

Tiny Tanks game structure.

The main game loop is controlled by game.cpp

Math Library

The math library is used primarily through the actor class to control movement etc of all game assets

Actor class (previously sprite class):

The Actor class is the parent class for each game object. An Actor object contains all base element required to display an object into the game and do basic transforms.

- A texture (class object)
- A parent node always present but parent is optional (class object)
- Movement functions
- A transform
- Defined CLASSID type through enum
- Rotate Point

The actor class should push through updates on the following objects tied to it

- Update texture
 - For animated sprites
- Node
 - Check if parent should be kept or not
 -

SpriteBatch class

Responsible for drawing all assets to screen. Is designed to work with lists ideally the following lists will be in use for the game

1. Player
 - a. Tank
 - b. Turret
 - c. Effects
2. Enemies
 - a. Tank
 - b. Turret
 - c. Effects
3. Bullets
 - a. Player
 - b. Baddie

Handles the following

- Draw asset
- Drawn from rotate point
- UV LOC

Spritebatch.begin

Texture Class

1. Holds texture and UV information
2. Sets initial texture rotation through
3. UV location

The texture class has been upgraded in the following ways:

Texture Manager

Animation abilities

The texture class now deals with animations in the following way:

1. Texture manager loads up all animation info for a given sprite
 - a. Including multi cycles
2. Textures ask for the information of a given sprite and control the linking of multiple spritesheets

Game Class

The game class handles loading initial settings and running the main game loop

Pushes the following game loops

1. Actor update
 - a. Actor drives controlled objects (see actor class definition)
2. Game object drawing
3. HUD update
- 4.

The Game class has the following “list” of objects that get iterated every frame

- Bullets - actor
- Player - actor
- Enemies - actor
- HUD
- Game Manager (TBC)

Node Class

The node class stores transform information about the current node and holds a reference to an optional parent node enabling each node to get its parents transform.

Gamestate manager

Control the current game state with a stack

- Load splash by default
-