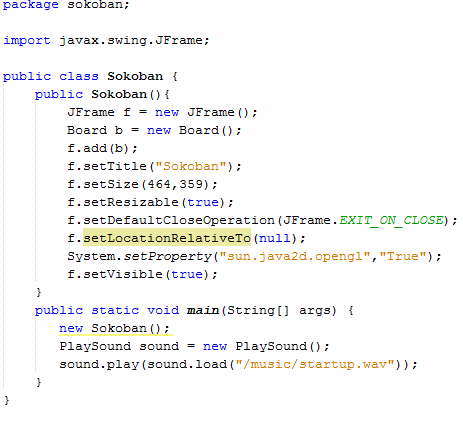
**Opis igrice:**

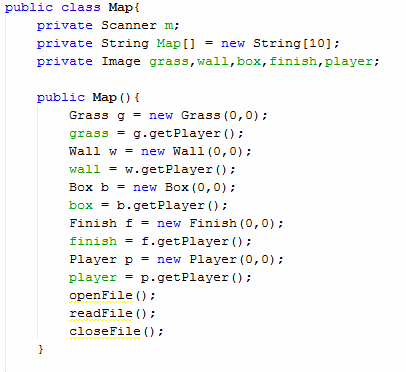
Igrica se zove Sokoban. Cilj ove igre je da igrač odgura sve kutije kroz lavirint na označene pozicije. Kontrole za upravljanje glavnog lika su strelice( gore, dole, levo, desno). U slučaju da odgurnete kutiju gde ne želite pretiskanjem slova “u” na tastaturi vraćate se za jedan korak unazad, a ako ste odgurnuli kutiju prilikom tog pomeranja vraća se I ona u prethodni položaj. Imate ograničeno vraćanje unazad na 3 puta, to jest kada se vratite treći put unazad igra se prekida, iscrtava se String za gotovu igru ispod glavnog lika I pusta melodija. Prilikom svakog pomeranja iscrtava se String – broj pomeraja, kao I broj vraćanja. Kada se sve kutije na cilju igra se prekida, iscrtava se String za pobedu I pusta melodija.

**Sokoban klasa:**

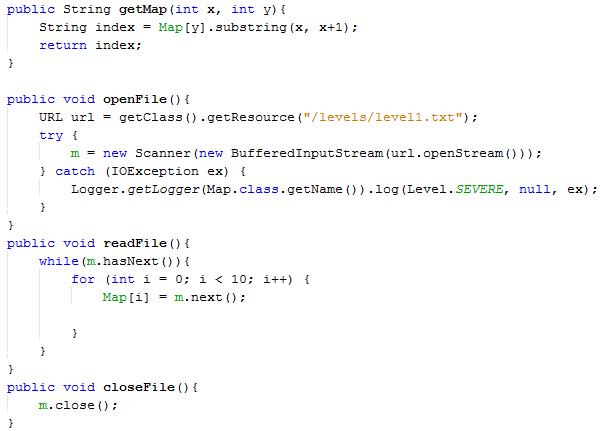


Ovo je main klasa. Ovde sam u konstruktoru Sokoban napravio objekat Board (glavna klasa koja vrši iscrtavanje I većinu ostalih funkcija) I dodao ga na frejm. Zatim postavio sam Naziv frejma, veličinu, da je resizable, da je uključen opengl – da bi hardverski ubrzao iscrtavanje slika, pošto ima dosta slika I iscrtavanja I na kraju da je vidljiv frejm. U main metodi pozivam konstruktor klase, kreiram objekat sound za pustanje “wav” melodija, zatim nad objektom sound pozivam metodu play koja kao argument uzima učitan fajl sa relativne putanje iz foldera /music/.

**Map klasa:**



Ovo je klasa Map koja kreira objekte koji su potrebni za iscrtavanje, Ovde su objekti samo kreirani sa kordinatama (0,0) koje će biti kasnije promenjene. Nad svakim objektom preko metode getPlayer dobijamo sliku tog objekta I smeštamo ih u promenljive tipa Image (grass, wall, box, finish, player). Nakon toga pozivamo metode openFile, readFile I closeFile. Nakon ovoga su geteri I seteri za promenljive.



