بسم لله الرحمن الرحیم

پکیج CrossWalk :

توضیحات :

پکیج اصلی شامل کلاس اصلی و متد main

کلاس ها : شامل یک کلاس

StartGame.java :‌ کلاس اجرا کننده ی برنامه که شامل متد main است .

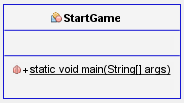
-ارث بری ها : بدون ارث بری

-فیلدها : ندارد

-متدها :

- متد استاتیک main برای شروع برنامه .

-دیاگرامUml :‌ (رسم شده توسط افزونه ی easy Uml برای netBeans ) :



پکیج CrossWalk.UI :

توضیحات :

پکیج جهت نمایش خروجی به کاربر و رابط کاربری

کلاس ها : شامل ۳ کلاس

GameListener.java‌ : کلاس نگهدارنده ی بازخورد ها به رویداد های ورودی مثلا کلیک موس و وارد کردن کلید کیبورد

-ارث بری ها : بدون ارث بری !

-فیلدها :

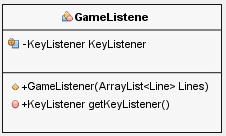
- private final KeyListener KeyListener : نگهداری پاسخ به بازخورد های صفحه کلید

-متدها :

public GameListener(ArrayList<Line> Lines)- : سازنده ی کلاس با ورودی یک مراجعه به لاین های بازی جهت ذخیره در فایل و بازیابی در هنگام ادامه ی بازی

-public KeyListener getKeyListener() : متد کوئری با خروجی فیلد KeyListener

-دیاگرامUml :‌ (رسم شده توسط افزونه ی easy Uml برای netBeans ) :



InitGame.java : کلاس تنظیم کننده ی نوع بازی مثلا بازی جدید ، مشاهده ی فیلم ،‌ حرکت خودکار ادمک ، بازی ذخیره شده

-ارث بری ها : ندارد

-فیلدها :

public static boolean GameStop- : فیلد استاتیک booleanبرای بررسی وضعیت توقف بازی

-public static boolean GameEnd : فیلد استاتیک booleanبرای بررسی وضعیت اتمام بازی

-متدها :

- InitGame() : سازنده ی پیشفرض بازی.

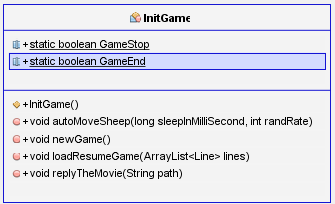
-autoMoveSheep() : پیاده سازی بازی براساس حرکت خودکار ادمک

-newGame() : پیاده سازی بازی جدید

-loadResumeGame() : پیاده سازی بازی براساس بازی قبلی و ادامه ی ان

- replyTheMovie() :‌ پیاده سازی فیلم بازی انجام شده.

-دیاگرامUml :‌ (رسم شده توسط افزونه ی easy Uml برای netBeans ) :



InitGraphic.java : کلاس تنظیم کننده ی گرافیک بازی و بخش UI

-ارث بری ها :

- ارث بری از کلاس JPanel برای پیاده سازی گرافیک با استفاده از swing

-پیاده سازی اینترفیس Runnable برای پیاده سازی thread

-فیلدها :

private ArrayList<Line> Lines- : نگهدارنده ی ارجاع به لاین های بازی

-private final int MiddleOfCrosswalkPosition‌ :‌ نگهدارنده ی مختصات وسط خط کشی عابر پیاده

-private BufferedImage CrosswalkImage : نگهدانده ی عکس خط کشی عابرپیاده

-private BufferedImage LineImage : نگهدارنده ی عکس لاین ها

-private BufferedImage MiddleLineImage :‌نگهدارنده ی عکس خط بین لاین ها

-private final int TopLineCount : تعداد لاین های بالایی

-public static Sheep Sheep = new Sheep() :‌ فیلد استاتیک برای نگهداری شی از ادمک

-private BufferedImage HeartImage :‌ نگهدارنده ی عکس قلب برای نمایش life ادمک

- private JFrame GameFrame : فریم اصلی بازی پیاده سازی رابطه ی Has a

-متدها :

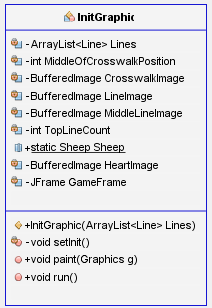
- public InitGraphic(ArrayList<Line> Lines) : سازنده با ورودی مراجعه به لاین ها

-private void setInit() :‌ تنظیم کردن پنل نمایشی بازی مانند title,size و ...

-public void paintComponent(Graphics g) : متد override شده از کلاس پدر جهت نمایش شی ها

-public void run() : پیاده سازی runnable جهت ایجاد یک ترد برای حرکت دراوردن اشیا

-دیاگرامUml :‌ (رسم شده توسط افزونه ی easy Uml برای netBeans ) :



پکیج CrossWalk.StoreData :

توضیحات :

پکیج ذخیره سازی اطلاعات روی فایل

کلاس ها : شامل ۲ کلاس

ResumeAndLoad.java :‌ کلاس ذخیره ساز داده های برای بخش ادامه ی بازی قبلی

-ارث بری: ندارد

فیلد : ندارد

متد ها :

-public ResumeAndLoad() :‌ سازنده ی پیشفرض

- public final void initForSaveGame(): تنظیمات اولیه برای ذخیره سازی مثلا ساخت فولدر Resume درصورت عدم وجود

