

KOLLISION (ENKEL)

Vi skall göra en enkel version av Collisiondetection som kanske inte alltid är att föredra men är den enklaste att implementera. Det vi skall göra är Rectangular Collision detection, vilket betyder att vi inte gör en "äkta" outline på våra sprites utan tar en rektangel runt dem som vi testat kollisionen med. I ett senare spelprojekt skall vi göra kollision på ett mer avancerat sätt.

1. Starta ett nytt projekt som ni kallar Tools_Collision. Försök därefter själva sätta ihop delarna Tools_Movement och Tools_Control.
2. Om player och bollen träffar varandra skall vi öka hastigheten på bollen så att det blir lite svårare. Vi måste även ändra riktning på bollens Y så att den åker åt motsatt riktning vid en kollision.

Så i Update() funktionen precis före `base.Update(gameTime);` lägger vi till följande kod:

```
//Kontrollera Kollision
Rectangle ballRect = new Rectangle((int) ballPosition.X, (int)ballPosition.Y,
ballSprite.Width, ballSprite.Height);

Rectangle playerRect = new Rectangle((int) playerPosition.X,
(int) playerPosition.Y, playerSprite.Width, playerSprite.Height);

if (ballRect.Intersects(playerRect))
{
    //Öka bollens hastighet
    ballSpeed.Y += 50;
    if (ballSpeed.X < 0)
    {
        ballSpeed.X -= 50;
    }
    else
        ballSpeed.X += 50;

    //Skicka bollen bakåt
    ballSpeed.Y *= -1;
}
```

Om ni nu testat koden kommer ni se att bollen studsar iväg så länge ni inte träffar den från sidan med player då ballar den ur. Detta beror på att vi måste lägga till i koden att den bara skall skicka iväg bollen ifall bollen är på väg ner (`ballSpeed.Y > 0`) och inte uppåt (`ballSpeed.Y < 0`). Så därför ändrar vi det i koden.

```
if (ballRect.Intersects(playerRect) && ballSpeed.Y > 0)
```