HÁZI FELADAT

Programozás Alapjai 2.

Végleges

Bertalan Áron

DO1H50

Feladat - Síkidomok

Készítsen absztrakt síkidom-osztályt, és valósítson meg segítségével szabályos háromszöget, négyzetet és kört! Ezen síkidomokat középpontjuk és egy csúcsuk (kör esetén a körvonal egy pontja) határozza meg, amelyek kétdimenziós koordinátákként olvashatóak be egy istream típusú objektumról. A síkidomoknak legyen olyan metódusa, amellyel eldönthető, hogy egy adott pont a síkidom területére esik-e!

Készítsen grafikus interface-t amivel változtatható a síkidomok pozíciója és mérete és orientációja, valamint lehet síkidomokat törölni és létrehozni. Az interface jelezze ha két síkidom átfedésben van.

Specifikáció

Alapvető működés

A feladatban 3 féle síkidomot kell tudni kezelni. Szabályos négyzetet, háromszöget és kört. A síkidomok tárolása a *Vector* nevű heterogén kollekció feladata, ennek tárolása pedig egy *Scene* objektumé. A *Scene* és a *Window* objektumok felelőssége továbbá a felhasználói interakciók kezelése (idomok mozgatása, létrehozása, stb.). A program a sík pontjait *Vec2* objektumként reprezentálja, mely egy kétdimenziós vektor.

Mentés és betöltés

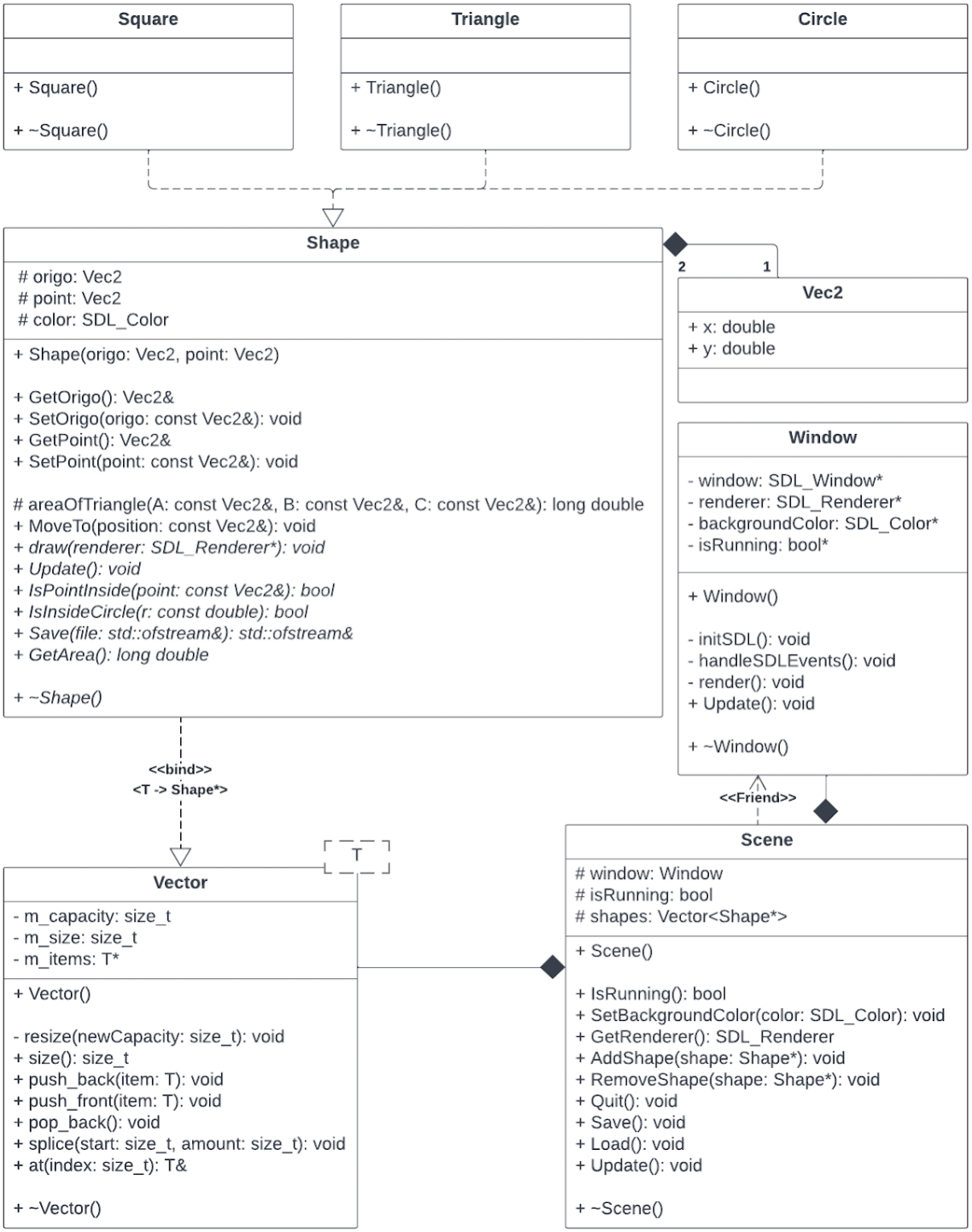
Az elkészített síkidomok egy *txt* formátumú fájlba lesznek menthetők. Az ide mentett objektumok formátuma a következő:

Minden sorban egy síkidom szerepel, és egy síkidom “*<típus> <középpont x koordinátája> <középpont y koordinátája> <csúcs x koordinátája> <csúcs y koordinátája>*”-ként van megadva. A típust egész reprezentálja: (1: négyzet, 2: háromszög, 3: kör)

Fontosabb algoritmusok

Minden idomnál szükséges megmondani, hogy egy adott pont a síkidomon belül helyezkedik e el. Ez a körnél nagyon egyszerű, itt csak Pitagorasz-tétellel eldöntjük, hogy a kör középpontjának és a pontnak a távolsága kisebb e a kör sugaránál. A háromszögnél és a

négyzetnél pedig ugyan azt az elvet lehet követni. Itt a sokszögek minden élére felveszünk egy háromszöget, melynek 3. csúcsa a megadott pont. Majd ha az így kapott háromszögek területének összege megegyezik az eredeti sokszög területével, a pont a síkidomon belül van, ellenkező esetben pedig kívül.



Tesztelés

Vec2, all
A kétdimenziós vektor műveleteit teszteli. Szimplán matek, nincs semmi különleges működés

Segment, all
Két-két merőleges és párhuzamos szakasszal kipróbálom, hogy megfelelően működik-e az `IsOverlappingWithSegment függvény.`

Shape, IsPointInside
Mindhárom alakzaton kipróbálom az `IsPointInside` függvényt egy olyan pontal ami az alakzaton belül, és egy olyannal ami az alakzaton kívül van.

Shape, IsOverlappingShape
Kipróbálom a függvényt az összes lehetséges kombinációval, valamint egy helytelen alakzattal, mely 2 szakaszt ad vissza a GetSegments függvényéből.

Shape, MoveTo
Tesztelem mind a háromfajta alakzat pozícióját elmozgatás előtt és után.

Shape, SetPoint
Kipróbálom, hogy mozgatható-e az alakzatokat definiáló csúcs a függvénnyel.

Shape, GetSegments
Kipróbálom, hogy az összes alakzat megfelelő számú szakaszt ad-e a GetSegments függvényével.

Shape, Save
Kipróbálom, hogy minden alakzatnak megfelelő-e a kimenete a Save függvényének.

Scene, Add_RemoveShape
Letesztelem, hogy hozzá tudok-e adni alakzatot egy Scene-hez és hogy utána ki tudom e törölni belőle.

Scene, Save_and_Load
Letesztelem, hogy le tudja-e menteni a Scene osztály a benne tárolt alakzatokat és, hogy megfelelő debug üzeneteket küld-e.

Vector, all
Letesztelem a Vector osztály összes függvényét.