- public boolean isExistResumeFile() :‌ بررسی اینکه ایا پوشه ی Resume موجود است؟

- public ArrayList<Line> loadGame() : برگرداندن لاین های ساخته شده از فایل موجود و مقدار دهی به فیلد های ادمک

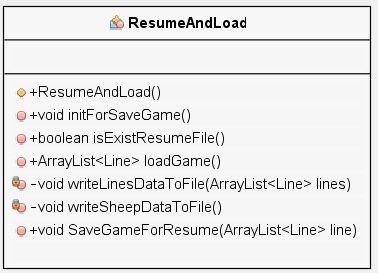
- public void saveGame(ArrayList<Line> lines) :‌ ذخیره کردن لاین ها که از ورودی میگیرید در فایل

- private void writeLinesDataToFile(ArrayList<Line> lines) : نوشتن اطلاعات لاین ها در فایل

- private void writeSheepDataToFile() :‌ نوشتن اطلاعات ادمک در فایل ها

-public void SaveGameForResume(ArrayList<Line> line) : سرویس دهنده به کلاس استفاده کننده و استفاده از دو متد بالا

-دیاگرامUml :‌ (رسم شده توسط افزونه ی easy Uml برای netBeans ) :



WriteReplyData.java :‌ کلاس ذخیره ساز داده های مورد نیاز جهت نمایش فیلم بازی انجام شده

-ارث بری: ندارد

-فیلد ها‌:

-private LocalDateTime StartedTime : نگهدارنده ی زمان جاری برای تنظیمات مربوط به حرکت خودکار ادمک

-‌ private String Path : نگهدارنده ی ادرس فولدری که قرار است داده ها درون ان ریخته شوند.

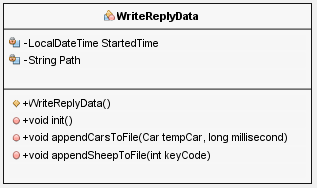
-متد ها :

- public final void init() : ساختن ادرس با استفاده از زمان جهت یکتا شدن ان و ساخت فایل تنظیمات جهت نگهداری تنظیمات بازی

- public void appendCarsToFile(Car tempCar, long millisecond) : اضافه کردن اطلاعات ماشین ها به فایل و مدت زمانی که ساخت ان طول کشیده است

public void appendSheepToFile(int keyCode)- : اضافه کردن اطلاعات حرکت ادمک به فایل

-دیاگرامUml :‌ (رسم شده توسط افزونه ی easy Uml برای netBeans ) :



پکیج CrossWalk.Utilities :

توضیحات :

پکیج شامل کلاس های مفید و استفاده شده در بازی

کلاس ها : شامل ۲ کلاس

ExceptionWriter.java :‌ کلاس ذخیره خطاهای رخ داده در برنامه و لاگ گرفتن از انها

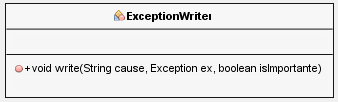
-ارث بری : ندارد

-فیلد‌: ندارد

-متد ها:

- public void write(String cause, Exception ex, boolean isImportante) : متدی برای لاگ گرفتن از خطا ها و نوشتن خطاها در فایل

-دیاگرامUml :‌ (رسم شده توسط افزونه ی easy Uml برای netBeans ) :



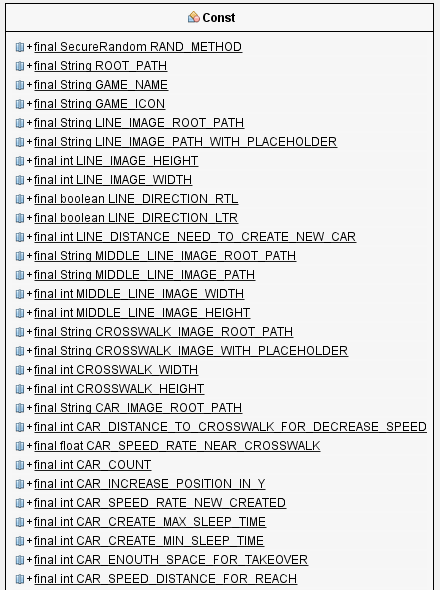
Const.java : کلاس نگهدارنده ی تنظیمات ثابت بازی

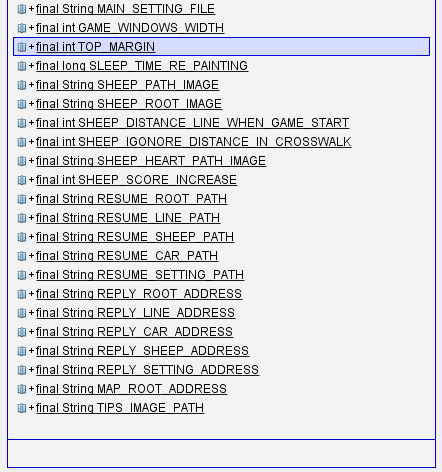
- ارث بری : ندارد

- فیلد‌: شامل تعداد زیادی از ثابت های بازی که درطول برنامه همیشه ثابت هستند مثلا مقدار کم شدن سرعت هنگام رسیدن به خط عابر پیاده ویا ادرس پوشه ی عکسهای بازی

- متد ها :‌ ندارد

-دیاگرامUml :‌ (رسم شده توسط افزونه ی easy Uml برای netBeans ) :