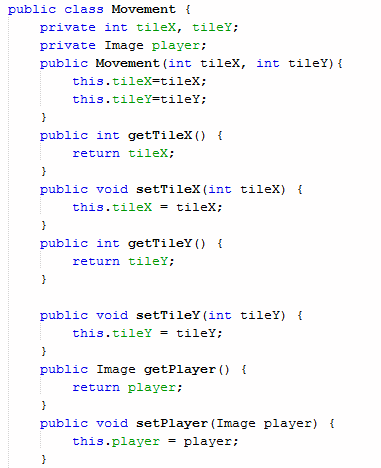
Metoda openFile preko url-a uzima relativnu adresu gde se nalazi txt fajl sa mapom nivoa /levels/level1.txt. U try bloku se kao argument za Scanner kreira novi Buffer za čitanje I otvara stream za kasnije čitanje fajla.

Metoda readFile čita fajl preko while loop-a sve dok ima sledećeg stringa. Pošto ima 10 redova u fajlu for loop ide do 10 I u niz Map[i] smešta ceo red.

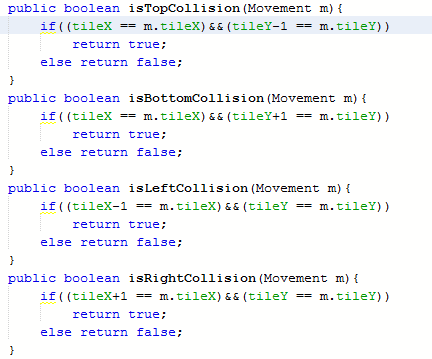
Metoda closeFile zatvara čitanje fajla.

Metoda getMap uzima kao argumente dva broja koja su koordinate karaktera koji se uzima iz mape. Map[y].sustring(x, x+1) Ovde se uzima red iz niza Map I onda se iz tog reda uzima String koji počinje u x koordinati a završava se u x+1.

**Movement klasa:**



Ovo je klasa Movement koja ima getere I setere za promenljive tileX, tileY koje predstavljaju polje gde se nalazi objekat a promenljiva player je slika tog objekta. Argumenti konstruktora ovde služe da se odredi polje prilikom inicijalizacije objekta.



Metoda isTopCollision služi za proveravanje sudaranja 2 objekta. Drugi objekat se uzima kao argument. Ako su coordinate x iste a koordinata y drugog objekta je za 1 veća onda ima gornjeg sudaranja.

Metoda isBottomCollision je ista kao I prethodna samo što se sada proverava sudaranje sa donje strane sa drugim objektom. A to je ispunjeno ako sui m x coordinate iste a y koordinata prvog objekta je za 1 manja od drugog.

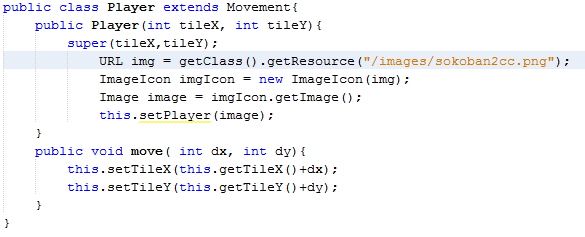
Metoda isLeftCollision proverava da li ima levog sudaranja, a to je ispunjeno kada su y coordinate jednake a x koordinata prvog objekta je veća za 1 od drugog.

Metoda isRightCollision vraća true ako su y koordinate 2 objekta iste a x prvog je za 1 veči od drugog.

Ove coordinate su coordinate polja, a coordinate za iscrtavanje se kasnije računaju prilikom iscrtavanja na osnovu polja u kome se nalazi objekat.

Ovu klasa je osnova za sledeće objekte koje ćemo kasnije da kreiramo.

