

# Period 5 - TD

## Föreläsning 3 av 3

Jonas Petersson

Felsökning / Felhantering

RenderTargets

Kamera

Lite om Windows Forms

# Felhantering av Exception (DivideByZeroException)

```
int Divide(int x, int y)
{
    return x / y;
}

int result = 0;
if (Mouse.GetState().LeftButton == ButtonState.Pressed)
{
    result = Divide(a, b); // kan bli gallet om b är 0
}
```

# Använd en if-sats

```
int Divide(int x, int y)
{
    return x / y;
}

int result = 0;
if (Mouse.GetState().LeftButton == ButtonState.Pressed)
{
    if (b != 0) // if:a bort "felet"
        result = Divide(a, b);
}
```

# Utöka med utskrift

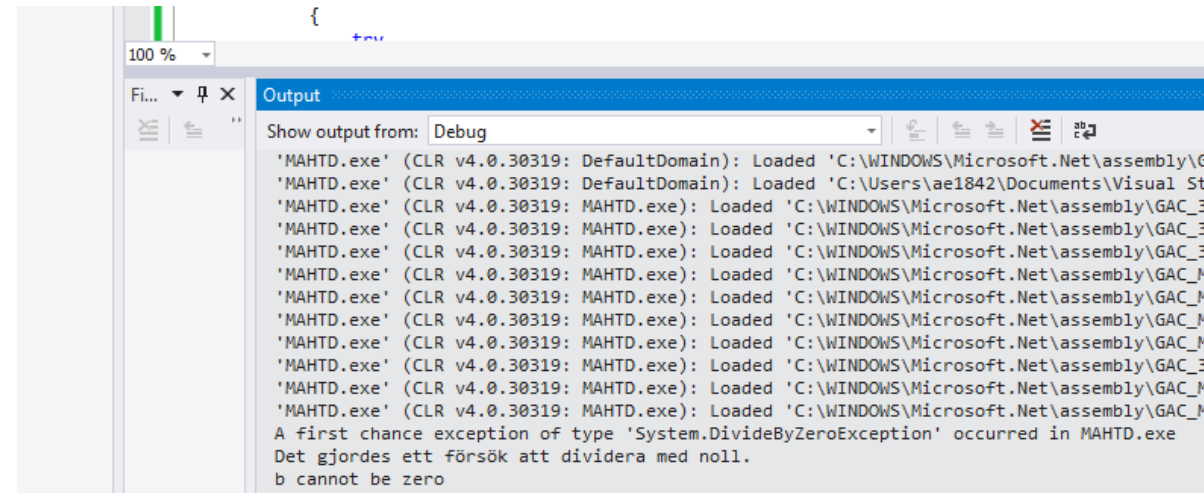
```
int Divide(int x, int y)
{
    return x / y;
}

int result = 0;
if (Mouse.GetState().LeftButton == ButtonState.Pressed)
{
    if (b != 0) // if:a bort "felet"
        result = Divide(a, b);
    else
        Console.WriteLine("b cannot be zero"); // Skriv ut till output-fönstret i Visual Studio
}
```

