<<Not a Singleton>>
ParseXML

Class Gamedata has a map that contains
constants and numeric values used by
most of the classes in the animation.

Gamedata

-gameData: map<string, string>
-parser: ParseXML

+displayData(): void
+getXmlInt(): int
+getXmlFloat(): float
+getXmlStr(): string

<>
Drawable

-position: Vector2f
-velocity: Vector2f
-maxVelocity: Vector2f

+<<const>> *draw(): void*
+*update(): void*
+X(): float
+X(x:float): void
+Y(): float
+Y(y:float): void
+getPosition(): Vector2f
+getVelocity(): Vector2f
+setPosition(pos:const Vector2f&): void
+setVelocity(vel:const Vector2f&): void

Frame

-screen: const SDL_Surface
-spriteSurface: const SDL_Surface
-spriteWidth: const Sint16
-spriteHeight: const Sint16
+spriteSourceX: Uint16
+spriteSourceY: Uint16

+<<constructor>> Frame(fn:const string&)
+<<const>> draw(x:Sint16,y:Sint16): void
+<<const>> getWidth(): Sint16
+<<const>> getHeight(): Sint16
+Frame(spr:SDL_Surface,sprite_width:Uint16,
       sprite_height:Uint16,src_x:Uint16,
       src_y:Uint16): constructor

main()

Manager

-screen: SDL_Surface* const
-orb: Sprite*

+play(): void
-drawBackground(): void

Sprite

#frame: Frame* const
-name: string

+<<constructor>> Sprite(pos:Vector2f,vel:Vector2f,
                        n:string,f:const Frame*)
+<<constructor>> Sprite(s:const Sprite&)
+<<const>> draw(): void
+update(): void
+<<const>> getFrame(): void
+deleteFrame(): void
#getDistance(const Sprite*): int

<<singleton>>
Clock

-started: bool
-paused: bool
-frames: unsigned
-timeAtStart: unsigned
-timeAtPause: unsigned
-ticks: unsigned
-currTicks: unsigned
-prevTicks: unsigned

+getInstance(): Clock&
+<<const>> getTicks(): unsigned

<<singleton>>
IOManager

-screen: SDL_Surface
-font: TTF_Font

+getInstance(): static IOManager&
+getScreen(): SDL_Surface*
+loadAndSet(filename:const string&): SDL_Surface*
+printMessageAt(msg:const string&,x:Uint32,
               y:Uint32): void

The IOManager class is a singleton that's used by most
of the classes in the animation. Most importantly, it
contains the screen, on which everything is drawn. There
should, normally, be one screen in the animation.
IOManager is also useful for loading sprite surfaces,
and printing text on the screen

Figure 1: **Class Diagram of a framework for animations**.

CpSc 416/616: Object Oriented Programming
Computer Science Department
Clemson University
Animation Framework
Brian Malloy, PhD
February 14, 2013