پکیج CrossWalk.AutoWork:

توضیحات :

پکیج شامل کلاس هایی که به صورت خودکار اقدام به انجام کار ها میکنندو اغلب یک thread پیاده سازی میکنند

کلاس ها : شامل ۵ کلاس

CreateCar.java :‌ کلاس abstract جهت ساخت ماشین های بازی و پیاده سازی اطلاعات لاین ها

-ارث بری :

-اینترفیس Runnable جهت پیاده سازی Thread

-فیلدها:

- private ArrayList<Line> Lines: نگهدارنده ی یک مراجعه به لاین های بازی

-private final int RtlLineCount :‌ نگهدارنده ی تعداد لاین های راست به چپ

-private final int LtrLineCount : نگهدارنده ی تعداد لاین های چپ به راست

-متد ها:

-public CreateCar() :‌ سازنده ی پیشفرض برای مقدار دهی اولیه ی تمام فیلدها

-public void InitLine(): متد سازنده ی لاین های بازی از فایل تنظیمات

-public void InitLineWithCars(ArrayList<Line> line) : متد سازنده ی لاین ها و اضافه کننده ی ماشین ها به لاین ها درزمان ادامه ی بازی

-public ArrayList<Line> getLine() : متد کوئری با خروجی لاین ها

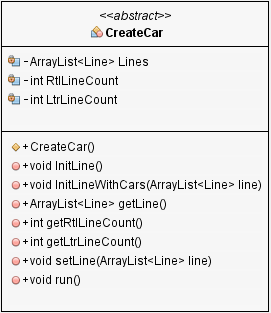
-public int getRtlLineCount() : متد کوئری با خروجی تعداد لاین های راست به چپ

-public int getLtrLineCount() : متد کوئری با خروجی تعداد لاین های چپ به راست.

- public final void setLine(ArrayList<Line> line) : متد set کننده ی فیلد لاین

- public abstract void run(): پیاده سازی متد اینترفیس جهت ایجاد Thead و سپردن بدنه ی تابع به فرزندان با کلمه ی کلیدی abstract

-دیاگرامUml :‌ (رسم شده توسط افزونه ی easy Uml برای netBeans ) :



CreateCarInNewGame.java : ساخت ماشین ها به صورت رندوم در بازی جدید

-ارث بری :

-ارث بری از کلاس abstract با نام CreateCar

-پیاده سازی اینترفیس Runnable که پیاده سازی ان از پدر به فزرند سپرده شده است

-فیلد ها :

- public boolean SaveReply : نگهدارنده ی این که ایا داده ها برای فیلم ذخیره شوند؟

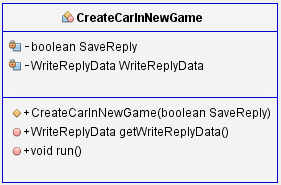
- public WriteReplyData WriteReplyData :‌ نگهداری کلاسی که جهت نوشتن اطلاعات برای فیلم بازی نیاز است در واقع پیاده سازی رابطه ی Has a

-متدها :

- public CreateCarInNewGame(boolean SaveReply) : سازنده با ورودی اینکه ایا داده ها برای فیلم ذخیره شوند یا خیر . درصورت true بودن مقدار دهی فیلد دیگر

- public void run() : override متد پدر برای ایجاد Thread

-دیاگرامUml :‌ (رسم شده توسط افزونه ی easy Uml برای netBeans ) :



CreateCarInReply.java : ساخت ماشین ها با اطلاعات قبلی در زمان نمایش فیلم

-ارث بری :

-ارث بری از کلاس abstract با نام CreateCar

-پیاده سازی اینترفیس Runnable که پیاده سازی ان از پدر به فزرند سپرده شده است

-فیلدها‌:

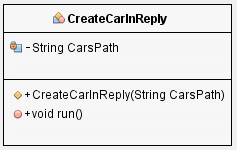
- private final String CarsPath نگهدارنده ی ادرس اطلاعات ذخیره شده ی ماشین ها

-متدها:

-public CreateCarInReply(String CarsPath) :‌ سازنده ی کلاس و مقدار دهی اولیه ی ادرس ماشین ها

- public void run() : override متد پدر برای ایجاد Thread

-دیاگرامUml :‌ (رسم شده توسط افزونه ی easy Uml برای netBeans ) :



SheepMoverInBotGame.java :‌ کلاس حرکت دهنده ی ادمک در بازی خودکار ادمک

-ارث بری :

-پیاده سازی اینترفیس Runnable برای ایجاد Thread

-فیلد ها :

-private final long SleepInMilliSecond : زمان بین دو حرکت ادمک

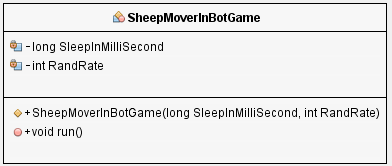
-private final int RandRate: نرخ حرکت به جلو

-متدها :

-public SheepMoverInBotGame(long SleepInMilliSecond, int RandRate) : سازنده ی کلاس و مقدار دهی به فیلدها با ورودی

-public void run() : پیاده سازی متد اینترفیس با Override کردن ان و ایجاد یک Thread

-دیاگرامUml :‌ (رسم شده توسط افزونه ی easy Uml برای netBeans ) :



SheepMoverInReply.java : کلاس حرمت دهنده ی خودکار ادمک در مشاهده ی فیلم

-ارث بری ها :

