

Intro to Game Programming with Dark GDK Spring 2015

Topics

- API
- Dark GDK
- Screen Coordinates
- Game Loop
- Simple Animation

API

- Application Program Interface (API)
- API
 - describes an interface of a software system
 - is a set of commands, functions, and protocols programmers can use when building software
 - allows programmers to use predefined functions when developing software
 - examples include: Standard Template Library in C++, Java API, Dark GDK, Google Maps, Android

Dark GDK

- Dark GDK is an API that can be used with C++ to write games
- Dark GDK was developed by The Game Creators in the UK (<http://www.thegamecreators.com/>)
- Dark GDK Download is at http://www.thegamecreators.com/?m=view_product&id=2128&page=download

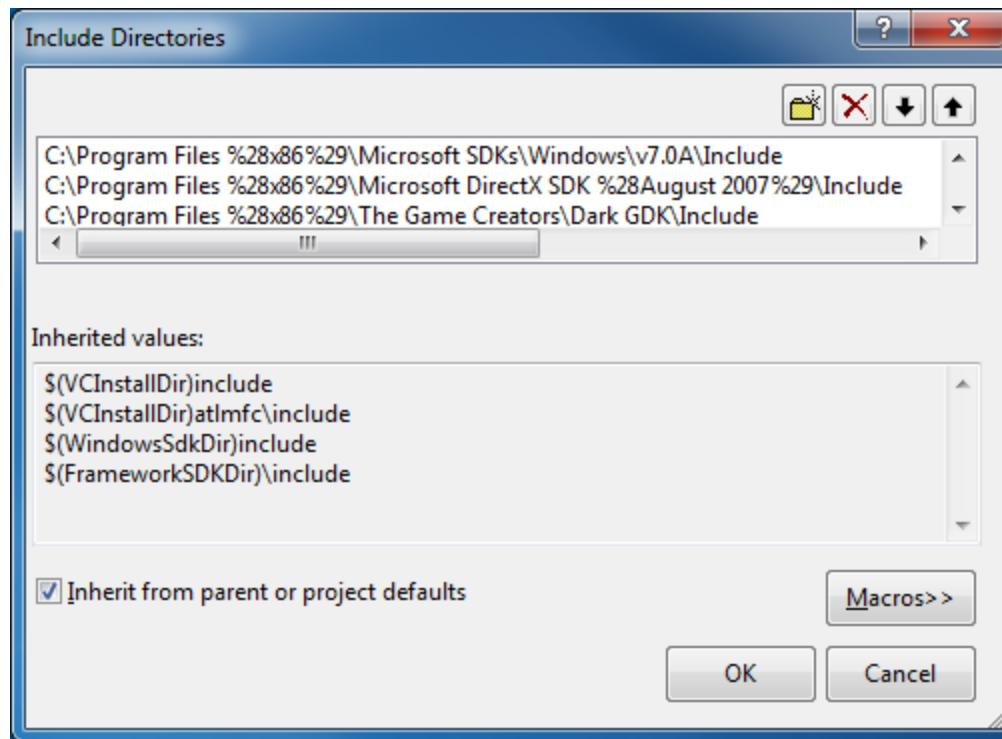
Dark GDK Setup

- Dark GDK Setup folder is in CS250 Public
- README Instructions exist in the folder
- The setup is for Visual Studio 2010 not Visual Studio 2008
- I have only tested this setup on Windows 7 systems ...
- Dark GDK machines in the CS Lab, Marsh Labs, and 24-hour library lab

