

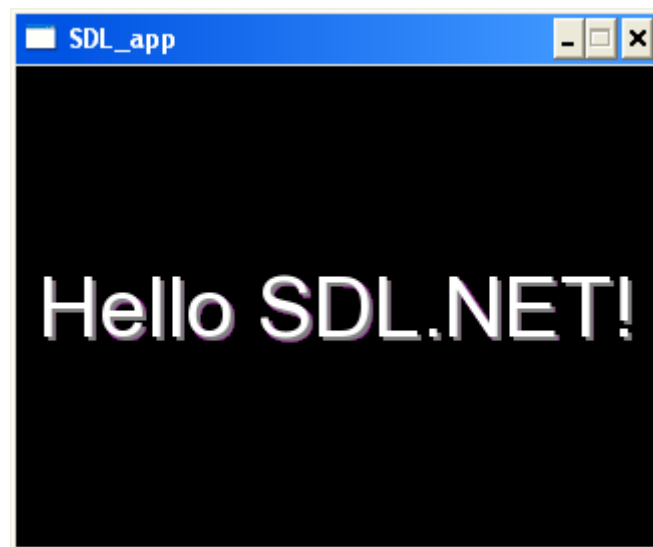
SDL LIBRARY

FONTS

Tekst is een moeilijke opgave om in producten- games- te integreren. De moeilijkheid ligt in de localization: niet iedereen spreekt bijvoorbeeld Engels. Misschien is het daarom niet slecht om de hoeveelheid tekst te minimaliseren of tekst te vervangen door iconen... Anyway, tekst is er , en blijft belangrijk, en daarom dit hoofdstuk.

TRUETYPE FONTS

TrueType fonts worden gerenderd in een Surface object waarna ze geblit wordt op het video sub-systeem. We zullen een project maken waar 2 surface objecten gerenderd worden, alwaar de ene tekst informatie bevat, terwijl de andere schaduw is.



```
public class MyProject
{
    private Surface m_VideoSurface;
    private Surface m_FontForegroundSurface;
    private Surface m_FontShadowSurface;

    public MyProject()
    { }

    public void Run()
    {
```

```

m_VideoSurface = Video.SetVideoMode(320, 240, false, false,
false, true);
SdlDotNet.Graphics.Font font = new
SdlDotNet.Graphics.Font(@"Arial.ttf", 42);
// Create the Font Surfaces
m_FontForegroundSurface = font.Render("Hello SDL.NET!",
Color.White);

m_FontForegroundSurface = font.Render("Hello SDL.NET!",
Color.White);
m_FontShadowSurface = font.Render("Hello SDL.NET!",
Color.White, Color.FromArgb(255, 0, 255));
m_FontShadowSurface.AlphaBlending = true;
m_FontShadowSurface.Alpha = 128;
m_FontShadowSurface.Transparent = true;
m_FontShadowSurface.TransparentColor = Color.FromArgb(255, 0,
255);

Events.Quit += new EventHandler<QuitEventArgs>(Events_Quit);

Events.Fps = 25;
Events.Tick += new EventHandler<TickEventArgs>(Events_Tick);
Events.Run();
}

void Events_Tick(object sender, TickEventArgs e)
{

    m_VideoSurface.Fill(Color.Black);

    m_VideoSurface.Blit(m_FontShadowSurface, new
    Point(m_VideoSurface.Width / 2 - m_FontShadowSurface.Width / 2
+ 2, m_VideoSurface.Height / 2 - m_FontShadowSurface.Height /
2 + 2));

    m_VideoSurface.Blit(m_FontForegroundSurface, new
    Point(m_VideoSurface.Width / 2 - m_FontForegroundSurface.Width
/ 2, m_VideoSurface.Height / 2 - m_FontForegroundSurface.Height
/ 2));

    m_VideoSurface.Update();
}

void Events_Quit(object sender, QuitEventArgs e)
{
}
}

```

Enkel de volgende 3 lijnen code zijn nieuw:

```

SdlDotNet.Graphics.Font font = new SdlDotNet.Graphics.Font( @"Arial.ttf",
42 );
m_FontForegroundSurface = font.Render( "Hello SDL.NET!", Color.White );
m_FontShadowSurface = font.Render( "Hello SDL.NET!", Color.Gray );

```

1. Creatie van een nieuwe instantie van de Font klasse (de 2^{de} parameter specificeert de font size).
2. Render de text met een voor –en/of achtergrond kleur op het Surface object

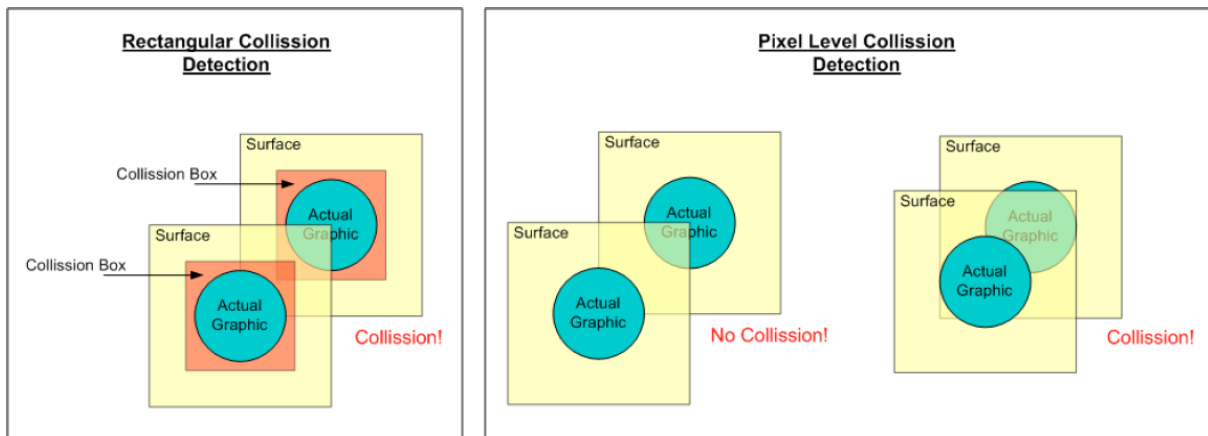
FONT INFORMATIE (JARGON)