**Player klasa:**



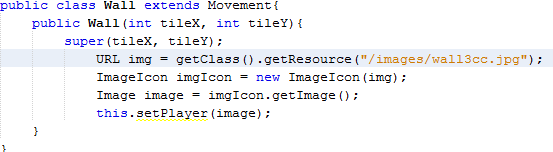
Ova klasa nasledjuje movement klasu da bi mogle da se setuju vrednosti objekta I da se koriste geteri, kao I za proveravanje sudaranja. Ovde koristimo nasledjivanje ne bi pisali sve ono što se nalazi u klasi movement, to jest za skraćivanje koda. Argumenti konstruktora su isti kao I u klasi koja se nasledjuje.

Preko URL-a se uzima slika sa relativne putanje koja se nalazi u folderu /images/. Zatim se koristi ImageIcon koji preuzima sliku kao icon I onda Image iz Icona uzima sliku koja se setuje na objekat.

Za slike ne koristimo BufferedImage da se program ne bi usporavao vremenom.

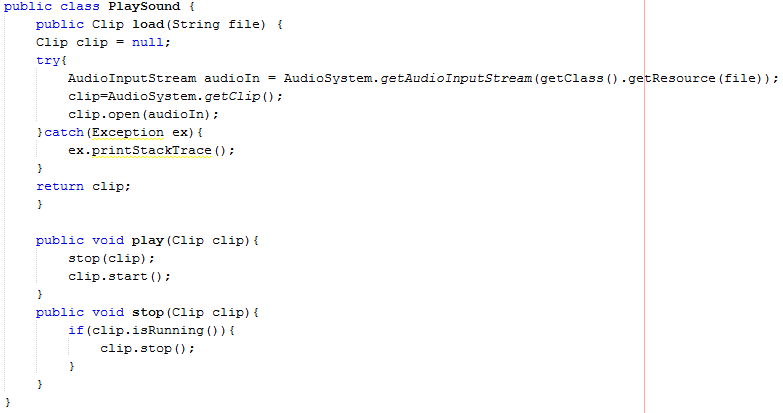
Metoda move kao argumente uzima dx I dy koji predstavljaju za koliko se pomera x I y polje. Preko this pristupamo metodama setTileX I setTileY koje se nalaze u klasi Movement ali pošto smo extendovali možemo da koristimo sve metode koje ima I klasa Movement. Vrednost se setuje na trenutnu (koju uzimamo preko metoda getTileX I getTileY) + dx ili dy.

Klasa Player I klasa Box su identične, razlikuju se samo zbog setovanja različitih slika na objekat.



Klase Wall, Finish I Grass nemaju metodu move jer se ne pomeraju već samo setuju sliku I kreiraju objekat koji se nalazi na odredjenom polju.

**PlaySound klasa:**

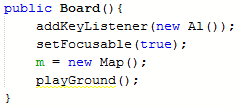


Ova klasa služi za pustanje melodija iz fajla. U metodi load se kao argument prosledjuje lokacija fajla sa relativne putanje kao String. Ta lokacija se prosledjuje u try bloku za učitavanje u AudioInputStream. Clip zatim uzima fajl koji je učitan. Ako fajl ne postoji catch blok hvata exception I štampa informacije o grešci iz steaka. Ova metoda vraća melodiju koja je učitana.

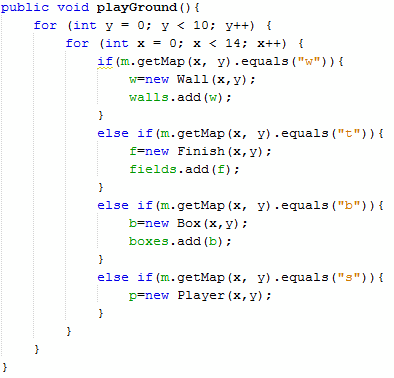
Metoda play služi za pustanje klipa koji se unosi kao argument a može I da se pozove medoda load kao argument za metodu play. U Ovom metodi se pusta klip.

Metoda stop zaustavlja klip.