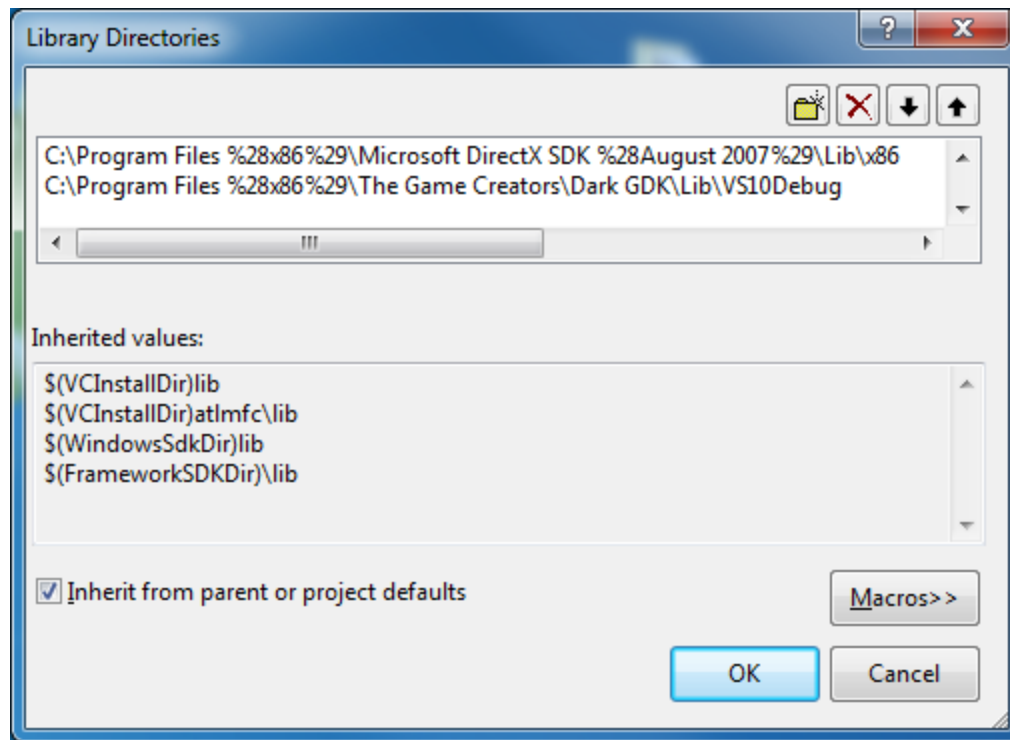
Dark GDK HelloWorld

- Grab the Visual Studio 2010 solution HelloDarkGDK from CS250 Public and place this solution on your Desktop
- Open the solution
- Rebuild the solution
- Start Without Debugging