# Lös med try och catch

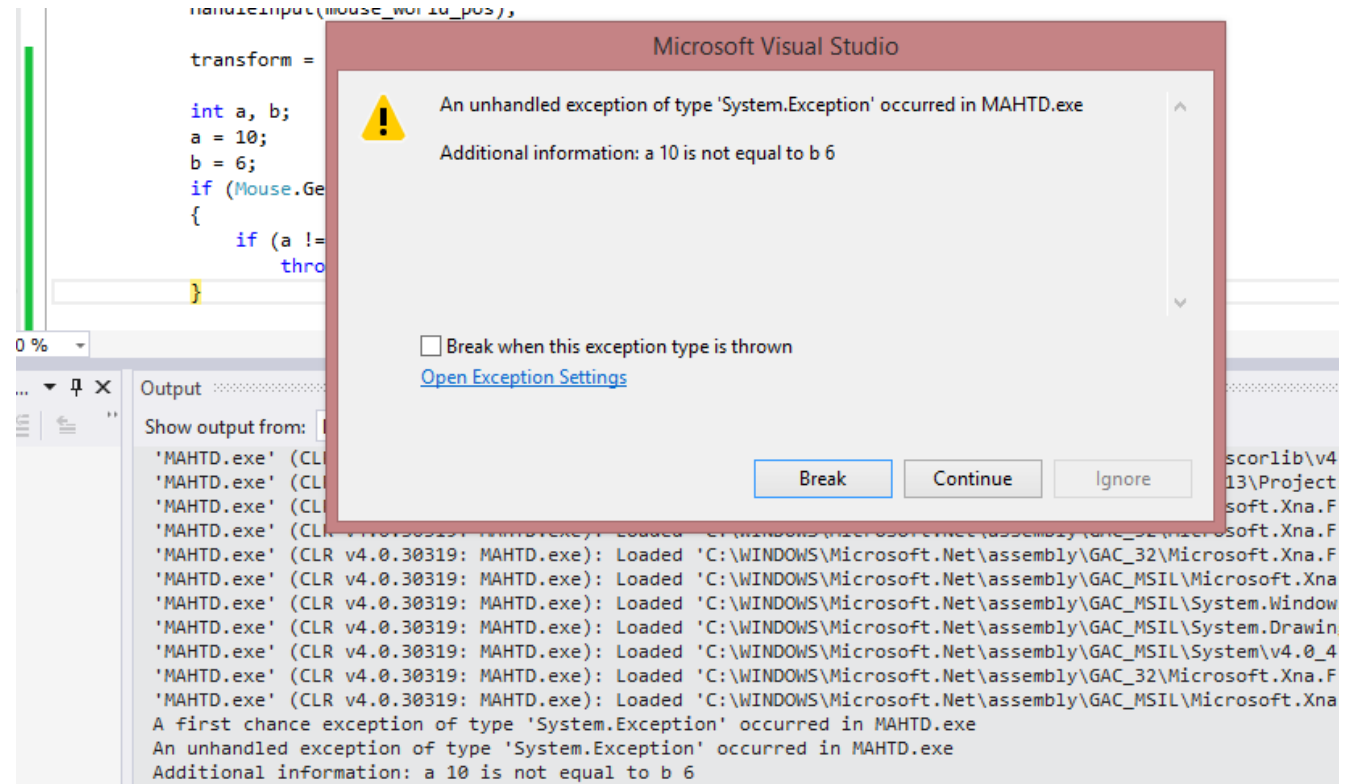
```
int Divide(int x, int y)
{
    return x / y;
}

int result = 0;
if (Mouse.GetState().LeftButton == ButtonState.Pressed)
{
    try
    {
        result = Divide(a, b);
        Console.WriteLine(result);
    }
    catch (DivideByZeroException ex)
    {
        Console.WriteLine(ex.Message);
        Console.WriteLine("b cannot be zero");
    }
}
```



# Kasta ett Exception med throw

```
int a, b;  
a = 10;  
b = 6;  
if (Mouse.GetState().LeftButton == ButtonState.Pressed)  
{  
    if (a != b)  
        throw new Exception(String.Format("a {0} is not equal to b {1}", a, b));  
}
```



# Felsökning i Visual Studio

Visar på datorn.

