if anything for updating. |
| Int | GetType |  | Returns the type of object this instance is. |

**Time to Complete Estimate**

* + Environment Object – less than one day

**Module Author(s)**

* Charles Meade

Enemy Object

This module derives off of base object and has all base data and functionality that each enemy object must have.

**Dependencies**

* Access to the following:
  + Object Manager
  + State System
  + Base Object
  + View
  + Base Behavior
  + Behavior Manager
  + Player Object
* Accessed by the following:
  + Object Manager
  + Base Object
  + State System
  + Behavior system
  + Base Behavior
  + View

**Public Interface**

|  |  |  |  |
| --- | --- | --- | --- |
| **Return** | **Name** | **Parameters** | **Description** |
| Void | Initialize | Int Type  // based off of the of the m\_eObjectTypes enum | Function that sets the initial starting variables. |
| Void | Reinitialize |  | Function that resets the variables to their starting values. Used for object recycling. |
| Void | Update | Float fElapsedTime | Updates all of the variables based off of its update logic. |
| Int | GetType |  | Returns the type of object this instance is. |

**Time to Complete Estimate**

* + Enemy Object- less than one day

**Module Author(s)**

* Charles Meade

Projectile Object

This module derives off of base object and has all base data and functionality that each projectiles must have

**Dependencies**

* Access to the following:
  + Object Manager
  + State System
  + Base Object
  + View
  + Base Behavior
  + Behavior Manager
* Accessed by the following:
  + Object Manager
  + Base Object
  + State System
  + Behavior system
  + Base Behavior
  + View

**Public Interface**

|  |  |  |  |
| --- | --- | --- | --- |
| **Return** | **Name** | **Parameters** | **Description** |
| Void | Initialize | Int Type  // based off of the of the m\_eObjectTypes enum | Function that sets the initial starting variables. |
| Void | Reinitialize |  | Function that resets the variables to their starting values. Used for object recycling. |
| Void | Update | Float fElapsedTime | Updates all of the variables based off of its update logic. |
| Int | GetType |  | Returns the type of object this instance is. |

**Time to Complete Estimate**

* + Projectile Object

**Module Author(s)**

* Charles Meade

Melee Object

This module derives off of enemy and has all base data and functionality that each melee minion must have.

**Dependencies**

* Access to the following:
  + Object Manager
  + State System
  + Base Object
  + View
  + Base Behavior
  + Behavior Manager
* Accessed by the following:
  + Object Manager
  + Base Object
  + State System
  + Behavior system
  + Base Behavior
  + View

**Public Interface**

|  |  |  |  |
| --- | --- | --- | --- |
| **Return** | **Name** | **Parameters** | **Description** |
| Void | Initialize | Int Type  // based off of the of the m\_eObjectTypes enum | Function that sets the initial starting variables. |
| Void | Reinitialize |  | Function that resets the variables to their starting values. Used for object recycling. |
| Void | Update | Float fElapsedTime | Updates all of the variables based off of its update logic. |
| Int | GetType |  | Returns the type of object this instance is. |

**Time to Complete Estimate**

* + Melee Object- less than one day

**Module Author(s)**

* Charles Meade

Explosive Object

This module derives off of enemy and has all base data and functionality that each explosive minion must have.

**Dependencies**

* Access to the following:
  + Object Manager
  + State System
  + Base Object
  + View
  + Base Behavior
  + Behavior Manager
* Accessed by the following:
  + Object Manager
  + Base Object
  + State System
  + Behavior system
  + Base Behavior
  + View

**Public Interface**

|  |  |  |  |
| --- | --- | --- | --- |
| **Return** | **Name** | **Parameters** | **Description** |
| Void | Initialize | Int Type  // based off of the of the m\_eObjectTypes enum | Function that sets the initial starting variables. |
| Void | Reinitialize |  | Function that resets the variables to their starting values. Used for object recycling. |
| Void | Update | Float fElapsedTime | Updates all of the variables based off of its update logic. |
| Int | GetType |  | Returns the type of object this instance is. |

**Time to Complete Estimate**

* + Explosive Object- less than one day

**Module Author(s)**

* Charles Meade

Ranged Object

This module derives off of base object and has all base data and functionality that each ranged enemy must have.

