Projektovanje i arhitektura računarskih sistema

Radionica dizajn

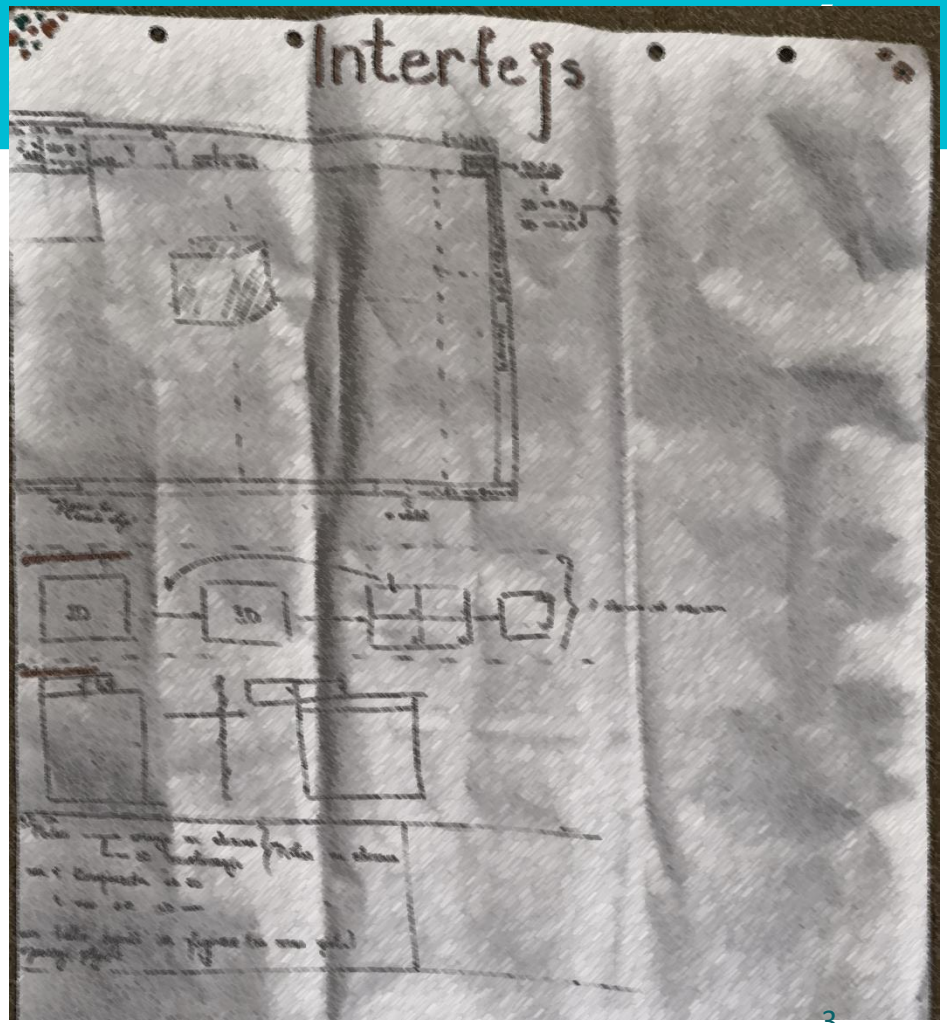
Odsek za računarsku tehniku i računarske komunikacije

Zadatak

- Da bi imali dovoljno vremena, smanjili smo obim zadatka na prikaz modela i manipulaciju sa modelom preko raznih transformacija
- Transformacije se uključuju u softver preko plugin mehanizma (može ih biti više ili manje)
- Zadatak smo podelili na 3 celine i 3 tima:
  - Team 1. Grafička sprega (interfejs)
  - Team 2. Plugin mehanizam
  - Team 3. Model podataka