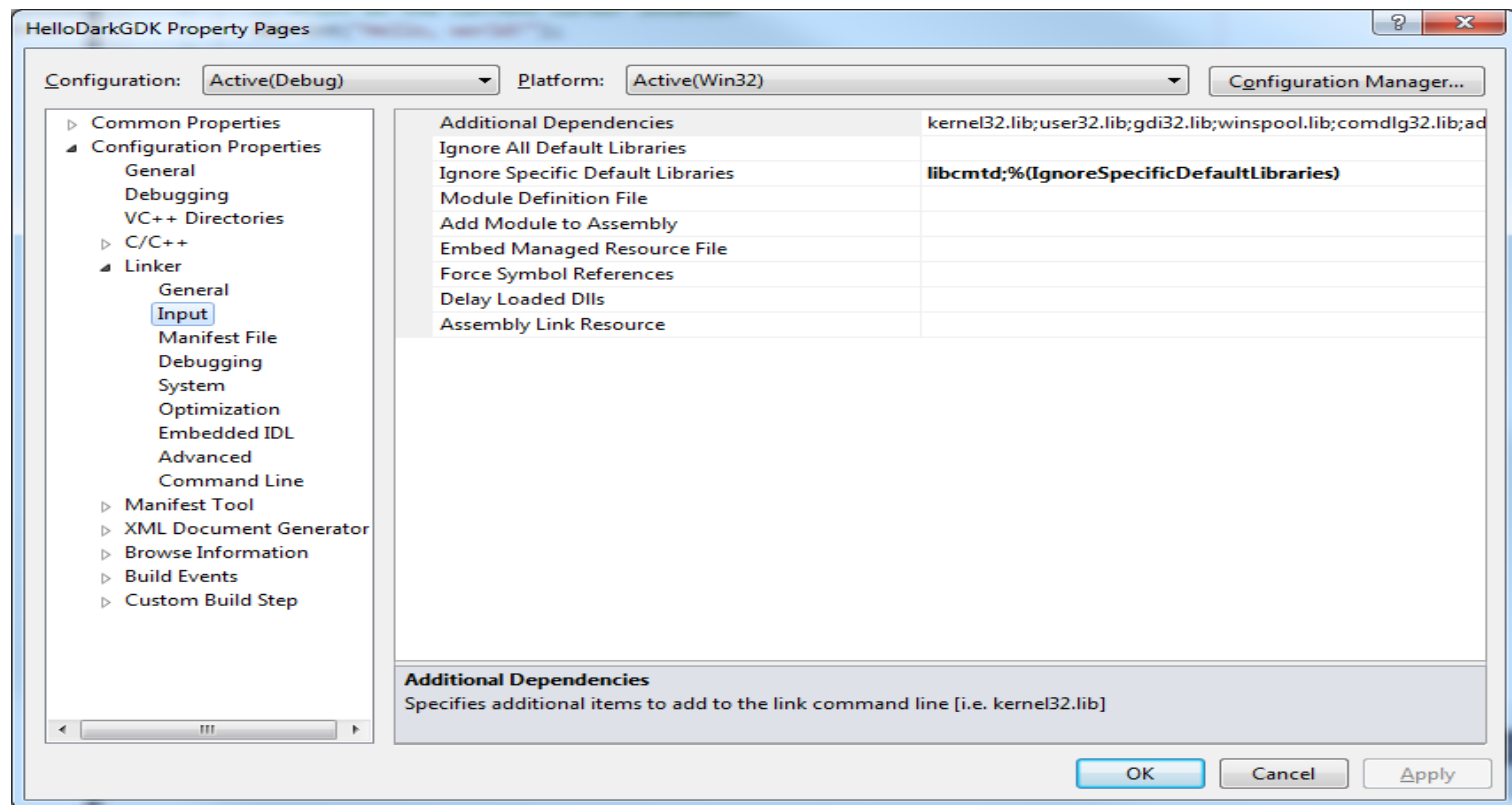
Include Directories



Library Directories



Ignore Specific Default Libraries



HelloDarkGDK

```
#include "DarkGDK.h"

void DarkGDK()
{
    // Print at the current cursor location
    dbPrint("Hello, world!");

    // Wait for input from the keyboard
    dbWaitKey();
}
```



Simple Graphics

```
void DarkGDK ()
{
    int x = 25;
    int y = 25;

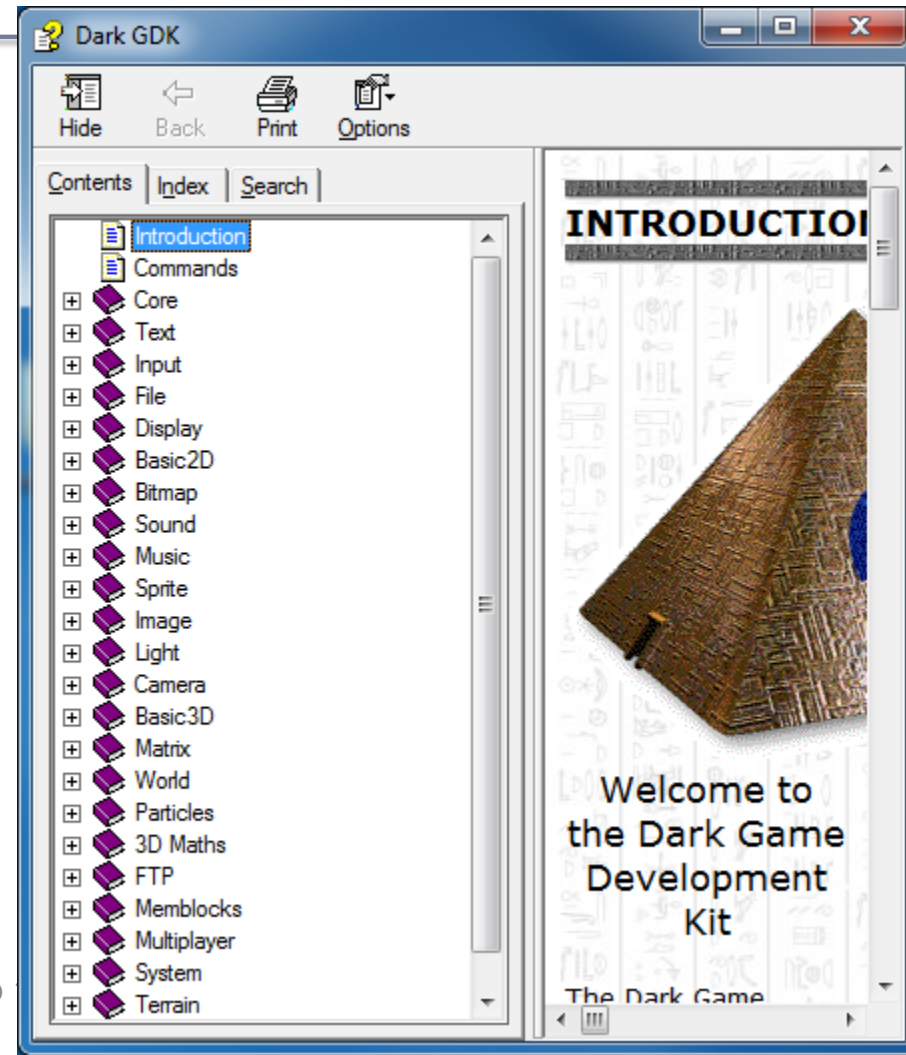
    for (int radius = 5; radius <= 25; radius += 5)
    {
        dbCircle (x, y, radius);
    }

    dbWaitKey ();
}
```