- اینترفیس Runnable جهت پیاده سازی Thread

-فیلد ها :

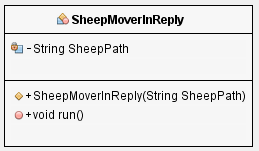
-private final String SheepPath : ادرس فایل نگهدارنده ی اطلاعات حرکت ادمک

-متدها :

-public SheepMoverInReply(String SheepPath) : سازنده ی کلاس و مقدار دهی فیلد از ورودی

-public void run() : پیاده سازی متد اینترفیس با Override کردن ان و ایجاد یک Thread

-دیاگرامUml :‌ (رسم شده توسط افزونه ی easy Uml برای netBeans ) :



پکیج CrossWalk.Object:

توضیحات : شامل ۸ کلاس

پکیج شامل کلاس های تعریف کننده ی اشیا بازی

کلاس ها :

Drawable.java : اینترفیس جهت پیاده سازی قابلیت کشیده شدن و نمایش برای کاربر

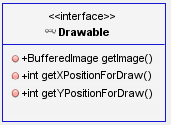
-متد ها :‌

- public BufferedImage getImage(): متد کوئری با خروجی عکس شی

- public int getYPositionForDraw(): متد کوئری با خروجی مختصات yشی

-public int getXPositionForDraw()‌: متد کوئری با خروجی مختصات x شی

-دیاگرامUml :‌ (رسم شده توسط افزونه ی easy Uml برای netBeans ) :

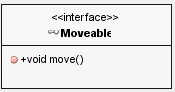


Moveable.java : اینترفیس جهت پیاده سازی قابلیت حرکت

-متدها :

- public void move() : متد حرکت دادن اشیا و تغییر مختصات انها

-دیاگرامUml :‌ (رسم شده توسط افزونه ی easy Uml برای netBeans ) :



Line.java : کلاس لاین های بازی

-ارث بری :

- پیاده سازی انترفیس Serializable جهت نوشتن درون فایل

- فیلدها :

- private int Id : ایدی هر لاین

- private int MaxCarSpeed حداکثر سرعت ماشین ها درون لاین

- private int MinCarSpeed حداقل سرعت ماشین ها درون لاین

- private boolean Direction :‌ جهت لاین (ثابت مورد استفاده درکلاس Const موجود است)

- private int YPosition :‌ مختصات Y هر لاین برای نمایش (X ان سراسر پنجره است)

- private ArrayList<Car> Cars : ایجاد رابطه ی Has a و دسترسی به ماشین های هر لاین

- private float[] CrosswalkXPosition : مختصات شروع و پایان X برای خط عابر پیاده

-private int CreatedCarCount تعداد ماشین های ساخته شده برای هر لاین

- public static int SheepCurrentLine : لاینی که ادمک روی ان قرار گرفته است

- private boolean CanCarOvertaking : ایا در لاین امکان سبقت وجود دارد ؟ بررسی وجود لاین کناری مناسب

-متد ها :

-public Line() : سازنده ی پیشفرض

public Line(int Id, int MaxCarSpeed, int MinCarSpeed, boolean Direction, int YPosition, boolean CanCarOvertaking) : سازنده ی کلاس

public Line(int Id, int MaxCarSpeed, int MinCarSpeed, boolean Direction, int Position, boolean CanCarOvertaking, int CreatedCarCount) :‌سازنده ی کلاس

-public int getYPosition()‌ : برگرداندن مختصات Y برای لاین

-public boolean getCanCarOvertaking() :‌ برگرداندن اینکه ایا امکان سبقت هست ؟

-public int getCreatedCarCount() : برگرداندن تعداد ماشین های ساخته شده

-public int getMinCarSpeed() برگرداندن حداقل سرعت ماشین های لاین

-public int getMaxCarSpeed() برگرداندن حداکثر سرعت ماشین ها

-public ArrayList<Car> getCars() :‌برگرداندن ماشین های درون لاین

-public float[] getCrosswalkXPosition() : برگرداندن محدوده ی مختصات x خط کشی عابر پیاده

-public boolean getDirection() برگرداندن جهت لاین

-public int getId() :‌برگرداندن ایدی لاین

-public String toString() :‌ override کردن برای نمایش و ذخیره ی اطلاعات لاین ها

-public void addCar(Car newCar) : متد اضافه کردن ماشینی که از قبل ساخته شده

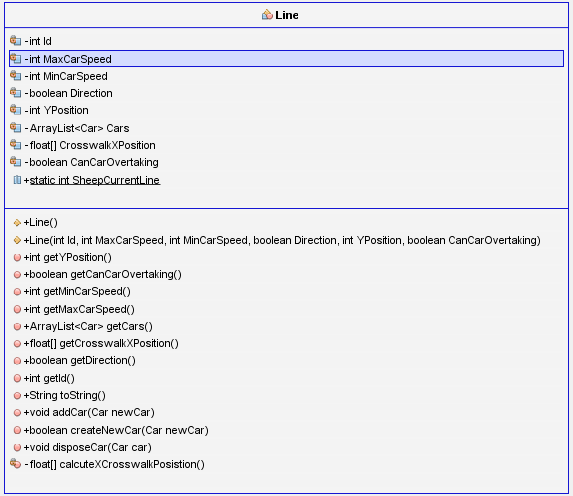
-public boolean createNewCar(Car newCar) :‌متد اضافه کردن ماشین جدید به لاین

