

Duggainformation

Duggan har två syften:

- Duggan tjänar som inträdesbiljett att få påbörja projektet. Duggan är den tekniska intervjun i rekryteringen av projektgruppsmedlemmar.
- Duggan tjänar som individuell examination av kursmål för betyg 3.

Duggan har en uppgift och bedöms G eller U. Vi kräver tydlig, lättfattlig och underhållbar kod, men vi kommer *inte* vara lika petiga som i labserien. En godkänd lösning ska fungera korrekt, lösningen använder lämplig klasshierarki, den saknar minnesläckor, den undviker kodupprepning, och lösningen har någorlunda namngivning och indentering.

Duggatid

Du har 120 minuter på dig att lösa duggan. Du med rätt till förlängd skrivtid har 160 minuters skrivtid. Rätt till förlängd tid beslutas av LiU centralt.

Ämnesinnehåll

Fokus på duggan kommer vara att skapa en minimal klasshierarki och få den att lösa en uppgift. Detta medför att god kunskap om syntax och semantik rörande klasser behövs.

1. Grunder: typer, variabler, satser, std::vector, funktioner, parametrar, argument
2. Klass: objekt/instans, konstruktor, inkapsling, datamedlem, medlemsfunktion, public, private
3. Arv: basklass, härledd klass, anrop av basklasskonstruktorkonstruktor, protected, anrop av basklassfunktion
4. Polymorfi: slicing, virtuell destruktör, dynamisk/statisk bindning, virtual, pure virtual, override
5. Minneshantering: allokering, avallokering, minnesläcka

Hjälpmedel

Under duggan råder standard tentamensregler. Du får ha med följande hjälpmedel:

- En bok om C++. För boken gäller följande regler:
 - Kommentarer eller noteringar som direkt rör text och exempel på sidan i fråga får finnas i sidmarginalen.
 - Egna sidflikar för att enkelt kunna hitta t.ex. de olika kapitlen är tillåtna.
 - Inga extra ark eller lappar, lösa eller fastsatta, får finnas.
 - Ingen programkod på tomma sidor, insidan av pärmarna, försättsblad, etc.
- Ett A4-ark. Valfritt innehåll på vardera sida. Går ej ersätta med två enkelsidiga ark.
- Penna för att anteckna. Du kan be tentamensvakt om kladdpapper.

Följande får INTE tas med:

- Elektroniska prylar. Till exempel miniräknare, mobiltelefon, hörlurar, tangentbord, smartklocka etc.

Inlogging

Tentamensvakt kontrollerar anmälan vid inpassage till salen. När datorns inloggningsskärm visar “tentamensläge aktivt” kan du logga in som vanligt med ditt LiU-ID och ditt privata lösenord och börja förbereda din programmeringsmiljö. Tentamensvakten kommer under duggan gå runt för att kontrollera legitimation och hjälpmedel.

Alias för kompilering

Under duggan finns det tre alias att använda sig av för kompilering med c++17:

w++17 Rekommenderas!

e++17 Kompilering med alla varningar som fel.

g++17 Kompilering utan varningar.

Frågor

Frågor ska ställas via studentklienten. Detta för att vi ska ha en historik av konversationen samt för att vi ska kunna ge samma hjälp till olika studenter. Räck upp handen om klienten, tangentbordet, musen eller datorn inte verkar fungera som de ska.

C++ referens

Det finns tillgång till valda delar av cpreference.com. Du måste starta webbläsaren via skrivbordsikonen “Web access” för att komma åt sidan.

Inlämning

När du har en lösning som kompilerar utan varningar, kör utan minnesläckor och löser uppgiften skickar du in den via studentklienten. Din inlämning bedöms direkt¹ och du får svar via studentklienten. Om du får betyg G är det bara att spara, avsluta, logga ut, och framåt kvällen fira! Om du får komplettering läser du noga vad som inte fungerar, fixar till det efter bästa förmåga, och skickar in igen.

Utloggning

När du är nöjd med ditt betyg (som står i studentklienten) kan du avsluta duggan. Stäng alla program och spara alla filer. Sedan loggar du ut som vanligt i menyn. Lämna inte din plats innan vanliga inloggningsskärmen syns.

¹Beroende på mängd samtidiga inskickningar kan det ta flera minuter att få svar, räck upp handen om du inte fått svar på 15 minuter.

