|  |
| --- |
| ***Teme:***   * ***Uporaba obstoječih razredov:***   + ***uporabniški vmesnik, dogodki na el. uporabniškega vmesnika, obdelava dogodkov***   + ***sestavljanje objektov iz obstoječih objektov***   + ***osnovne operacije nad objekti v prostoru*** |

|  |
| --- |
| **Opis problema, izhodiščna navodila in viri:**  Cilj vaje je sestaviti 3D kocko, kot element poljubne namizne igre s kockami. Kocko bi želeli sestaviti tako, da bi jo v prostoru lahko premikali in vrteli (kar se dogaja pri metu kocke).  Ozadje:  JavaFX pozna znotraj Shape3D pred-definirano obliko Box (javafx.scene.shape.Box), s katero lahko predstavimo kocko. Vendar je problem v rabi tega objekta tem, da ne moremo kontrolirati/nastavljati/spreminjati vsake stranice +škatle+ posebej. Zato smo prisiljeni kocko sami (z)modelirati; sestaviti iz posameznih stranic. Stranice so pri tem dimenzijsko enake, razlikujejo se zgolj po svoji pojavni obliki (slika pik na stranici), ter poziciji v ravnini in orientaciji. Za postavitev moramo biti seznanjeni z osnovnimi operacijami nad objekti v prostoru: skaliranjem(povečevanjem,zmanjševanjem), translacijo(premik v izbrano smer) in rotacijo(orientacija,zasuk). Vse tri operacije so v javaFX na vseh objektih tipa Node realizirani.  Da nam bo modeliranje lažje in da je možen +in-time+ preskus z vizualizacijo postavitev posameznih objektov, za izvedbo vzamemo primer vizualizacije kemijske molekule vode, ki jo Oracle prijazno ponuja v osnovnem tečaju izgradnje 3D aplikacij : <http://docs.oracle.com/javafx/8/3d_graphics/MoleculeSampleApp.zip>, sept. 2024. Aplikacija omogoča 3D vizualizacijo prostorskih objektov, … , in spremenljiv pogled na objekt, kar bo dovolj, da izpeljemo približek vaje. |

**Naloga 1**

*Prenesemo MoleculeSampleApp, aplikacijo prevedemo in zaženemo.*

**Naloga 2**

*Analiza delovanja: izvedemo hitro analizo delovanja obstoječe aplikacije z vpogledom v izvodno kodo aplikacije:*

1. *vemo : JavaFX aplikacija se +prične+ v metodi start: pregledamo metodo in skušamo ugotoviti:*

*katera metoda zgradi molekulo ?*

*katera metoda zgradi osi koordinatnega sistema v prostoru?*

*Katera metoda postavi v opazovalca, ki opazuje objekt (kamero)?*

1. *Vemo: aplikacija je dogodkovno krmiljena-->odziva se na dogodke : kateri objekti prožijo dogodke. Izpolnimo tabele, vpišemo vrednosti za +tipke+ in njihove vloge. V pomoč naj vam bo izvleček iz obstoječe kode:*

*Scena – keypressed –--> tipka vloga tipke*

*Scena – mousePressed ---> tipka vloga tipke*

*Scena – mouseDragged --> tipka vloga tipke*

**Naloga 3**

Modifikacija kode: ohranimo prikaz zgolj komponent, ki jih potrebujemo: prostor, mehanizmi operacij nad objekti v prostoru, prikazane osi. Pri tem kode NE brišemo (mogoče jo bomo še potrebovali). Namesto brisanja bomo klice metod, ki jih ne potrebujemo enostavno komentirali.

1. Trajno vizualiziramo osi prostora, da ne bo potrebno vedno pritiskati tipke:

Pri pregledu kode vidimo, da je rezultat metode buildAxis() dejansko axisGroup(os X+osY+osZ), ki je dodana v 3D svet. Ker ta objekt ni definiran v sami metodi, preiščemo celotno kodo programa in pogledamo, če se axisGroup še kje manipulira.Pri iskanju najdemo pojavitev v metodi handleKeyboard, kjer s tipko kontroliramo prikaz/skrivanje osi (glej nalogo 2b). Vidimo lahko, da je pritisk na tipko enostaven preklop vidnosti. Ker začetna vidnost ni nikjer nastavljanjena, jo nastavimo ob koncu metode buildAxes (kako, je razvidno iz samega preklopa v handleKeyboard)

1. Onemogočimo klic metode buildMolecule() v metodi start aplikacije.

Dobiti bi morali svet, v katerem so trajno vizualizirane zgolj osi.