-public void disposeCar(Car car) متد پاک کردن ماشین از لاین

-private float[] calcuteXCrosswalkPosistion() ‌: محاسبه ی محدوده ی x خط کشی عابرپیاده

-public void increaseCreatedCarCount() اضافه کردن یک واحد به تعداد ماشین های ساخته شده ی لاین

-دیاگرامUml :‌ (رسم شده توسط افزونه ی easy Uml برای netBeans ) :



Car.java : کلاس abstract ماشین های بازی

-ارث بری ها :

-پیاده سازی اینترفیس Moveable برای اضافه کردن قابلیت حرکت

-پیاده سازی اینترفیس Drawable برای اضافه کردن قابلیت نمایش دادن به کاربر

-پیاده سازی Serializable برای ذخیره کلاس روی فایل

-فیلد ها :

- private float HeadPosition : مختصات x جلوی ماشین

- public float Speed :‌ سرعت ماشین

- private final CarType CarType : رابطه ی Has a با کلاس CarType بیانگر ویژگی های ماشین

- private float TempCarSpeed :‌ نگه دارنده ی تغییرات سرعت در حین بازی

- private boolean IsNowOverTaking : زمانی True است که ماشین در حال سبقت گرفتن باشد

- private int IncreasLinePosition : اضافه کردن تدریجی y برای سبقت گرفتن تدریجی

- private Line Line : پیاده سازی رابطه ی Has a با لاین ماشین

-متد ها :

- public Car(float HeadPosition, int Speed, CarType CarType, Line Line)سازنده

- public Car(float HeadPosition, float Speed, String CarType, Line Line) : سازنده

- public CarType getCarType() برگرداندن شی از کلاس CarType برای نشان دادن نوع ماشین

- public Line getLine() برگرداندن لاین ماشین

- private float getSpeedInCrosswalk(): برگرداندن سرعت نزدیکی خط عابر پیاده

- public float getNowSpeed()برگرداندن سرعت ماشین در لحظه

- private float getSpeed() :‌سرعت ماشین که با ان سرعت ساخته شده

- public boolean getIsNowOverTaking() ایا در حال حاظر در حال سبقت گرفتن است؟

- public int getIncreasLinePosition() برگرداندن مقدار y که در سبقت کم یا زیاد میشود.

- public String toString() : override متد برای نمایش اطلاعات لاین

- public void setSpeed(float speed): تغییر سرعت

- public void setLine(Line line) : تغییر لاین

- public void setTempCarSpeed(float tempCarSpeed) : تغییر سرعت حال ماشین

- public void setIsNowOverTaking(boolean isNowOverTaking) تغییر وضعیت سبقت گرفتن

- public void setIncreasLinePosition(int increasLinePosition) :‌تنظیم مقدار y اضافه شده برای سبقت.

- public void setHeadPosition(float headPosition) تنظیم مختصات x ماشین

- public float getHeadPosition()‌برگرداندن مقدار x ماشین

- public boolean isNearToSheepAccident() ایا نزدیک به تصادف با ادمک هست؟

- public boolean isNeartheCrosswalk()بررسی اینکه ایا نزدیک خط عابر هست یا خیر.

- public BufferedImage getImage() پیاده سازی متد Drawable و برگرداندن عکس ماشین

- public abstract boolean isFirstCar() : متد abstract بیانگر اینکه ایا ماشین اول هست؟

- public abstract int getYPositionForDraw() : مختصات y جهت کشیدن ، متد Drawable

- public abstract void checkSheepAccident() : بررسی تصادف کردن ادمک

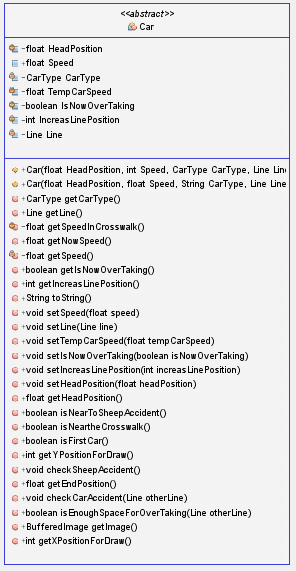
- public abstract float getEndPosition() گرفتن انتهای ماشین

- public abstract void checkCarAccident(Line otherLine) بررسی برخورد ماشین ها

- public abstract boolean isEnoughSpaceForOverTaking(Line otherLine) : بررسی اینکه فضا برای سبقت وجود دارد یا خیر

- public abstract int getXPositionForDraw() : مقدار X برای نمایش شکل

-دیاگرامUml :‌ (رسم شده توسط افزونه ی easy Uml برای netBeans ):



CarRtl.java : کلاس ماشین هایی که جهت انها از راست به چپ است

-ارث بری :

-ارث بری از کلاس abstract‌ با نام Car

-فیلد ها : ندارد

-متدها :

- public CarLtr(int Speed, CrossWalk.Object.CarType CarType, CrossWalk.Object.Line Line): سازنده

- public CarLtr(float HeadPosition, float Speed, String CarType, CrossWalk.Object.Line Line) سازنده

