[projekti nimi] dokumentatsioon

# Eessõna

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Sisukord

[Eessõna 1](#_Toc448305562)

[Mängumootor 4](#_Toc448305563)

[Sisenditöötlus 5](#_Toc448305564)

[Objektiloome 6](#_Toc448305565)

[Ruumide struktuur 7](#_Toc448305566)

[(package private) class Room 7](#_Toc448305567)

[Klassi Room väljad 7](#_Toc448305568)

[Klassi Room meetodid 7](#_Toc448305569)

[(package private) class ViewPoint 7](#_Toc448305570)

[Klassi ViewPoint väljad 7](#_Toc448305571)

[Klassi ViewPoint meetodid 7](#_Toc448305572)

[public abstract class Prop 7](#_Toc448305573)

[Klassi Prop väljad 7](#_Toc448305574)

[Klassi Prop meetodid 8](#_Toc448305575)

[public class StaticProp 8](#_Toc448305576)

[Klassi StaticProp väljad 8](#_Toc448305577)

[Klassi StaticProp meetodid 8](#_Toc448305578)

[public class DynamicProp 8](#_Toc448305579)

[Klassi DynamicProp väljad 8](#_Toc448305580)

[Klassi DynamicProp meetodid 9](#_Toc448305581)

[public class ContainerProp 9](#_Toc448305582)

[Klassi ContainerProp väljad 9](#_Toc448305583)

[Klassi ContainerProp meetodid 9](#_Toc448305584)

[(package private) abstract class Puzzle 10](#_Toc448305585)

[Klassi Puzzle väljad 10](#_Toc448305586)

[Klassi Puzzle meetodid 10](#_Toc448305587)

[(package private) class WindowedPuzzle 10](#_Toc448305588)

[Klassi WindowedPuzzle väljad 10](#_Toc448305589)

[Klassi WindowedPuzzle meetodid 10](#_Toc448305590)

[(package private) class InteractionPuzzle 10](#_Toc448305591)

[Klassi InteractionPuzzle väljad 10](#_Toc448305592)

[Klassi InteractionPuzzle meetodid 11](#_Toc448305593)

[Liidesed 11](#_Toc448305594)

[interface Clickable 11](#_Toc448305595)

[interface Switchable 11](#_Toc448305596)

[Seisundid 12](#_Toc448305597)

[Ideed 13](#_Toc448305598)

[Interpretaator 13](#_Toc448305599)

# Mängumootor

# Sisenditöötlus

# Objektiloome

# Ruumide struktuur

Kõik klassid ja liidesed, mis puudutavad ruumide struktuurilist ülesehitust ning ruumikomponentide omavahelist suhtlust, paiknevad paketis RoomStructure.

## (package private) class Room

Klass, mis esinab ruumi kui tervikut. Väljade seas ei ole visuaalseid elemente, sest nad defineeritakse alles alamobjektide (ViewPoint, Prop) sees.

### Klassi Room väljad

private int ID – täisarvuline identiteedinumber, mis lihtsustab ruumile viitamist massiivides ja sõnastikes.

private ViewPoint[4] viewPoints – neljaliikmeline massiiv ruumi vaatenurkadest.

### Klassi Room meetodid

public void Load() – jookva ruumi laadimine (sisseloetud andmete realiseerimine)

public void nextViewPoint(int direction) – vaatenurga vahetus. Parameetri 1 korral pööratakse paremale, -1 korral vasakule.

## (package private) class ViewPoint

Esindab üht ruumi neljast vaatenurgast, sisaldab sõnastikku nähtaval olevatest esemetest (Prop). Ei saa eksisteerida ilma klassi Room isendita (kompositsioon).

### Klassi ViewPoint väljad

private HashMap<int, Prop> members – täisarvuline võti on eseme ID väärtus, Prop tüüpi on vaatepunktist nähtavad esemed. Prop-esemed eksisteerivad ka väljaspool antud vaadet (agregatsioon).

private int subID – antud vaate indeks klassi Room massiivis viewPoints. Et ViewPoint isendi olemasolu sõltub Room isendi omast, siis täpsemat identiteediarvu pole tarvis.