Uppgift

I ett nytt äventyrsspel med 2D grafik ska en spelare kunna röra sig runt och interagera med objekt i spelet. I spelet ska det finnas en speciell typ av objekt vi kallar *modifiers*. Dessa objekt ska på ett eller annat sätt påverka egenskaper hos spelaren. Din uppgift är att skapa en prototyp till hur *modifiers* ska fungera. För att ha en spelare att påverka har du redan gjort klart en så enkel och minimal spelare som möjligt. Det är ett aggregat (en `struct`) med spelarens position, radie och historia. Spelaraggregatet (`struct Player`) finns givet i filen `gamedraft.cc`.

Alla *modifiers* har en position (x och y) och en storlek (radie). Med hjälp av dessa går det att avgöra om modifiern *överlappar* med spelarens position (dvs de kolliderar). Ett överlapp avgörs genom att undersöka om avståndet mellan objekten är mindre än summan av deras radie. Detta blir tydligt om du ritar en figur. Om vi benämner modifiern m och spelaren p så kolliderar spelaren med modifiern om $((p.x - m.x)^2 + (p.y - m.y)^2) < (p.r + m.r)^2$. Alla koordinater och avstånd hanteras som flyttal.

För att få till ett tillräckligt “proof of concept” ska du implementera två olika *modifiers*.

- Det ska finnas en *size-mod* som *påverkar* spelaren genom att addera en konstant (*delta*) till spelarens radie. Effekten blir att spelaren efter detta lättare kolliderar med andra modifiers (positiv konstant) eller tvärtom (negativ konstant). Denna modifier lägger även till strängen "Size adjusted by " + `to_string(delta)` + "`\n`" till spelarens historia (*story*). Konstanten ska anges som argument när ett *size-mod*-objekt skapas.
- Det ska finnas en *story-mod* som *påverkar* spelaren genom att addera en sträng till spelarens historia. Strängen ska anges som argument när ett *story-mod*-objekt skapas.

TIPS: I filen `gamedraft.cc` finns ett givet huvudprogram som i pseudokod skapar ett antal modifiers, placerar spelaren på ett antal olika positioner, på varje position undersöker vilka modifiers spelaren kolliderar med, och låter de modifiers som kolliderar med spelaren utföra sin påverkan. Körexemplet nedan visar vilken utdata detta huvudprogram ska ge i terminalen när det fungerar korrekt. Den givna koden ger ledning till lämplig funktionsindelning och namngivning. Notera att den givna koden kan behöva justeras för att erhålla dynamisk bindning.

Körexempel

```
=== Player One ===  
  
=== Player One ===  
Size adjusted by -3.000000  
  
=== Player One ===  
Size adjusted by -3.000000  
  
=== Player One ===  
Size adjusted by -3.000000  
Jumps on table  
Size adjusted by 5.000000  
  
=== Player One ===  
Size adjusted by -3.000000  
Jumps on table  
Size adjusted by 5.000000  
Bumped chair 4  
Bumped chair 5  
  
=== Player One ===  
Size adjusted by -3.000000  
Jumps on table  
Size adjusted by 5.000000  
Bumped chair 4  
Bumped chair 5  
Found treasure
```

Objektens placering och spelarens positioner grafiskt

Nedan figur bifogas enbart för att du ska få en bättre uppfattning om var den givna koden placerar de modifiers som skapas och vilka positioner spelaren besöker. Tanken är att det kan vara **till hjälp vid felsökning**. Det finns ingen kod för att rita ut figuren och det ingår *INTE* att skapa sådan kod (om du får tid över och vill skryta: varsågod!).

Figuren visar respektive modifiers placering grafiskt (ascii-grafik). Cirkeln med 1:or är den första size-modden i vektorn. Cirkelarna med siffrorna 2-7 är de 6 följande story-moddarna (kanske stolar). Centrala cirkeln med 8:or är en story-mod (kanske ett bord). Näst sist i vektorn finns en story-mod (kanske en skatt) som visas med tre 9:or i nedre högra hörnet av figuren. Sist i vektorn finns en size-mod som visas med kolon och ligger “över” bordet (8:orna).

Spelaren besöker de 6 positionerna märkta med två stjärnor (**) i ordning från övre vänstra diagonalt ned till nedre högra:

- På första positionen(4,4) händer ingenting.
- På andra positionen(12,14) kolliderar spelaren med modden(1) som minskar radien.
- På tredje positionen(14,17) händer ingenting eftersom radien nu är så liten.
- På fjärde positionen(20,25) påverkas spelaren av både bordet(8) och modden(:) som ökar radien.
- På femte positionen(27,36) påverkas spelaren av två stolar (4 och 5).
- På sjätte positionen(35,45) påverkas spelaren av modden som representerar en skatt.