- public float getEndPosition() :‌override متد پدر توضیح در کلاس پدر

- public int getXPositionForDraw() :‌override متد پدر توضیح در کلاس پدر

- public void checkSheepAccident() :‌override متد پدر توضیح در کلاس پدر

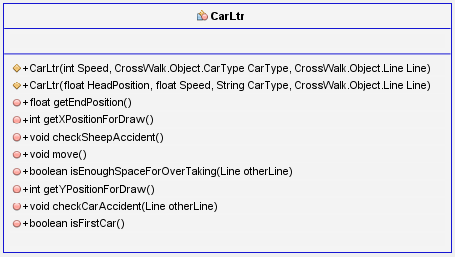
- public void move() :‌override متد پدر که پدر انرا از Moveable ارث برده است متد حرکت دادن شی

- public int getYPositionForDraw() :‌override متد پدر توضیح در کلاس پدر

- public void checkCarAccident(Line otherLine) :‌override متد پدر توضیح در کلاس پدر

- public boolean isFirstCar():‌override متد پدر توضیح در کلاس پدر

-دیاگرامUml :‌ (رسم شده توسط افزونه ی easy Uml برای netBeans ):



CarLtr.java‌ : کلاس ماشین های که جهت انها از چپ به راست است

-ارث بری :

-ارث بری از کلاس abstract‌ با نام Car

-فیلد ها : ندارد

-متدها :

- public CarRtl(int Speed, CrossWalk.Object.CarType CarType, CrossWalk.Object.Line Line): سازنده

- public carRtl(float HeadPosition, float Speed, String CarType, CrossWalk.Object.Line Line) سازنده

- public float getEndPosition() :‌override متد پدر توضیح در کلاس پدر

- public int getXPositionForDraw() :‌override متد پدر توضیح در کلاس پدر

- public void checkSheepAccident() :‌override متد پدر توضیح در کلاس پدر

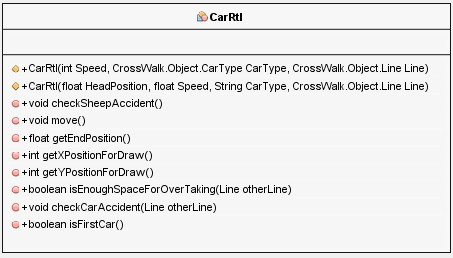
- public void move() :‌override متد پدر که پدر انرا از Moveable ارث برده است متد حرکت دادن شی

- public int getYPositionForDraw() :‌override متد پدر توضیح در کلاس پدر

- public void checkCarAccident(Line otherLine) :‌override متد پدر توضیح در کلاس پدر

- public boolean isFirstCar():‌override متد پدر توضیح در کلاس پدر

-دیاگرامUml :‌ (رسم شده توسط افزونه ی easy Uml برای netBeans ):



CarType.java : کلاسی که نوع ماشین را بیان میکند

-ارث بری ها : ندارد

-فیلد ها :

-private int CarWidth :‌ طول ماشین

-private int CarHeight ارتفاع ماشین

-private BufferedImage Image :‌ عکس ماشین

-private String CarNameAndType : نوع ماشین در واقع نام عکس ماشین

-متد ها :

-public CarType(boolean LineDirection)‌: سازنده

-public CarType(boolean LineDirection, String CarNameAndType) : سازنده

-private void initCarImage() : تنظیم عکس ماشین و محاسبه ی طول و عرض ماشین

-public int getCarWidth() : برگرداندن طول ماشین

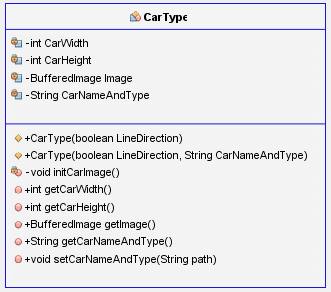
-public int getCarHeight() :‌برگرداندن ارتفاع ماشین

-public BufferedImage getImage() : گرفتن عکس ماشین

-public String getCarNameAndType() برگرداندن اسم ماشین (‌اسم عکس)

-public void setCarNameAndType(String path) تنظیم ادرس فایل عکس ماشین

-دیاگرامUml :‌ (رسم شده توسط افزونه ی easy Uml برای netBeans ) :



Sheep.java :‌ کلاس ادمک بازی

-ارث بری :

-پیاده سازی اینترفیس Drawable برای اضافه کردن قابلیت نمایش دادن به کاربر

-پیاده سازی Serializable برای ذخیره کلاس روی فایل

-فیلد :

- private float[] PositionOfSheep :‌ مختصات ادمک

- private float[] SheepSizeاندازه ی عکس ادمک

- private int[] MoveRate = new int[]{5, 25} : نرخ حرکت ادمک

- private BufferedImage[] ShapeImage :ارایه از عکس های ادمک

- private int ImageStatus : وضعیت ادمک یعنی در حال حرکت به کدام طرف

- private final String ImageCode : شماره ی عکس ادمک در واقع نام فایل ان

- private boolean SaveChanges : ایا اطلاعات برای فیلم ذخیره شوند ؟

- private int Level ‌:‌ لول ادمک

- private int Score امتیاز ادمک

- private int Life تعداد تلاش های ممکن برای ادمک

- private WriteReplyData WriteReplyData کلاس نویسنده ی اطلاعات ادمک

- public static boolean AutoMove = false ایا کاربر بازی میکند یا خود برنامه.