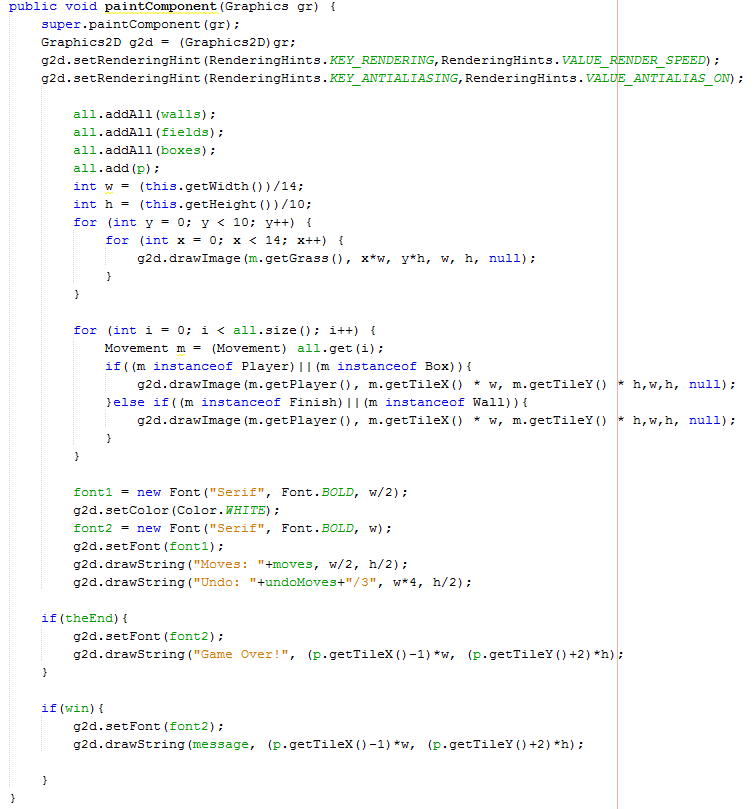
**Board klasa:**



Ova klasa ekstenduje JPanel na koji ćemo da iscrtavamo. U konstruktoru klase smo stavili addkeyListener – listener za dugmiće na tastaturi. Argument za ovaj listener je klasa Al koja na svaki keyPressed izvršava neku operaciju. SetFocusable je true da bi event sa tastature bio prosledjen komponenti koja ima focus a to je JPanel. Zatim se inicijalizuje mapa koja čita iz fajla raspored komponenti I kreira objekte sa neodredjenim koordinatama. Na kraju metoda Playground na osnovu rasporeda iz mape kreira nove objekte sa koordinatama polja na panelu.



Metoda playground ima 2 for loop-a koji idu do 10 – broj redova mape I do 14 – broj kolona mape. U if pitamo da li je karakter na poziciji x,y jednak karakteru koji označava odredjeni objekat. Ako jeste onda inicijalizujemo novi objekat I dodajemo ga u listu. A za Sokoban ne dodajemo u listu jer je on jedinstven.



Metoda paintComponent iscrtava komponente na panelu. Prvo kastujemo grafiku na 2D grafiku. Renderovanje je postavljeno na RENDER SPEED I ANTIALIAS je uključen. Ovde ima renderovanja pošto se slike menjaju veličinu u zavisnosti od veličine frejma.

U listu all dodajemo sve liste komponenti koje su kreirane. Svaka od tih listi ima komponente sa objektima, a svaki objekat ima druge coordinate polja.

Promenljiva w predstavlja dužinu polja koju računamo tako što dužinu frejma podelimo sa 14 – br. Kolona. A za širinu delimo sa 10 – br. Redova.