**Naloga 4**

V prostor postavimo 1 ploskev. Uporabili bomo enak mehanizem, kot je uporabljen v celotni aplikaciji (Zakaj? Preprosto zato, ker dana aplikacija dela /v nulo/). Ploskev bo shape.Rectangle (pravokotnik).

Analiziramo gradnjo molekule (metodo buildMolecule() ):

1. Za molekulo vode vemo (vidimo), da je sestavljena iz dveh atomov vodika (hydrogen) in atoma kisika (oxygen). Izberemo en sam objekt, npr. +oxygen+ in pogledamo kako je zgrajen in v katerih elementih še nastopa:
2. Vrstica 160: Sphere oxygenSphere = new Sphere(40.0); // naredi kroglico za kisik
3. Vrstica 186: oxygenXform.getChildren().add(oxygenSphere); // doda v Xform
4. Vrstica 183: moleculeXform.getChildren().add(oxygenXform); // se doda v molecule form
5. Vrstica 198: moleculeGroup.getChildren().add(moleculeXform); //se doda v molekulo
6. Vrstica 200: world.getChildren().addAll(moleculeGroup); //doda molekulo v world

Pogledamo še kaj so oxygenXform, moleculeXform in moleculeGroup:

1. Vrstica 154: Xform oxygenXform = new Xform();
2. Vrstica 153: Xform moleculeXform = new Xform();
3. Vrstica 68: final Xform moleculeGroup = new Xform();

Ko najdemo vse definicije in odvisnosti, lahko zgradimo na isti način svojo skupino za 6 stranic kocke, po receptu, ki je enak principu v ostoječi kodi:

1. Za vrstico 68 dodamo : final Xform kockaGroup = new Xform(); // očitno mora biti skupina globalno dostopna
2. Za metodo buildMolecule() dodamo metodo private void buildKocka() in vanjo dodamo zgolj kodo:

Xform kockaXform = new Xform();

Xform stranicaXform = new Xform();

Rectangle stranicaKocke = new Rectangle();

// določimo(sami) velikost 50x50 pik, rectangle bi moral biti avtomatično črn

stranicaXform.getChildren().add(stranicaKocke);

kockaXform.getChildren().add(stranicaXForm);

kockaGroup.getChildren().add(kockaXfrom); //sem noter gredo vse naslednje stranice

world.getChildren().add(kockaGroup);

1. Dodamo klic nove metode v metodo start za onemogočenim klicem buildMolecule()

Iz opazovanja:

* + 1. 2D objekt Rectangle je bil postavljen v izhodišče ravnice s prijemališčem lika (spodaj levo)
    2. Širi se zgolj v ravnini XY proti večjim X in Y, dimenzija v razsežnosti Z je 0

**Naloga 5**

Preskusimo rotacije in translacije:

Pri sestavljanju kocke bo očitno postopke sledeč: postavimo novo stranico/rectangle v prostor. Ta se bo vedno postavil v (0,0,0) prostora. Če bomo želeli stranico na +pravi+ poziciji, jo bo potrebno a) ustrezno zavrteti b) ustrezno premakniti. Težimo k temu, da bodo vse stranice postavljene tako, da bo središče sestavljenega objekta vedno v točki (0,0,0).

Oracle je bil prijazen in nam je pripravil razred Xform. Če ga malo pogledate, boste videli, da omogoča vse objekte tega tipa prestavljati in obračati.

Znotraj metode buildKocka() za objekt stranicaXform preverimo, kaj se zgodi v primeru:

stranicaXform.setTranslate(-25,-25,-25);

ali pa

stranicaXform.setRotate(90,0,0); // po osi x za 90 stopinj

Iz opazovanja:

Pri rotaciji se za prijemališče ne uporabi spodnji levi vogal objekta temveč +središče+ objekta

**Naloga 6**

Dopolnite metodo buildKocka() tako, da boste s-kreirali vseh 6 stranic in jih sestavili v kocko. Kocka mora biti sestaljena tako, da če celotno kockaGroup premaknete ali zasukate za poljubno razdaljo/kot mora po premiku oz. Rotaciji še vedno biti kocka (monoliten objekt).

**Naloga 7**

Namesto objekta tipa Rectangle uporabite ImageView (košček slike; demo imate v vaji 22) in poskusite sestaviti kocko iz slik (delov slik), da bo na vsaki stranici vsebovala od 1 do 6 pik.

**Naloga 8**

Poskusite animirati rotacijo (recimo: ob pritisku na tipko +presledek+ se začne kocka vrteti za izbrani kot okoli izbranih osi, ob ponovnem pritisku na +presledek+ se animacija ustavi. Mehanizem za proženje in zaustavljanje že imate v originalni kodi.