**Dependencies**

* Access to the following:
  + Object Manager
  + State System
  + Base Object
  + View
  + Base Behavior
  + Behavior Manager
* Accessed by the following:
  + Object Manager
  + Base Object
  + State System
  + Behavior system
  + Base Behavior
  + View

**Public Interface**

|  |  |  |  |
| --- | --- | --- | --- |
| **Return** | **Name** | **Parameters** | **Description** |
| Void | Initialize | Int Type  // based off of the of the m\_eObjectTypes enum | Function that sets the initial starting variables. |
| Void | Reinitialize |  | Function that resets the variables to their starting values. Used for object recycling. |
| Void | Update | Float fElapsedTime | Updates all of the variables based off of its update logic. |
| Int | GetType |  | Returns the type of object this instance is. |

**Time to Complete Estimate**

* + Ranged Object - less than one day

**Module Author(s)**

* Charles Meade

Ice Object

This module derives off of enemy object and has all base data and functionality that each Ice Bease object must have.

**Dependencies**

* Access to the following:
  + Object Manager
  + State System
  + Base Object
  + View
  + Base Behavior
  + Behavior Manager
* Accessed by the following:
  + Object Manager
  + Base Object
  + State System
  + Behavior system
  + Base Behavior
  + View

**Public Interface**

|  |  |  |  |
| --- | --- | --- | --- |
| **Return** | **Name** | **Parameters** | **Description** |
| Void | Initialize | Int Type  // based off of the of the m\_eObjectTypes enum | Function that sets the initial starting variables. |
| Void | Reinitialize |  | Function that resets the variables to their starting values. Used for object recycling. |
| Void | Update | Float fElapsedTime | Updates all of the variables based off of its update logic. |
| Int | GetType |  | Returns the type of object this instance is. |

**Time to Complete Estimate**

* + Ice Object – one day

**Module Author(s)**

* Charles Meade

Fire Object

This module derives off of enemy object and has all base data and functionality that each Fire Beast object must have.

**Dependencies**

* Access to the following:
  + Object Manager
  + State System
  + Base Object
  + View
  + Base Behavior
  + Behavior Manager
* Accessed by the following:
  + Object Manager
  + Base Object
  + State System
  + Behavior system
  + Base Behavior
  + View

**Public Interface**

|  |  |  |  |
| --- | --- | --- | --- |
| **Return** | **Name** | **Parameters** | **Description** |
| Void | Initialize | Int Type  // based off of the of the m\_eObjectTypes enum | Function that sets the initial starting variables. |
| Void | Reinitialize |  | Function that resets the variables to their starting values. Used for object recycling. |
| Void | Update | Float fElapsedTime | Updates all of the variables based off of its update logic. |
| Int | GetType |  | Returns the type of object this instance is. |

**Time to Complete Estimate**

* + Fire Object- one day

**Module Author(s)**

* Charles Meade

Wizard Object

This module derives off of enemy object and has all base data and functionality that each wizard object must have.

**Dependencies**

* Access to the following:
  + Object Manager
  + State System
  + Base Object
  + View
  + Base Behavior
  + Behavior Manager
* Accessed by the following:
  + Object Manager
  + Base Object
  + State System
  + Behavior system
  + Base Behavior
  + View

**Public Interface**

|  |  |  |  |
| --- | --- | --- | --- |
| **Return** | **Name** | **Parameters** | **Description** |
| Void | Initialize | Int Type  // based off of the of the m\_eObjectTypes enum | Function that sets the initial starting variables. |
| Void | Reinitialize |  | Function that resets the variables to their starting values. Used for object recycling. |
| Void | Update | Float fElapsedTime | Updates all of the variables based off of its update logic. |
| Int | GetType |  | Returns the type of object this instance is. |

**Time to Complete Estimate**

* + Wizard Object- 2 days

**Module Author(s)**

* Charles Meade

Collision Library

This module stores all of the functions that must be used for all collision calculations.

**Dependencies**

* Access to the following:
  + Math Library
  + Base Object (friend)
* Accessed by the following:
  + Object Manager
  + State System

**Public Interface**

|  |  |  |  |
| --- | --- | --- | --- |
| **Return** | **Name** | **Parameters** | **Description** |
| Bool | CheckCollision | Int Type  // based off of the of the m\_eObjectTypes enum  Int CollisionType  // based off of an enum for aabb or spheres  BaseObject\* object1  BaseObject\*  object2 | This checks the collisions between two base objects and returns yes or no. |
| Void | GroundClamp | BaseObject\* object | Function that clamps an object to the terrain that it is currently on or close to. |

**Time to Complete Estimate**

* + Collision Library- one day

**Module Author(s)**

* Charles Meade