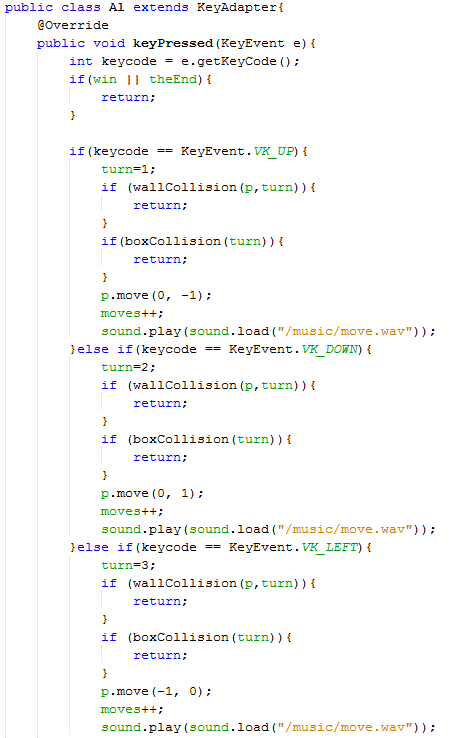
U prvom for loop-u iscrtavamo pozadinu, svako polje dobija je sada obojeno slikom za pozadinu. U drawImage prvo uzimamo sliku iz mape, zatim coordinate polja množimo sa w,h da bismo dobili prave coordinate za iscrtavanje, zatim stavljamo dužinu I širinu slike.

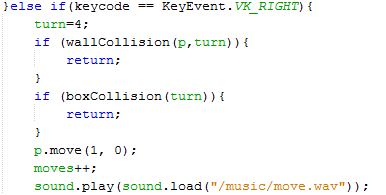
U drugom for loop-u kastujemo na objekat svaki element koji uzimamo iz liste koji je takodje objekat I iscrtavamo objekat u zavisnosti od njegove instance.

Fontovi su inicijalizovani tako da se njihova veličina I pozicija menjaju u zavisnosti od veličine frejma.

Crtanje Stringova na vrhu frejma koji predstavljaju pomeranja sokobana I pomeranja unazad.

Ako je theEnd true, a on je true kada je broj vraćanja unazad jednak 3, onda se iscrtava String ispod sokobana I igra se prekida. Ako je win true iscrtava se druga poruka I igra se prekida.

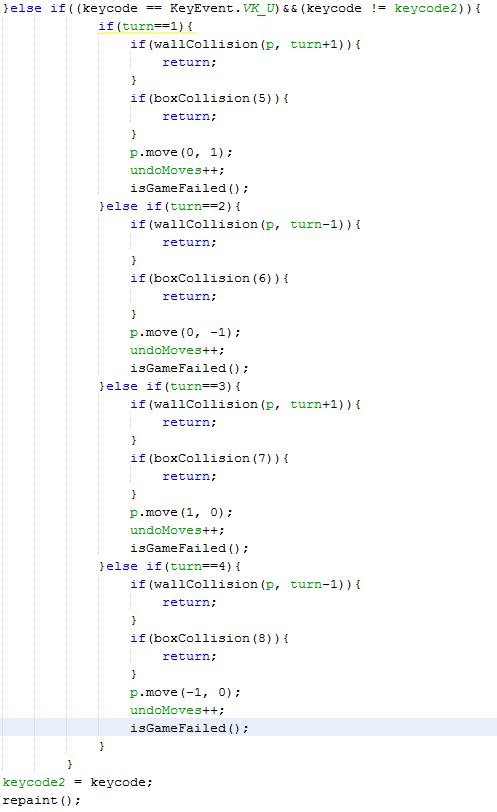




U ovoj klasi imamo metodu keyPressed koja u zavisnosti od pretisnutog dugmeta na tastaturi vrši pomeraj. Ako je win true ili ako je theEnd true igra se prekida zato što je ovaj uslov na početku I na svaki stisak dugmeta proverava se ovaj uslov ako nije ispunjen onda se izvršava kod koji sledi a ako je ispunjen onda se ništa ne dešava.

Ako je gornja strelica pretisnuta turn=1, prvi uslov je da li ima sudaranja sokobana sa zidom. Ako ima onda se ništa ne dešava ako nema onda pitamo da li ima sudaranja sokobana sa kutijom. Ova metoda vraća false kada nema sudaranja sa drugom kutijom ili kad nema sudaranja kutije sa zidom ,takodje ova metoda pomera kutiju. Kada je kutija pomerena boxCollision je false, Sokoban se onda pomera I broj pomeranja se inkrementuje za 1. Na svaki stisak strelice turn dobija novu vrednost I u zavisnosti od njega se preoverava Collision.