-متد ها:

- public Sheep(float[] PositionOfSheep, int[] Rate, int ImageStatus, String ImageCode, int Level, int Score, int Life) سازنده ی کلاس

- public Sheep() سازنده ی پیشفرض

- public void newSheep() تنظیم به حالت اولیه

- public String toString() override برای نمایش اطلاعات ادمک

- public final void setDefaultPosition() تنظیم مختصات به حالت اولیه

- public void setWriteReplyData(WriteReplyData writeReplyData)‌:‌تنظیم کردن کلاس نویسنده ی اطلاعات

- private void initImage() تنظیم کردن عکس ادمک

- public int getXPositionForDraw() متد ارث برده شده برای برگرداندن مختصات xشی

- public int getYPositionForDraw() متد ارث برده شده برای برگرداندن مختصات yشی

- public int[] getRate() متد برگشت دهنده ی نرخ حرکت

- public float getSheepWidth() برگرداندن طول ادمک

- public float getSheepHeight() برگرداندن ارتفاع ادمک

- public void setSaveChanges(boolean saveChanges) تنظیم کننده ی اینکه ایا اطلاعات برای فیلم ذخیره بشوند یا خیر

- private void goUp() متد حرکت به بالا

- private void goDown() متد حرکت به پایین

- private void goRight() متد حرکت به راست

- private void goLeft() متد حرکت به چپ

- public void move(int keyCode) پیاده سازی متد حرکت برای ادمک

- public void checkLine() بررسی شماره ی لاین که ادمک در ان حضور دارد.

- public void setRate(int[] rate) تنظیم کننده ی نرخ حرکت

- public void win() متد برنده شدن ادمک

- public void gameOver() متد باختن ادمک

- public BufferedImage getImage() برگرداندن عکس ادمک برای کشیده شدن

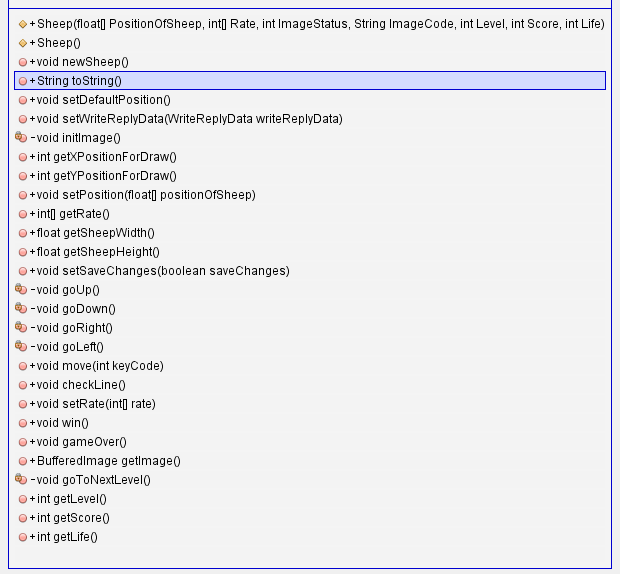
- private void goToNextLevel() متد ارتقا لول

- public int getLevel() برگرداندن لول

- public int getScore() برگردان امتیاز ادمک

- public int getLife() برگرداندن تعداد تلاش های باقی مانده

-دیاگرامUml :‌ (رسم شده توسط افزونه ی easy Uml برای netBeans ) :



پکیج CrossWalk.Menu :

توضیحات :

پکیج منو های بازی

کلاس ها : شامل ۱۳ کلاس

Menu.java : کلاس abstract جهت ارث بری منوها و پیاده سازی فیلد ها و متدهای مشترک

-ارث بری : ندارد

-فیلد ها :

- private final String Title : عنوان منو

- private final int Height ارتفاع منو

- private final int Width طول منو

- public JFrame Frame فریم اصلی بازی

-متدها :

- public Menu(String Title, int Height, int Width) سازنده ی کلاس

- public void show() متد نمایش دهنده ی منو

- public void setInit() تنظیم کننده ی تنظیمات فریم اصلی

- protected JButton createButton(String Title, ActionListener actionListener) : متد سازنده ی دکمه برای بازی

- protected JSpinner createSpinner(String Title, int defaultValue, int minValue, int maxValue, int step, ChangeListener changeListener) : متد سازنده ی کمبو باکس یا همان منوی ابشاری

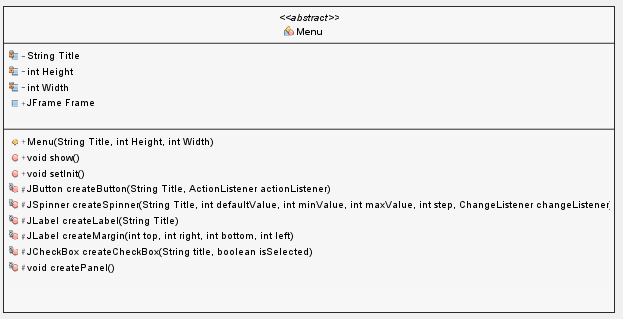
- protected JLabel createLabel(String Title) : متد سازنده ی لیبل

- protected JLabel createMargin(int top, int right, int bottom, int left) متد سازنده ی فاصله بین اجزای منو

- protected JCheckBox createCheckBox(String title, boolean isSelected): متد سازنده ی چک باکس

- protected abstract void createPanel() متد سازنده ی پنل اصلی منو که پیاده سازی بدنه به فرزندان سپرده شده است.