### Klassi ViewPoint meetodid

public void Load() – taustapildi ning kõikide Prop tüüpi esemete piltide kuvamine

## public abstract class Prop

Abstraktne klass, mis annab kõikidele ruumides paiknevatele esemetele üldise tüübi. Pärib JavaFX klassilt Node ja realiseerib liidest Clickable.

### Klassi Prop väljad

private int ID – eseme identiteediarv

private Puzzle puzzle – Ülesanne või pusle, mille alla antud element kuulub. Võib olla ka null.

**private Puzzle condition** – Ülesanne või pusle, mis peab olema lahendatud enne kui antud eset kasutada saab. Võib olla ka null.

**(väli kustutatud!)**

### Klassi Prop meetodid

public void Load() – eseme pildifaili laadimine ja paigutamine taustale

**(meetod kustutatud!)**

## public class StaticProp

Klass, mis pärib klassilt Prop ning esindab neid esemeid, mille asukoht ei muutu, kuid võib muutuda mingi osa (nt kaminatuli põleb/ei põle, aga kamin ise ei liigu). Realiseerib liidest Switchable.

### Klassi StaticProp väljad

private boolean stateLocked – tõeväärtus, mis näitab, kas eseme olekut tohib antud hetkel muuta.

**private ContainerProp container (lisatud!)** – konteiner, mille alla antud ese kuulub. Võib olla ka null.

private StaticState currentState – eseme jooksev olek. Sellele väljale seatakse väärtuseks üks järgnevatest:

public StaticActiveState activeState – State objekt, mis esindab eseme aktiivset olekut

public StaticIdleState idleState – State objekt, mis esindab eseme passiivset olekut

### Klassi StaticProp meetodid

public void lockState() – meetod, mis lukustab eseme seisundi.

public void unlockState() – meetod, mis teeb eseme seisundi (uuesti) muudetavaks.

## public class DynamicProp

Klass, mis pärib klassilt Prop ning esindab neid esemeid, mille asukohta saab muuta. Antud tüübiga märgistatud objekte on võimalik üles võtta (paigutada inventorysse) ning osadel juhtudel ka asetada muude kohta mängustseenil. Realiseerib liidest Switchable.

### Klassi DynamicProp väljad

private boolean stateLocked – tõeväärtus, mis näitab, kas eseme olekut tohib antud hetkel muuta.

**private ContainerProp container (lisatud!)** – konteiner, millele ese kuulub. Võib olla ka null.

private StaticState currentState – eseme jooksev olek. Sellele väljale seatakse väärtuseks üks järgnevatest:

public DynamicInventoryState inventoryState – State objekt, mis esindab eseme paiknemist inventorys.

public DynamicDragState dragState – State objekt, mis esindab eseme liikumisolekut (kasutaja poolt hiirega lohistamisel)

public DynamicIdleState idleState – State objekt, mis esindab eseme paigalolekut stseenil.

**private HashMap<int, int[5]> possibleLocations** – sõnastik, mis sisaldab selle eseme kõiki võimalikke asukohti. Võtmeks on ruumi ID, väärtuseks massiiv viiest täisarvust: vaatepunkti ID, x-koordinaat taustapildil, y-koordinaat taustapildil, eseme pildi laius antud asukohas, eseme pildi pikkus antud asukohas. **Võimalik, et tuleb võtme ja väärtuse rollid vahetada, kaasata ruumi ID massiivi ning määrata uueks väärtuseks pildifaili (sest vari jms on eri asukohtades erinev; alternatiiviks on nähtamatute konteinerite kasutamine placeholderitena).**

### Klassi DynamicProp meetodid

public void lockState() – meetod, mis lukustab eseme seisundi.

public void unlockState() – meetod, mis teeb eseme seisundi (uuesti) muudetavaks.

**public boolean isLocationValid(int room, int view, int x, int y, …)** – meetod, mis tagastab true siis, kui parameetritena antud asukoht on olemas sõnastikus possibleLocations.

## public class ContainerProp

Klass, mis pärib klassilt Prop ning esindab neid esemeid, mille sees (või küljes) paiknevad teised Prop tüüpi esemed. ContainerProp käitub sarnaselt StaticProp isenditele, tema asukoht on muutumatu, kuid saab muuta teatud osade olekut (kuid mitte konteineri kui terviku olekut).