Nastavak ove metode…



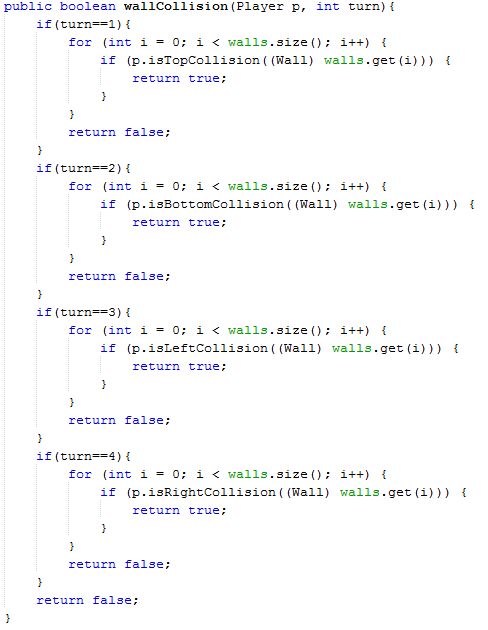
Na kraju ove metode keycode2 čuva vrednost keycode – a. A na početku metode keycode uzima vrednost pretisnutog dugmeta. Ako je pretisnuto slovo U I ako prošli stisak na taster nije isti onda se izvršava kod za vraćanje unazad. Ovim uslovom smo ograničili da Sokoban može da se vraća unazad samo jednom.

Sada u zavisnosti od turn - a proveravamo wallCollision ali sada menjamo vrednost turn – a pošto Sokoban ide u suprotan smer. Ovim smo iskoristili postojeću metodu da ne bismo pisali novu.

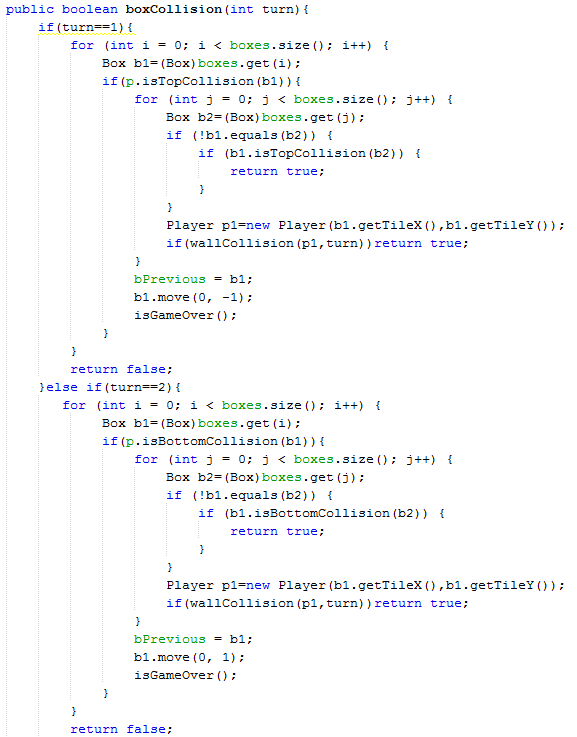
Za boxCollision imamo turn od 5 do 8 koji služi za unazad.

Ako ove 2 metode vraćaju false, kutija I Sokoban se pomeraju za jedno polje unazad, undoMoves se inkrementuje I poziva se metoda koja proverava da li je igra gotova.

Na kraju metode se iscrtavaju komponente.



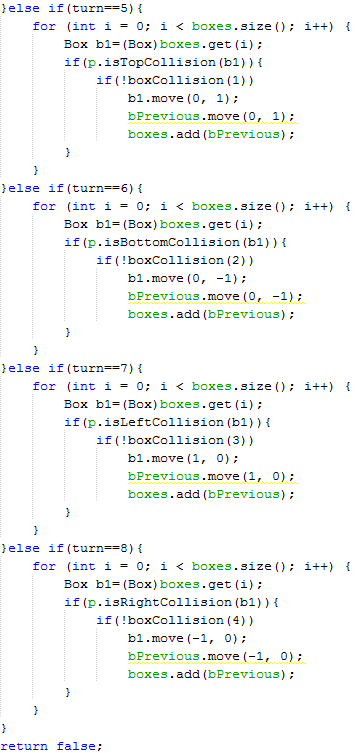
U metodi wallCollision proveravamo u zavisnosti od koordinata sokobana (ovde imamo kao argument objekat zato da bi iskoristili ovu metodu I za proveravanje sudaranja kutije sa zidom) I od vrednosti turn – a proverava se za svaki zid iz liste da li ima sudaranja sa sokobanom.



