GMTB Version 2 Engine Documentation

Class Descriptions

* AI Manager – Class - Singleton
  + Controls all AI specific routines
  + Primarily informs AIs of the player’s current location
* AManager – Abstract Class
  + Allows additional subroutine managers to be created by the game file and be loaded into the core manager in the engine.
* AnimatingEntity – Abstract class
  + Inherits from Entity
  + Handles animation cycle
* Camera2D – Class
  + Overrides default camera to produce a player following camera
* Collidable – Interface
  + Gives Collision manager access to collision methods and variables in an entity
  + Implement this interface in an entity to reveal it to the Collision Manager
* CollisionManager – Class - Singleton
  + Primary Collision detector
* CoreManager – Class – Singleton
  + Runs all other managers required for gameplay
  + Suspends manager operation based on gamestate
* DialogueBox – Class – Singleton
  + Draws text on screen during dialogue
* Door – Class
  + Colidable object that triggers new level
* Entity – Abstract Class
  + Main class for physical objects
  + Includes methods for creation, basic collision and object disposal
* EntityManager – Class – Singleton
  + Main manager for creating and disposing entities
    - Note: Disposal not quite perfect
* FullArtRenderer - Class
  + Class to render a characters full screen artwork
    - Not finished
* GameOver – Class
  + Game over screen displayer
  + Runs as a menu
* Global – Static Class
  + Storage for global variables
* hasProximity – Interface
  + Gives Proximity Manager access to the proximity box of an object
    - Used to determine if the player is near an object – separate to collision
* HidingPlace – Class
  + Object the player can hide in
* IEntity – Interface
  + Main interface for Entity class
* IInventory – Interface
  + Interface for Inventory class
* IItem – Interface
  + Interface for Item class
* ILevel – Interface
  + Interface for Level classes
* Input – Class -Singleton
  + Main input detection and manager
* Inventory – Class
  + Records all collected items, gives them appropriate display location in inventory display.
* IWall – Interface
  + Interface for invisible walls
* Level – Abstract Class
  + Level descriptors inherit from this
* LevelManager – Class -Singleton
  + Lists all available levels during initialization
  + Handles level loading
* MainMenu – Class
  + Displays main Menu
  + Calls game start routine
* MenuManager – Class – Singleton
  + Ensures only one of each menu is created
* PauseMenu – Class
  + Same as Main
* Player – Class
  + Main player class
* ProximityManager – Class -Singleton
  + Same as collision manager but triggers proximity events
* RoomManager – Class – Singleton
  + Renders background
* SaveLoadManager – Class – Singleton
  + Controls file saving and loading
* SaveData – Struct – Serializable
  + Save file contents
* SceneManager – Class -Singleton
  + Controls creation of entity textures
  + Calls all entities to be drawn on command
* Script – Class – Singleton
  + Loads Dialogue from entity
  + Sends Dialogue line by line to DialogueBox
* SolidObject – Class
  + Unmoving object with collision

Class Interactions

Player Creation process

EntitiyManager

IEntity newEntity<T>(PlayerIndex pPlayerNum) where T : IEntity, new()

{ Create PlayerObject

Give PlayerObject UID

Add PlayerObject to master List

Return PlayerObject

}

GameManager

InitializeGame()

{ Call Entity Manager to create PlayerObject

Pass PlayerObject and coordinates to SceneManager

Call LevelManager to start L1

}

LevelManager

NewLevel(string LevelID)

{ Find LevelID and Begin }

SceneManager

newEntity(IEntity createdEntity, int x, int y)

{

Add Entity to Master List

Load Entity Texture from itself

Position at X,Y

}

Player

Constructor ()

{ Call Parent constructor

Do own thing }

Texture { Get {return mTexture}}

Entity

constructor ()

{ Do own thing }

AnimatingEntity

constructor ()

{ Call Parent constructor

Do own thing }