# RenderTarget

## Front Buffer



SpriteBatch.Draw() ritar till aktivt RenderTarget

# Frame 2

Aktivt RenderTarget

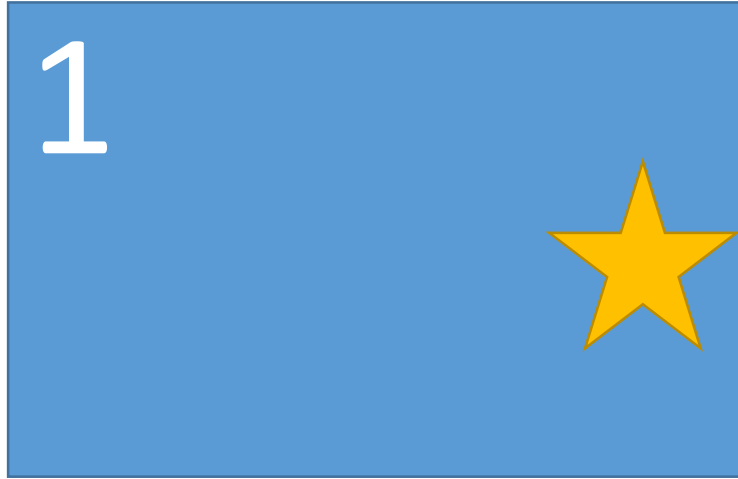


## Back Buffer

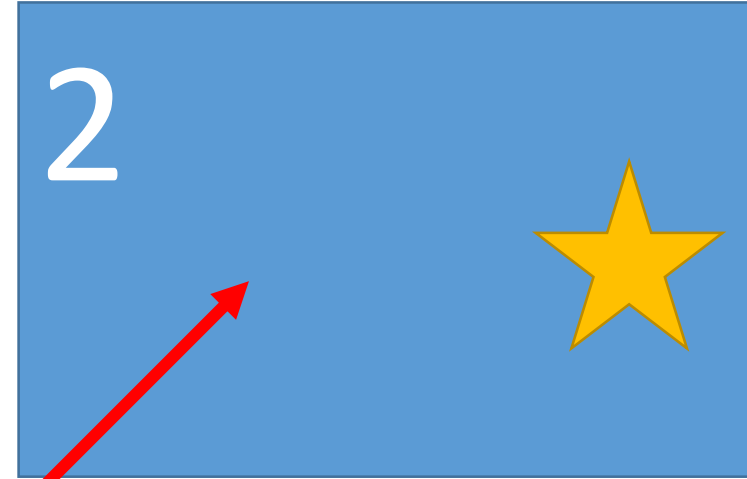


# RenderTarget

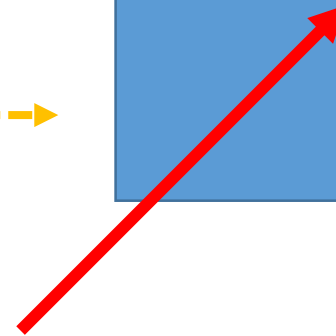
1. Front buffer (det vi ser på skärmen vid aktuell frame)



2. Back buffer (bild som ska visas nästa frame)



Aktivt RenderTarget



3

Skapa ett nytt RenderTarget

```
backgroundLayer = new RenderTarget2D(GraphicsDevice, 1280, 720);
```

# Nyttja RenderTarget för pixelperfekt kollision

1. Skapa ett nytt RenderTarget.

```
renderTargetName = new RenderTarget2D(GraphicsDevice, 1280, 720);
```

2. Skapa en ny SpriteBatch

```
SpriteBatch sb = new SpriteBatch(device);
```

3. Ändra aktivt RenderTarget till det egna

```
device.SetRenderTarget(renderTargetName);
```

4. Rensa aktivt RenderTarget (`renderTargetName`) med transperant färg (RGBA : 0000)

```
device.Clear(Color.Transparent);
```

5. Rita till `renderTargetName` (aktivt RenderTarget)

```
sb.Begin();
```

```
// Rita alla objekt som ska finnas i renderTargetName
```

```
// sb.Draw(...)
```

```
sb.End();
```

6. Sätt tillbaka aktivt RenderTarget till DefaultRenderTarget (Back Buffern)

```
device.SetRenderTarget(null);
```

7. Nyttja `renderTargetName` för pixelperfekt kollision.

8. Hoppa till steg 3 (alternativt 2) och gör allt igen för nästa frame.

# Skapa en bana i TD som nyttjar RenderTarget

Visa på tavlan.

# Kamera, en matris som "ändrar" på ett objekts translation, skala, rotation - transformera objekt

```
float zoom;
```

```
Matrix transform;
```

```
Vector2 pos;
```

```
public Vector2 origin;
```

```
float rotation;
```

```
float speed;
```

```
public float Zoom
{
    get { return zoom; }
    set
    {
        zoom = value;
        if (zoom < 0.1f) // En negativ zoom flippar bilden
            zoom = 0.1f;
    }
}
```

```
public void Move(Vector2 displacement, bool respect_rotation = false)
{
    if (respect_rotation)
    {
        displacement = Vector2.Transform(displacement, Matrix.CreateRotationZ(-rotation));
    }

    pos += displacement;
}
```

# Kamera-matris

```
public Matrix get_transformation()
{
    transform =
        Matrix.CreateTranslation(new Vector3(-pos, 0)) *
        Matrix.CreateTranslation(new Vector3(-origin, 0)) *
        Matrix.CreateRotationZ(Rotation) *
        Matrix.CreateScale(new Vector3(Zoom, Zoom, 1)) *
        Matrix.CreateTranslation(new Vector3(origin, 0));

    return transform;
}
```

Flyttar alla objekt  
i motsatt riktning  
till kameran.