De grootte van een font is gelijk aan de hoogte van een karakter. Maar verschillende letters hebben een descent, bijvoorbeeld p,g,q, die onder een denkbeeldige lijn komen, terwijl andere een ascent hebben, bijvoorbeeld m, n , b, die enkel boven een denkbeeldige lijn komen. Deze denkbeeldige lijn noemt men de base line. De ascent + descent geeft je de hoogte van het karakter.

COLLISION DETECTION

In de meeste games is collision detection een essentieel onderdeel. Collision detection wil zeggen dat er gecontroleerd wordt of 2 objecten elkaar raken. Er zijn 2 methoden om dergelijks te checken:

- Rectangle Collision Detection
- Per Pixel Collision Detection



collision detection een essentieel onderdeel. Collision detection wil zeggen dat er gecontroleerd wordt



We gaan 2 personages laten bewegen door de gebruiker. Personage1 beweegt door de linker en rechterpijl in te drukken, terwijl personage2 beweegt door bijvoorbeeld Q en W. Indien de personages elkaar raken verschijnt er AAAAHHH op het scherm.

PROJECT 1: COLLISION DETECTION

Herneem project 2 van sessie 3 en 4 – een eerste animatie en voeg het volgende toe: In de manager klasse

```
SpriteCollection coll;
Character ch1;
Character ch2;

private void BuildApp()
{
    coll = new SpriteCollection();
    ch1 = new Character(m_Video,1);
    ch2 = new Character(m_Video,2);

    coll.Add(ch1);
    coll.Add(ch2);

    coll.EnableKeyboardEvent();
    coll.EnableKeyboardDownEvent();
    coll.EnableKeyboardUpEvent();
    coll.EnableTickEvent();
}
```

Verder schrijven we in de manager klasse nog volgende functie:

```
private bool GetBoundingBoxCollisionStatus()
{
    if (ch1.CollisionRectangle.Intersects(ch2.CollisionRectangle))
        return true;

    return false;
}
```

En de TICK afhandeling:

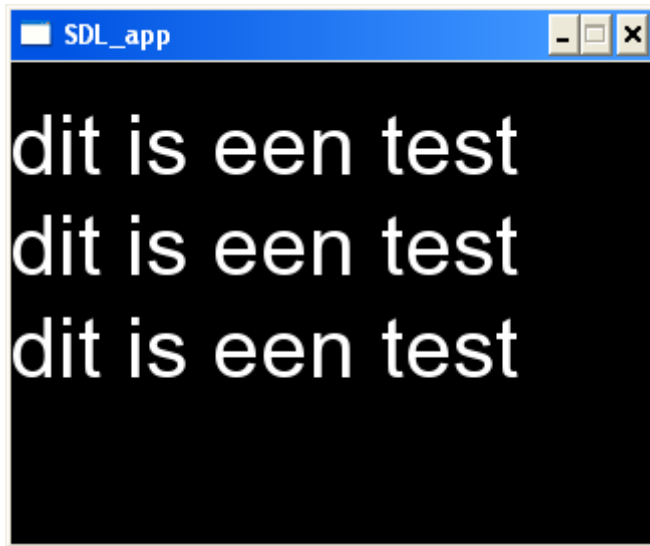
```
void Events_Tick(object sender, TickEventArgs e)
{
    if (GetBoundingBoxCollisionStatus())
    {
        SdlDotNet.Graphics.Font font = new
        SdlDotNet.Graphics.Font(@"Arial.ttf", 42);

        Surface aFontSurface = font.Render("AAAAAAHHHHH!",
        Color.GreenYellow);
        m_Video.Blit(aFontSurface);
        m_Video.Update();
    }
    Else
        m_Video.Update();
}
```

OEFENING DE CHARACTER KLASSE HERSCHRIJF JE ZELF ZODAT 2 GEBRUIKERS EEN PERSONAGE KUNNEN BESTUREN (DE ENE MET PIJLTJES TOETSEN, DE ANDERE MET BIJVOORBEELD Q EN W).

OEFENREEKS

1.ONTWERP EEN PROGRAMMA DAT TEKST VAN BOVEN NAAR BENEDEN LAAT SCROLLEN OVER HET SCHERM TERWIJL DEZE HANDELING ZICH OOK ELKE 2 SECONDEN HERHAALT.



TIP: 2 SECONDEN KAN MET BEHULP VAN EEN TIMER INGESTELD WORDEN:

```
SDLDOTNET.CORE.TIMER
```

DEZE TIMER HEEFT VOLGENDE FUNCTIE:

```
SDLDOTNET.CORE.TIMER.DELAYSECONDS(2); -> HET GEVOLG IS DAT WE DIE 2  
SECONDEN NIET RESPONSIEF ZIJN IN DE APPLICATIE, DAAROM MOET DEZE FUNCTIE (MET ALS  
ENIGE DOEL 2 SECONDEN WACHTEN, EN EEN EVENT NAAR DE MAIN STUREN DAT ZEGT DAT DE  
2 SECONDEN OM ZIJN) IN EEN APARTE THREAD UITGEVOERD WORDEN
```