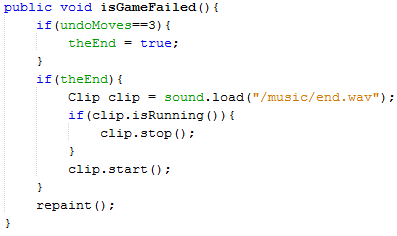
Metoda boxCollision uzima kao argument turn. Uzimamo svaki box iz liste I u zavisnosti od vrednosti turn – a proverava se collision (top, bottom, left, right) sokobana I kutije. Ako nema sudaranja vraća se false I ništa se ne dešava, a ako ima sudaranja onda proveravamo da li ima sudaranja kutije sa drugom kutijom tako što opet uzimamo svaku kutiju iz liste a ako one nisu iste proveravamo njihovo sudaranje.

A ako se Sokoban sudara sa kutijom I ona se ne sudara sa drugom kutijom onda postavljamo uslov da li se prva kutija sudara sa zidom, ako se ne sudara onda se vraća false, čuva se pozicija kutije u bPrevious za kasnije vraćanje unazad, kutija se pomera I proverava se da li je kraj igre.

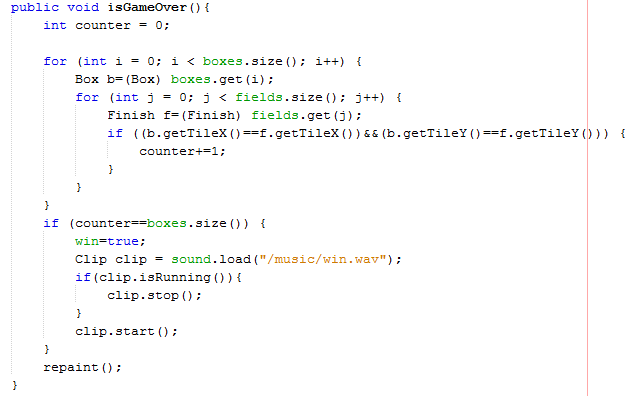
Za turn 3 I 4 je kod identičan.



Turn od 5 do 8 predstavljaju proveravanje sudaranja kutije za kretanje unazad. Za svaku kutiju iz liste proveravamo da li ima sudaranja sa sokobanom. Ako ima onda proveravamo da li je boxCollision za kretanje unapred false – a to znači da je kutija pomerena I da je Sokoban pomeren unapred. Ako je I ovaj uslov ispunjen kutija se pomera unazad to jest na coordinate bPrevious koje su prethodno sačuvane. I bPrevious se dodaje u listu kutija. Na kraju se vraća vrednost false.



Metoda isGameFailed proverava da li je broj vraćanja unazad jednak 3, ako jeste theEnd postaje true I igra se prekida jer je ovo uslov u metodi keyPressed za prekidanje kretanja. Takodje se iscrtava String u metodi PaintComponent. U ovoj metodi se nalazi pustanje klipa jer se ona izvršava samo jednom. Kada bi se klip pustao u paint metodi onda bi na svaki resize frejma klip pustao od početka.



Poslednja metoda u klasi Board je isGameOver koja proverava da li su sve kutije na odredjenim pozicijama. Za svaku kutiju proveravamo da li su joj coordinate jednake koordinatama finish polja, ako jesu povećavamo brojači ako je taj brojač jednak broju kutija to jest broju polja onda win postaje true, igra se prekida jer se kod za kretanje izvršava jedino kada je win false ili kada je theEnd false. Iscrtava se String ispod sokobana u zavisnosti gde se on nalazi na frejmu I pusta se wav klip.