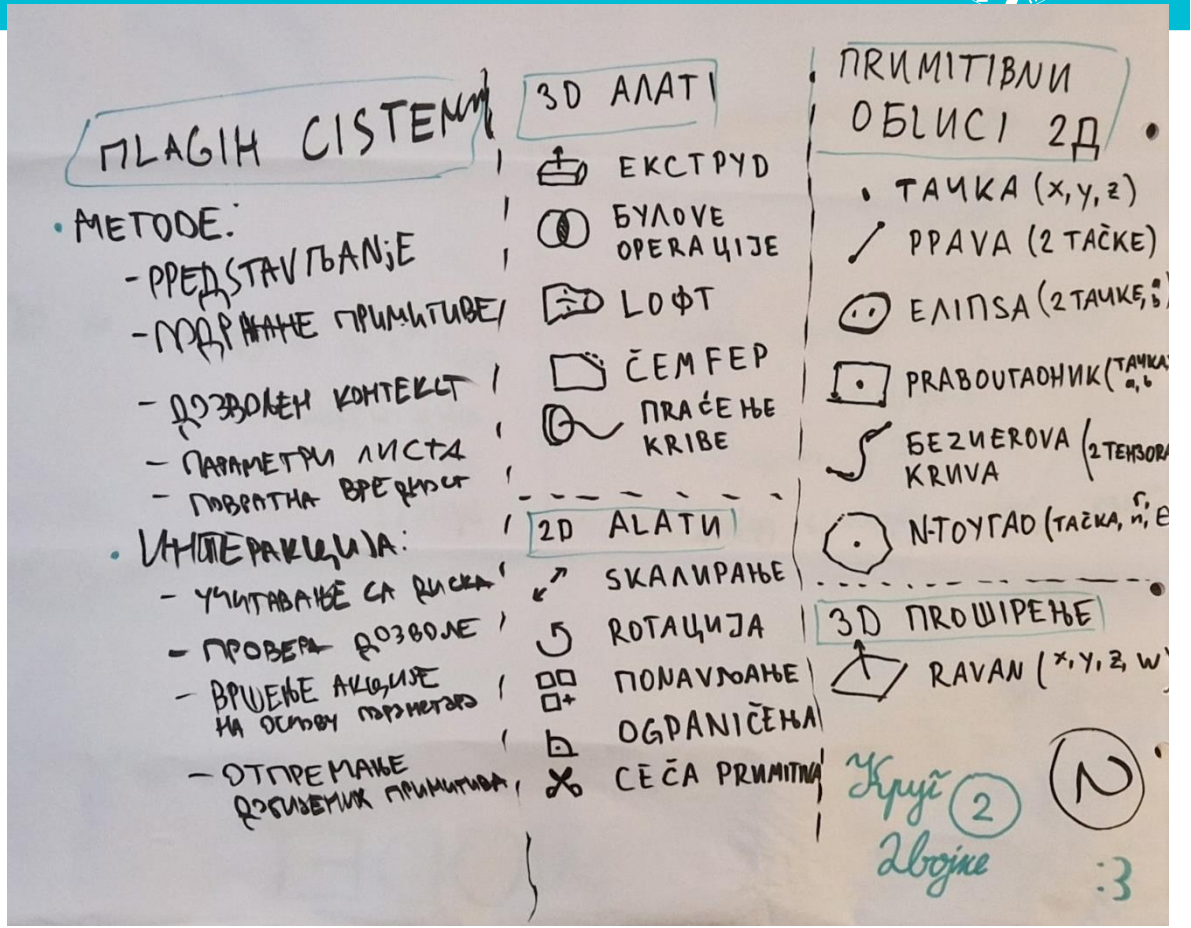
Team 1 - Interfejs

- Nismo imali isto razumevanje šta se očekuje
- Tim se fokusirao na vizuelni dizajn i tok upotrebe programa