Inget djup,  
därav 0 på Z  
i Vector3

objekten ska  
rotera och skalas  
från mitten av  
kameran

flytta tillbaka  
objekten så de  
hamnar på rätt  
ställe vid utritning

# Kamera-exempel (på tavlan)

```
Update()
```

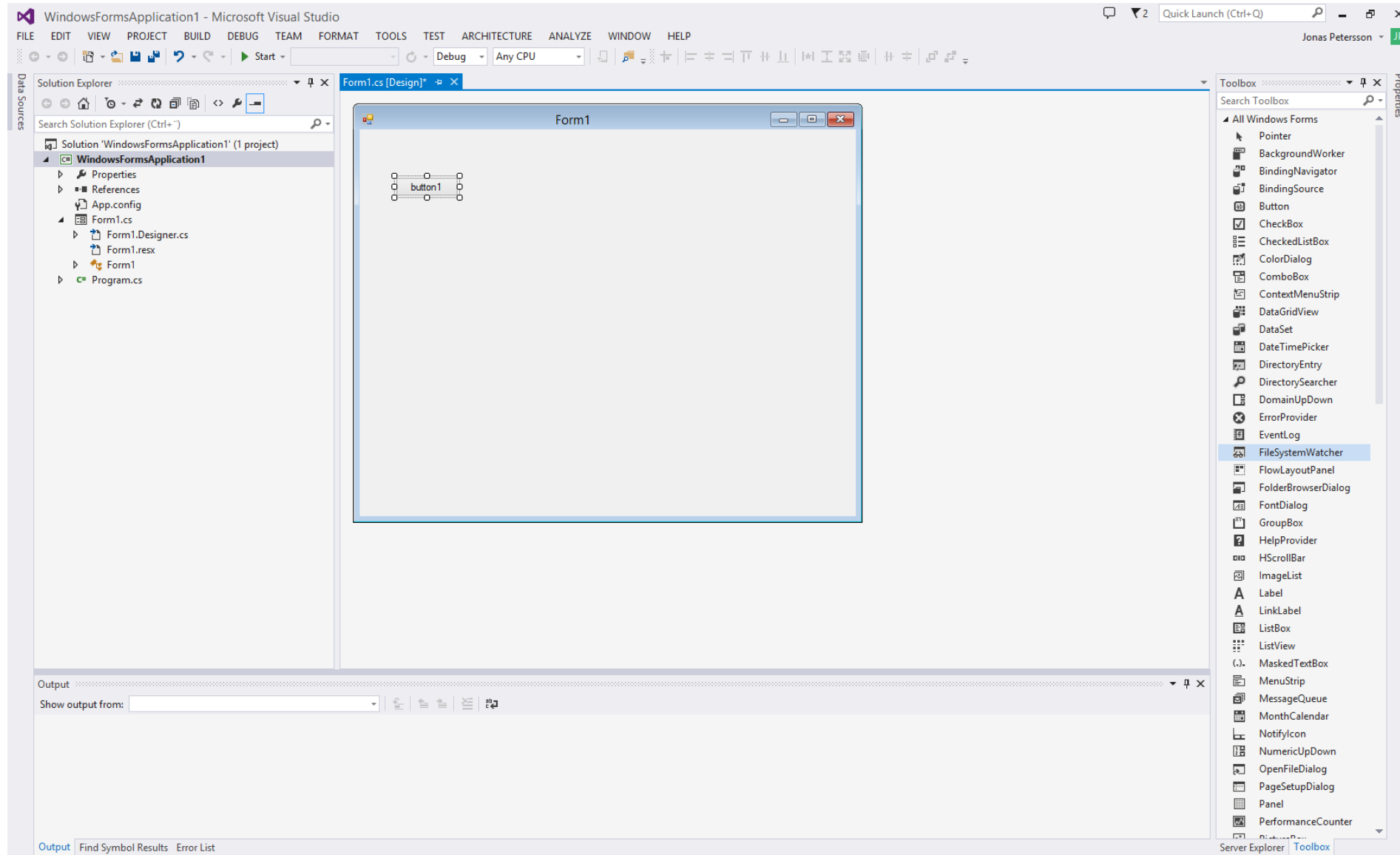
```
    transform = camera.get_transformation(); // Uppdatera transformations-matrisen
```

```
Draw() // nyttja en annan Begin(...) som använder en transformation för att bearbeta objekten innan utritning  
    spriteBatch.Begin(SpriteSortMode.Deferred, BlendState.AlphaBlend, null, null, null, null, transform);
```

```
public Vector2 WorldToScreen(Vector2 world_position)  
{  
    return Vector2.Transform(world_position, transform);  
}
```

```
public Vector2 ScreenToWorld(Vector2 screen_position)  
{  
    return Vector2.Transform(screen_position, Matrix.Invert(transform));  
}
```

