

Siftables

Class

Application

public

Siftables()

private

Application\_Exit(object sender, EventArgs e) : void

Application\_Startup(object sender, StartupEventArgs e) : void

Application\_UnhandledException(object sender, ApplicationUnhandledExceptionEventArgs e) : void

ReportErrorToDOM(ApplicationUnhandledExceptionEventArgs e) : void

BaseApp

Class

public

BaseApp(List<Cube> cubes)

FrameRate : int

Setup() : void

Tick() : void

private

Cubes : List<Cube>

MainWindow

Class

UserControl

public

MainWindow()

private

cubes : CubeSet

loadAProgramButton\_Click(object sender, RoutedEventArgs e) : void

MainWindow\_Loaded(object sender, RoutedEventArgs e) : void

numberOfCubesUpDown\_ValueChanged(object sender, RoutedProp...

snapToGridButton\_Click(object sender, RoutedEventArgs e) : void

MyApp

Class

BaseApp

This is a generic class that represents any application developed for the emulator or the actual cube platform.

Cube

Class

UserControl

public

Cube()

dimension : int

FillRect(Color c, int x, int y, int w, int h) : void

FillScreen(Color c) : void

Image(string name, [int x = 0], [int y = 0], [int s...

SCREEN\_HEIGHT : int

SCREEN\_MAX\_X : int

SCREEN\_MAX\_Y : int

SCREEN\_MIN\_X : int

SCREEN\_MIN\_Y : int

SCREEN\_WIDTH : int

Nested Types

Side

Enum

Top

Right

Bottom

Left

None

CubeSet

Class

List<Cube>

public

AddCubes(int count) : void

CubeSet(Canvas workspace)

RemoveCubes(int count) : vo...

private

workspace : Canvas

Neighbors

Class

public

Bottom { get; set; } : Cube

Contains(Cube c) : bool

Count { get; } : int

CubeOnSide(Side s) : Cube

IsEmpty { get; } : bool

Left { get; set; } : Cube

Neighbors()

Right { get; set; } : Cube

SideOf(Cube c) : Side

Top { get; set; } : Cube

private

\_neighbors : Cube[]

\_numNeighbors : int

sideUtil(Side s, Cube value) : void

DragAndDropBehavior

Class

Behavior<UIElement>

protected

OnAttached() : void

OnDetaching() : void

private

\_isDragging : bool

AssociatedObject\_MouseLeftButtonDown(object sender, MouseButtonEventArgs e) : void

AssociatedObject\_MouseLeftButtonUp(object sender, MouseButtonEventArgs e) : void

AssociatedObject\_MouseMove(object sender, MouseEventArgs e) : void

DragAndDropBehavior\_MouseEnter(object sender, MouseEventArgs e) : void

DragAndDropBehavior\_MouseLeave(object sender, MouseEventArgs e) : void

mouseClickPosition : Point

parent : DependencyObject