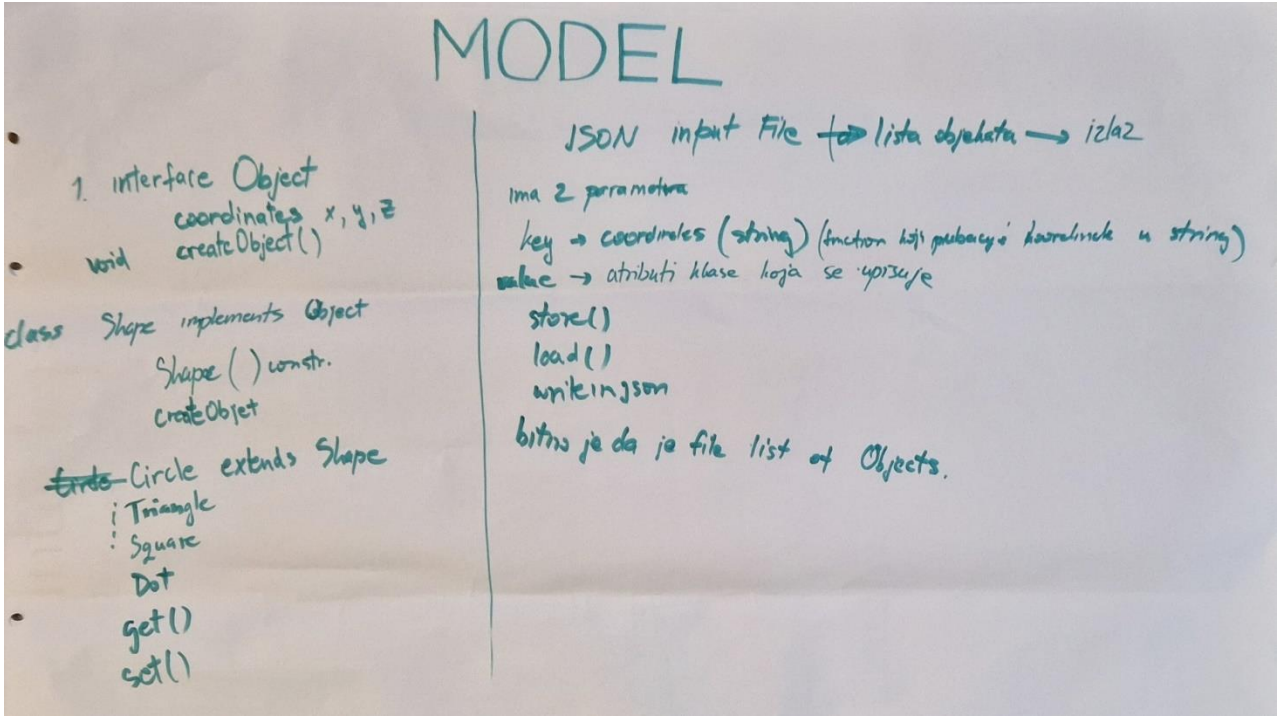
Team 2 - Transformacije

- Nismo imali isto razumevanje koji nivo detalja je potreban
- Opis ima premalo detalja da bi se dogovorili sa ostalim timovima ali i da se razreše dizajn dileme (na primer parametri pojedinih obrada)