### Klassi ContainerProp väljad

**Kustutada int ID!**

**private ArrayList<Prop>** subProps – loend, mis sisaldab konteineri elemente.

### Klassi ContainerProp meetodid

public void Add(Prop newProp) – elementide lisamise meetod

**public void Remove(Prop oldProp)** – elementide eelmaldamise meetod

public @Override void React() – Ülemklassi liidesest saadud meetodi ülekate, sisuliselt suunatakse klikireaktsioon ümber konteinerilt selle elemendile.

## (package private) abstract class Puzzle

Abstraktne klass, mis defineerib mängu ülesannete/mõistatuste üldise tüübi ning konkreetsed isendiväljad ja –meetodid.

### Klassi Puzzle väljad

private boolean[] checkpoints – tõeväärtuste massiiv, mis sümboliseerib ülesande täitsmise progressi. Iga massiivi element vastab ühele ülesande etapile ning selle väärtus on vaikimisi false. Väärtus muudetakse true’ks, kui vastav etapp saab täidetud.

### Klassi Puzzle meetodid

public void completeCheckpoint(int **index**) – meetod, mille kaudu muudetakse massiivi checkpoints elementide tõeväärtust true’ks.

public boolean isSolved() – meetod, mis tagastab true ainult siis, kui kõik massiivi checkpoints elemendid on true.

## (package private) class WindowedPuzzle

Klass, mis pärib klassilt Puzzle ning esindab ülesandeid, mille lahendamine toimub hüpikaknas. Selle klassi massiivi checkpoints pikkus on alati 1.

### Klassi WindowedPuzzle väljad

private Group(?) frame – JavaFX grupp, mis sisaldab antud pusle elemente.

**private Node[] elements (kustutada, sest pole vaja. Grupil jne on olemas .getChildren() )**

### Klassi WindowedPuzzle meetodid

public void makeaMove(int elementIdnex) – meetod, mille kaudu saab elemente hüpikaknas liigutada. Täisarv elementIndex on liigutatava elemendi indeks frame’i alamate loendis (children).

private boolean PuzzleCompleted() – Tagastab true, kui pusle on lahendusele vastavas olekus. See meetod kutsutakse välja ülemklassis defineerutud isSolved() meetodis.

## (package private) class InteractionPuzzle

Klass, mis pärib klassilt Puzzle ning esindab ülesandeid, mille lahendamiseks tuleb sooritada mitu tegevust erinevates kohtades (esemeid ümber tõsta).

### Klassi InteractionPuzzle väljad

private HashMap<int, int[2]> checkPointDestinations – sõnastik, mis näitab, milliseid esemeid ja kuhu on vaja tõsta, et ülesanne loetaks lõpetatuks. Võtmeks on eseme identiteediarv ning väärtuseks ruumi ja vaatepunkti identiteediarve sisaldav massiiv. Täpsem asukoht on kirjas vastava eseme isendiväljas possibleLocations.

### Klassi InteractionPuzzle meetodid

public void locationChanged(Prop checkpoint) – meetod, mis teavitab ülesande alla kuuluva elemendi asukohamuutusest. Kui uus asukoht on kirjeldatud sõnastikus checkpointDestinations, siis loetakse *vastav samm* täidetuks.

## Liidesed

Paketis RoomStructure on (hetkel) defineeritud järgnevad liidesed:

### interface Clickable

Võimaldab täpsustada vastust hiireklikile. Koosneb abstraktsest parameetriteta void meetodist React(), millelt eeldatakse klassisisese reaktsiooni implementeerimist.

### interface Switchable

Lubab vahetada seisundit mingi teatud sündmuse korral. Koosneb abstraktsest void meetodist enterState(State newState), mis seab currentState uueks väärtuseks parameetri newState.

# Seisundid

# Ideed

## Interpretaator

Kui ruumides paiknevad objektid tulevad liiga spetsiifilised, siis on võimalik neile defineerida üldine meetodite „sõnaraamat“ eraldi klassi(de)na, kus on üks suunav meetod, mis võtab vastu meetodi ID-numbri (ja tulevased parameetrid, nt DynamicProp tüüpi objekti) ning määrab switchiga, mida iga numbri korral teha (tegevused võivad olla eraldi meetodid). Kui sõnastikke läheb vaja rohkem kui üks, on neid võimalik rühmitada vastavalt ruumile või puslele.