# Lite om Windows Forms



Istället för en spelloop  
nyttjar man så kallade events.

# Sista föreläsningen den här kursen 😊? ☹️?

- Spline finns färdig på it'slearning!  
Man behöver alltså inte implementera någonting!  
På föreläsning P5 F1 finns det info om Spline.
- Kursschema för resten av kursen (kolla nästa bild).
- Frågor på labbar!  
Övrig tid mejla mig.
- Något som vi ska ta upp och diskutera redan nu?



**Vecka 49, 2014**

Ons	3 Dec	13:15-15:00	TGSPA14h		TSJEER	K2B105	Föreläsning	2014-04-22
Tor	4 Dec	09:15-12:00	TGSPA14h	Grupp blå		K2D157	Laboration	2014-11-18
		09:15-12:00	TGSPA14h	Grupp röd		K2D155	Laboration	2014-11-18
Fre	5 Dec	09:15-12:00	TGSPA14h	Grupp grön		K2D155	Laboration	2014-08-12

**Vecka 50, 2014**

Tis	9 Dec	09:15-12:00	TGSPA14h	Grupp grön		K2D155	Laboration	2014-04-22
		13:15-16:00	TGSPA14h	Grupp röd		K2D155	Laboration	2014-04-22
		13:15-16:00	TGSPA14h	Grupp blå		K2D157	Laboration	2014-08-12
Tor	11 Dec	09:15-12:00	TGSPA14h	Grupp röd		K2D155	Laboration	2014-11-18
		09:15-12:00	TGSPA14h	Grupp blå		K2D157	Laboration	2014-11-18
Fre	12 Dec	09:15-12:00	TGSPA14h	Grupp grön		K2D155	Laboration	2014-08-12

**Vecka 51, 2014**

Tis	16 Dec	09:15-12:00	TGSPA14h	Grupp grön		K2D155	Laboration	2014-04-22
		13:15-16:00	TGSPA14h	Grupp blå		K2D157	Laboration	2014-08-12
		13:15-16:00	TGSPA14h	Grupp röd		K2D155	Laboration	2014-04-22
Tor	18 Dec	09:15-12:00	TGSPA14h	Grupp röd		K2D155	Laboration	2014-11-18
		09:15-12:00	TGSPA14h	Grupp blå		K2D157	Laboration	2014-11-18
Fre	19 Dec	09:15-12:00	TGSPA14h	Grupp grön		K2D155	Laboration	2014-08-12

**Vecka 2, 2015**

Ons	7 Jan	09:15-12:00	TGSPA14h	Grupp blå		K2D157	Laboration	2014-04-22
		09:15-12:00	TGSPA14h	Grupp grön		K2D155	Laboration	2014-04-22
		13:15-16:00	TGSPA14h	Grupp röd		K2D155	Laboration	2014-04-22
Tor	8 Jan	09:15-12:00	TGSPA14h	Grupp grön		K2D155	Laboration	2014-04-22
		09:15-12:00	TGSPA14h	Grupp blå		K2D157	Laboration	2014-04-22
		13:15-16:00	TGSPA14h	Grupp röd		K2D155	Laboration	2014-04-22

**Vecka 3, 2015**

Mån	12 Jan	09:15-16:00	TGSPA14h			K2D155	Redovisning TD	2014-04-22
Tis	13 Jan	09:15-16:00	TGSPA14h			K2D155	Redovisning TD	2014-04-22
Fre	16 Jan	08:15-13:15	TGSPA14h		TSJEER	K2C214	Tentamen (1113)	2014-04-22

Labbtillfällen kvar den här  
Perioden.

Uppsamling.  
Redovisa extrauppgifter för  
övriga perioder.