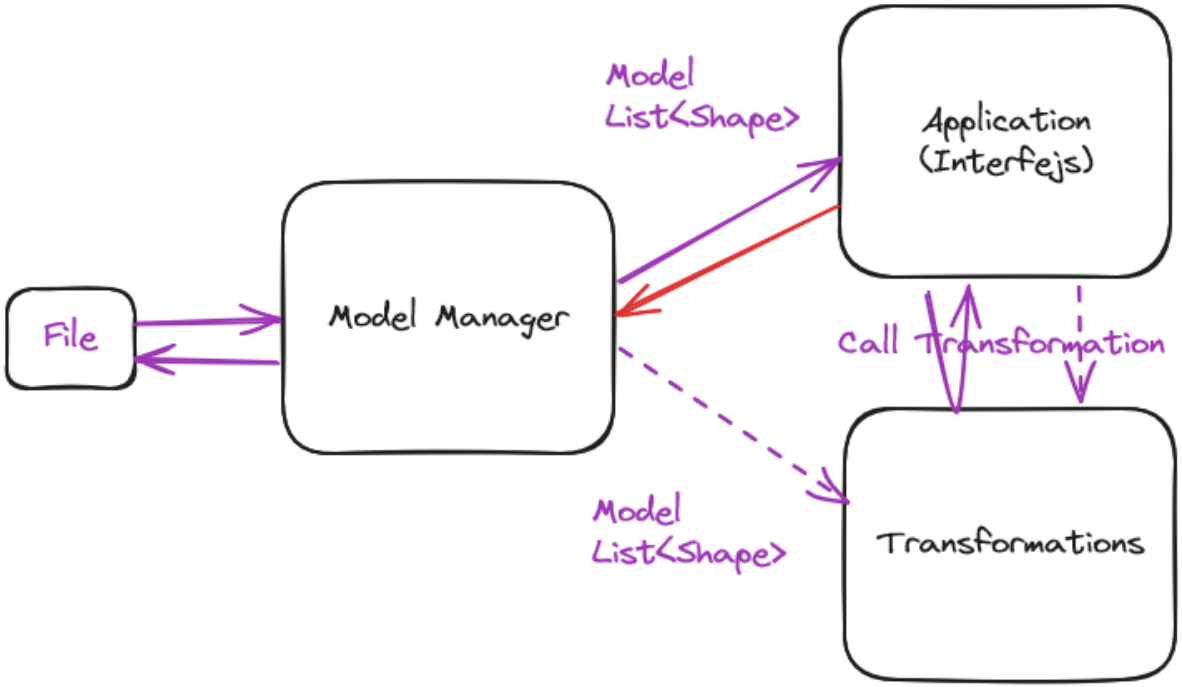


Team 3 – Model podataka

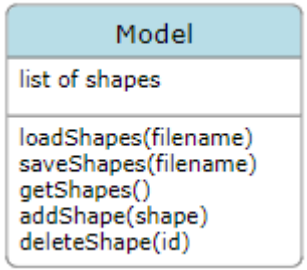
- Najviše smo diskutovali sa timom 3
- Pričali smo o nasleđivanju, hijerahiji klasa, polimorfizmu kao rešenju za snimanje objekata u fajl, pa čak i za iscrtavanje



Komponente softvera



Model

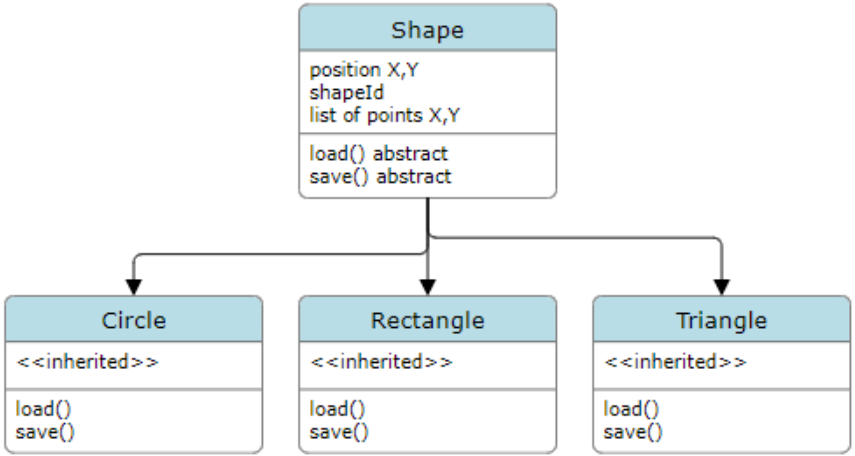



```
# Save to file

open(file);
writeHeader(file);
listOfShapes.forEach(shape) {
  // Polymorphism
  shape.saveTo(file);
}
writeFooter(file);
close(file);
```

```
# Load from file

open(file);
readHeader(file);
while(!eof(file)) {
    shapeStr = readShapeDescription(file);
    shape = ShapeFactory.create(shape);
    listOfShapes.add(shape);
}
readFooter(file);
close(file);
```

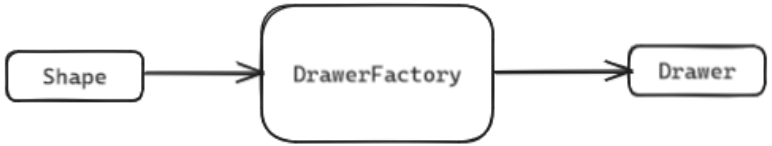


Interfejs

```
# Create drawers
listOfShapes.forEach(shape) {
  // Factory
  drawer = DrawerFactory.createDrawer(shape);
  listOfDrawers.add(drawer);
}

# Draw

listOfDrawers.forEach(drawer) {
  // Polymorphism
  drawer.draw(context);
}
```



Transformacije

- Plugin mehanizam
- Potrebno je definisati dve sprege
  - Ka pluginima
  - Ka korisniku
- Razmotriti
  - Životni vek pojedinih elemenata
  - Način upotrebe plugin
  - Tok podataka od korisnika do plugin i nazad

Zaključci

- Rešenje se ne vidi na prvi pogled
- Raspoloživo vreme stvara pritisak i brzo se potroši
- Dizajn treba da obezbedi dovoljno informacija za saradnju timova i implementaciju pojedinih delova
- Saradnja unosi dodatnu poteškoću
- Tokom opisa ideje rešenja (koncepta) koristiti konkretnije stvari
- Ne ići u detalje, ići samo do momenta da pohvatate šta je specifično a šta generalno
- Ukoliko nije zapisano, ne može se iterirati
- Svaka iteracija mora da doda ili razjasni neke detalje