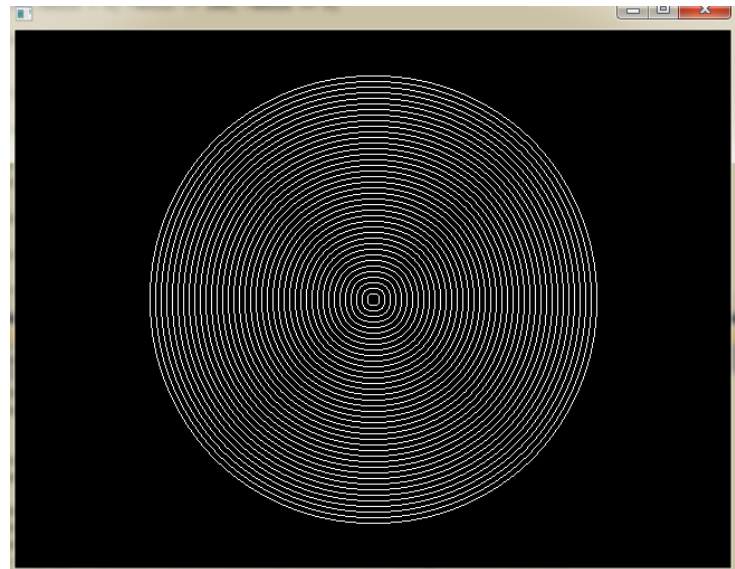
DarkGDK Documentation

- Local Disk (C:)->
Program Files (x86)->
The Game Creators->
Dark GDK->
Documentation->
Dark GDK



Problem

- Using the help functions, edit the program to draw 40 concentric circles starting from the center of the screen, where the radius starts at 5 pixels and grows by 5 pixels for each circle

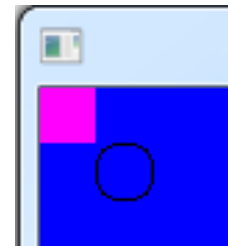


Screen Info

- Default 640 pixels wide by 480 pixels high
- `dbSetDisplayMode (width, height, colorDepth);`
- RGB color is RGBA or a 32-bit value storing the red channel value (8-bits), green (8-bits), blue (8-bits), and alpha channel (8-bits) used for transparency
- `dbInk (foregroundColor, backgroundColor);` where the `backgroundColor` only applies to text

Color Example

```
void DarkGDK ()
{
    DWORD blue = dbRGB (0, 0, 255);
    DWORD magenta = dbRGB (255, 0, 255);
    DWORD black = dbRGB ( 0, 0, 0);
    //clear window to blue background
    dbClear (0, 0, 255);
    dbInk (magenta, black);
    dbBox (0, 0, 20, 20);
    dbInk (black, black);
    dbCircle (30, 30, 10);
    dbWaitKey ();
}
```



Color Example

- You can use a Color Picker to help you select the appropriate color:
 - <http://www.colorpicker.com/>

Game Loop Animation

- Game Loop
 - special loop used in games and animation programs
 - contains update code
 - synchronizes the screen refresh

```
dbSyncOn ();           // we will handle screen updates
dbSyncRate (60);       // maximum times per second screen is updated
while (LoopGDK ())     // LoopGDK controls loop executions per second
{
    updateScreen ();    // you write some kind of update code here
    dbSync ();         // forces a screen update
}
```

Circle Animation

```
void DarkGDK ()
{
    int x = 20, y = 20;

    dbSyncOn ();
    dbSyncRate (60);

    while (LoopGDK ())
    {
        dbClear (0, 0, 255);
        dbCircle (x, y, 10);
        ++x;
        ++y;
        dbSync ();
    }
}
```

Circle Animation

- What happens if we remove `dbClear` from the code on the previous slide?
- Modify the code on the previous slide to animate two circles.
 - One starts from the top left and moves down to the bottom right,
 - and the other starts from the top right and moves down to the bottom left

Circle Animation

- Modify the code so that the radius of the circles starts at 1 and grows by 1 during every iteration of the game loop.
- Modify the code so that the radius of the circle starts at 1 and grows by 1 for every 5th iteration of the game loop