

Labbrapport 5

D0037D Objektorienterad Programmering

Handledare:

Johannes Hirche

Gustav Sterbrant

Pontus Stenlund

ponste-5@student.ltu.se

Luleå Tekniska Universitet

Skellefteå

2016-01-13

Sammanfattning

Vi skulle fortsätta med att rita ur figurer på skärmen fast den här gången med hastighet och att dom skulle studsas mot skärmens "väggar". Vi skulle även implementera en boll som reflekterades bort från andra objekt och tog bort dom från skärmen.

Innehåll

- Problemspecifikation.....4
- Användarhandledning.....5
- Lösningens begränsningar.....6
- Problem och reflektioner.....7
- Bilagor.....8

Problemspecifikation

Uppgiften gick ut på att med våra fyra implementerade klasser Shape, Square, Triangle och Circle rita ut figurer på skärmen som kunde studsas mot skärmens kanter. Vi skulle även implementera en "boll" av klassen Circle. När bollen sen träffade andra objekt så skulle den tas bort från skärmen. Vi skulle även lägga till en till dimension i vår 2D Vektor klass och i 2D Matrix. Denna dimension gör så att vi kan spara figurernas position och rotation i våra matriser. I och med det så måste vi ju självklart ändra våra äldre funktioner som gjorde uträkningar.

Användarhandledning

För att kunna köra min lösning så är det bara att kompilera programmet och sedan trycka 1,2,3 på tangentbordet för att lägga till nya figurer på skärmen. Trycker man Q,W,E så tar man bort alla figurer av vardera sort. Q för fyrkanter, W för trianglar och E för cirklar. En sak kan vara bra att veta och det är att om man ritar ut ett par cirklar så tar det lite mycket systemkraft och därför så går allt lite segare tills det bara finns en till två stycken kvar av cirkarna kvar på skärmen.

Lösningens begränsningar

Finns säker fler begränsningar som jag inte kan komma på i skrivandets stund.

Det finns ett par begränsningar för min lösning. Det första är att om man ritar ut lite för många cirklar så segar datorn ihop. Detta är för att datorn räknar ut flera punkter runt cirkeln och drar streck mellan dom i takt med `AssignmentApp::Update()` som körs en gång per frame tror jag.

En annan begränsning är min reflektion från andra figurer. För fyrkanter så studsar den inte perfekt vid linjen om den träffar fyrkantens hörn utan bollen åker in en bit innan den reflekteras.

För trianglar så har jag räknat ut som en rektangel runt triangeln som agerar som "träffzon". Vilket fungerar ganska bra när du träffar basen på triangeln men om man träffar diagonalen så ser man det ganska tydligt.

Nu vid inlämningen av laborationen så har jag satt alla figurer till en bestämd storlek och figurerna har ingen hastighet så dom rör sig inte. Men det går att sätta in slumpmässig storlek och hastighet genom att ändra de bestämda argumenten i `AssignmentApp::KeyEvent()`. Direkt i den funktionen så slumpas allt till variabler så det är bara att sätta in dom som argument. Och för att sätta hastighet på figurerna så är det bara att avkommentera kodstycket för varje figur.

Som sagt min lösning tar lite prestanda och bollen kan ha en konstig träffzon i vissa lägen men jag är trots allt rätt nöjd med resultatet.

Problem och reflektioner

Ett av mina största problem som jag har haft under denna laboration är att räkna ut reflektionen. Vi fick formeln inkluderad i beskrivningen men jag var osäker på vad alla bokstäver var för något. T.ex vad är "v" i formeln? Är den en hastighet, en vektor eller något annat? Efter lite googlande så hittade jag svaret. Sen kom jag inte riktigt på hur jag skulle implementera formeln på ett bra sätt. Men efter en stunds diskuterande med min bror så fick jag till funktionen på ett bra sätt.

Ett annat problem som jag har haft är i slutskedet av min laboration så fick jag inte mina funktioner att fungera som jag ville. T.ex så kollision vid andra figurer. Jag satt ett bra tag och funderade. Till slut så kom jag flyttade jag koden till Circle och ändrade ett par saker. När jag sedan trodde att det skulle fungera så stötte jag på nästa problem. Och det var min array som jag sparade mina Shape pekare på inte kunde komma åt i Circle. Efter ytterligare en tankestund så insåg jag att den inte kunde vara privat i AssignmentApp.

Sedan fungerade mitt program ungefär som jag ville. Sedan kom nästa bakslag. Jag insåg att när jag satt positionen som hamnade mina figurer alltid i mitten av skärmen. Jag kunde inte lista ut vad som var fel så jag bad Johannes Hirche om hjälp. Tack Johannes för hjälpen.

Då var det dags att tackla på nästa problem. Jag ville att bollen skulle radera figurerna på figurernas kropp och inte exakt när den träffade mittpunkten. Det var även lite konstiga reflektioner när den träffade någonting fast ingen figur var i närheten. Det kan jag säga har varit mitt största problem. Satt nästan en hel dag att försöka lista ut vad jag gjorde för fel. Satt och ändrade parametrar, testade nya uträkningar men ingenting fungerade.

Bad ännu en gång min bror Patrik till hjälp. Han fick se koden och vi satt och diskuterade och försökte oss på andra lösningar. Sen började Patrik att tänka utanför lådan och bad mig att visa upp hur figurerna ritades ut. Visade honom Render() funktionen. Efter en stund frågade han vad min matris innehöll för data. Jag sa att den innehöll rotation och position.

Då hittade vi den största buggen jag har haft under den här kursen. För att beskriva det så satt jag mittpunkten men sedan när jag skulle rita ut figuren så tog jag och ritade ut figuren på ett helt annat ställe.

Det innebar då att min träffzon inte stämde överens med figuren eftersom mittpunkten var på en helt annan position. Efter jag hade fixat det så fungerar nu träffzonen godtyckligt bra.

Bilagor

Var vänligen och kolla på fronter i rummet D0037D under "Laborationer 2015/Laboration 5" efter källkod.