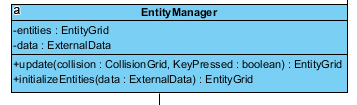
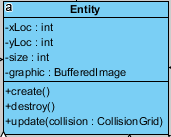
**EntityManager Class**



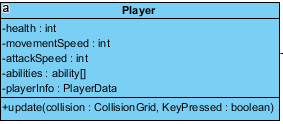
This class is created to manage all entity objects in the game, this class has two variables, data from the external data and “entities” from the entity grid, it has two functions which are “update” and “initializeEntities”, these functions return entity grid which will be needed for running other classes such as “CollisionManager” class

**Entity Class**

****

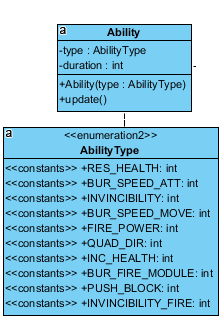
Entity class has main variables which will be used for all objects, “xLoc” and “yLoc” will store location of the object, “size” will have area object occupies, “graphic” will have the image to use specifying object. “create” function will be used for creating this specified object in the level, “destroy” will delete the specified object from the level, and update will send recent data to the “EntityManager” class.

**Player Class**

****

“Player” class will inherit the “Entity” Class and will be used to create a character user is going to use in the game. This entity will have health which will determine if game will be continued or not, “movementSpeed” will specify how fast Player will go into the field, “attackSpeed” will specify how frequently user will fire the projectiles, “abilities” will be used in order to boost the player’s stats temporarily or permanently, or give a new feature to the player. “update” method will send the recent data to the “Entity” class.

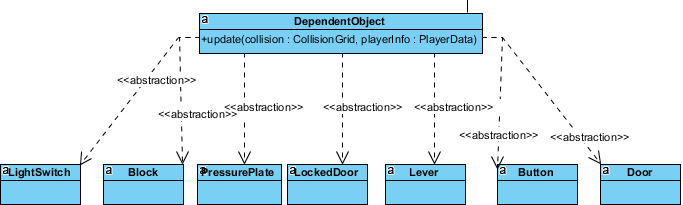
**Ability and AbilityType Classes**



“Ability” class will be used to change stats of player or give player a new ability. “type” variable will have specific ability which will be given to the player, “duration” variable will specify how many seconds this ability will last (10 seconds, permanently, etc.), “Ability(type)” is the constructor of the class, “update” method will send recent data to the “Player” class.

“AbilityType” class will store which abilities Player could get during the game. “RES\_HEALT” will restore the health of the player by 1, “BUR\_SPEED\_ATT” will increase movement speed by certain amount for certain duration or permanently, “INVINCIBILITY” will make player become immune to any source of damage for certain duration, “BUR\_SPEED\_MOVE” will increase attack speed by certain amount for certain duration or permanently, “FIRE\_POWER” will increase damage by certain amount for certain duration, “QUAD\_DIR” will allow player will fire towards all direction at once. “INC\_HEALTH” will increase maximum health by 1, “BUR\_FİRE\_MODULE” will allow player fire more than one projectile at once, “PUSH\_BLOCK” will allow player push certain blocks, “INVINCIBILITY\_FIRE” will protect player from the fire based damage

**DependendentObject and its abstract Classes**



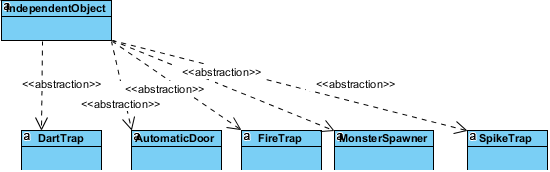
“DependentObject” class will contain dependent objects who will response according to the player data if the collision occures. “LightSwitch” abstract class will be used for turning off and on the light. “Block” abstract class will be used for block object which could be pushed if player gets related ability. “Button” abstract class will contain objects which will trigger some other objects. “PressurePlate” abstract class will have objects that have same principle with button but will be triggered by applying pressure on it. “Lever” abstract class will contain objects with same principle with button, yet levers could be triggered only once. “LockedDoor” abstract class will be used for passing next level, if player has a key, object will allow player to pass next level. “Door” abstract class will allow player to pass other parts of the current room, unlike the “LockedDoor” object, it will not require key.

**Projectile Class**



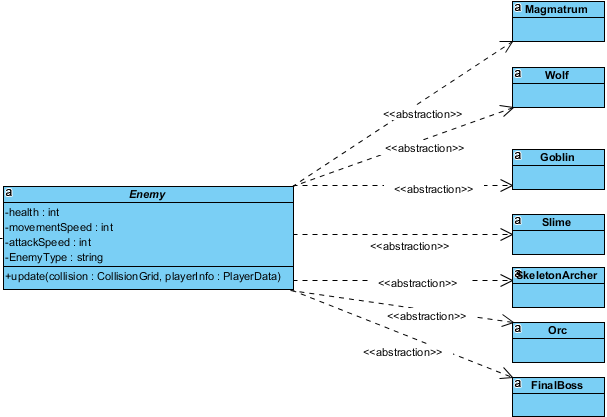
“Projectile” class will contain projectiles player and enemies fire, it will have “speed” variable which define how fast it will move from one pixel to another.

**IndependentObject and its Abstract Classes**



“IndependentObject” class will consist of object which will stay active by not depending on player data or occurrence of collision. “DartTrap” abstract class will create objects that shoot dart at the fixed direction, “AutomaticDoor” abstract class will create objects which will open and closed in a period, “SpikeTrap” abstract class will create objects with meele version of the dart trap, “FireTrap” abstract class will create objects with same principle with dart trap, but their range is between spike trap and dart trap, “Monster Spawner” abstract class will consist of objects which spawn mosnters in a period.

**Enemy and its Abstract classes**

****

“Enemy” class will be the main structure of the enemies which player will deal with to access next level. It has “health” variable which will determine if enemy is destroyed or not, “movementSpeed” will determine how fast enemy can move, “attackSpeed” will determine how fast enemy can fire projectile, “EnemyType” variable will determine type of the enemy, so system will use specified graphic for the this enemy. It will send the recent data to the “Entity” class with “update” method. “Slime” abstract class will create weak enemy that moves randomly, “Wolf” abstract class will create slightly stronger and faster enemy than “Slime”, “Orc” abstract class will create slower but stronger enemy than “Wolf”, “Goblin” abstract class will create enemy like Slime, yet moves in fixed route, “SkeletonArcher” abstract class will create stationary enemies which shoot arrow from a distance, “Magmatrum” abstract class will create objects which is a stronger enemy than “Slime” with special effect which changes a terrain to a “Scorched Earth” that damages the player when player touched it, “FinalBoss” abstract class will be used to create the enemy at the last room, so destroying final boss will end the game with a success, it will move fast, track the player and attack him with projectiles from a distance.