-دیاگرامUml :‌ (رسم شده توسط افزونه ی easy Uml برای netBeans ) :



MainMenu.java :‌ کلاس منوی اصلی

ارث بری‌:

-کلاس Menu

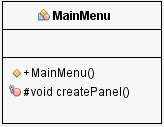
-فیلد ها : ندارد

-متد ها :

- public MainMenu() سازنده ی پیشفرض که تعیین کننده ی فیلد های پدر است

- protected void createPanel() متد پیاده سازی شده ی پدر

- دیاگرامUml :‌ (رسم شده توسط افزونه ی easy Uml برای netBeans ) :



CarMenu.java : کلاس انتخاب ماشین در تنظیمات

ارث بری‌:

-کلاس Menu

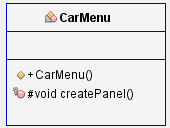
-فیلد ها : ندارد

-متد ها :

- public CarMenu() سازنده ی پیشفرض که تعیین کننده ی فیلد های پدر است

- protected void createPanel() متد پیاده سازی شده ی پدر

- دیاگرامUml :‌ (رسم شده توسط افزونه ی easy Uml برای netBeans ) :



LineMenu.java :‌ کلاس انتخاب لاین ها ، خط عابر پیاده ، خط بین لاین ها

ارث بری‌:

-کلاس Menu

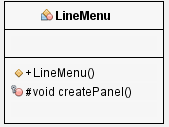
-فیلد ها : ندارد

-متد ها :

- public LineMenu () سازنده ی پیشفرض که تعیین کننده ی فیلد های پدر است

- protected void createPanel() متد پیاده سازی شده ی پدر

- دیاگرامUml :‌ (رسم شده توسط افزونه ی easy Uml برای netBeans )



ReplyMenu.java :‌ کلاس منوی نمایش دهنده ی بازی های انجام شده .

ارث بری‌:

-کلاس Menu

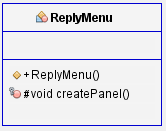
-فیلد ها : ندارد

-متد ها :

- public ReplyMenu() سازنده ی پیشفرض که تعیین کننده ی فیلد های پدر است

- protected void createPanel() متد پیاده سازی شده ی پدر

- دیاگرامUml :‌ (رسم شده توسط افزونه ی easy Uml برای netBeans ):



ResumeGameMenu.java :‌ کلاس منوی توقف در حین بازی

ارث بری‌:

-کلاس Menu

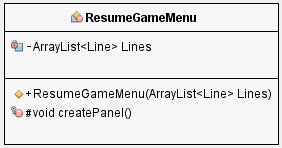
-فیلد ها : ندارد

-متد ها :

- public LineMenu () سازنده ی پیشفرض که تعیین کننده ی فیلد های پدر است

- protected void createPanel() متد پیاده سازی شده ی پدر

- دیاگرامUml :‌ (رسم شده توسط افزونه ی easy Uml برای netBeans )



SelectMapMenu.java : کلاس منوی انتخاب مپ قبل از شروع بازی

ارث بری‌:

-کلاس Menu

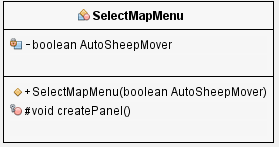
-فیلد ها : ندارد

-متد ها :

- public LineMenu () سازنده ی پیشفرض که تعیین کننده ی فیلد های پدر است

- protected void createPanel() متد پیاده سازی شده ی پدر

- دیاگرامUml :‌ (رسم شده توسط افزونه ی easy Uml برای netBeans ):



SettingMenu.java :‌ کلاس جهت نمایش تنظیمات اصلی بازی

ارث بری‌:

-کلاس Menu

-فیلد ها : ندارد

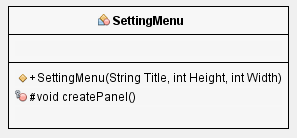
-متد ها :

- public SettingMenu() سازنده ی پیشفرض که تعیین کننده ی فیلد های پدر است

- protected void createPanel() متد پیاده سازی شده ی پدر

- دیاگرامUml :‌ (رسم شده توسط افزونه ی easy Uml برای netBeans )

.



SheepMenu.java : کلاس برای انتخاب شخصیت بازی

ارث بری‌:

-کلاس Menu

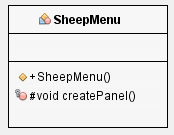
-فیلد ها : ندارد

-متد ها :

- public LineMenu () سازنده ی پیشفرض که تعیین کننده ی فیلد های پدر است

- protected void createPanel() متد پیاده سازی شده ی پدر

- دیاگرامUml :‌ (رسم شده توسط افزونه ی easy Uml برای netBeans )



GameSetting.java :‌ کلاس استاتیک جهت ذخیره سازی تنظیمات.

ارث بری‌:ندارد

-فیلد ها :

-شامل تعداد زیادی فیلد ذخیره کننده ی اطلاعات بازی که در هربازی ممکن است تغییر کنند

-متد ها :

-شامل متد get,set برای همه فیلد ها

- public static void SaveChanges() ذخیره ی تنظیمات در یک فایل xml

- public static void UpdateSettings() خواندن تنظیمات از یک فایل xml

- public static void readSetting(String pathAddress) خواندن تنظیمات از یک فایل xml که ادرس از ورودی گرفته میشود .

- public static void readSetting(String pathAddress) نوشتن تنظیمات درون یک فایل xml که ادرس از ورودی گرفته میشود .

- دیاگرامUml :‌ (رسم شده توسط افزونه ی easy Uml برای netBeans )



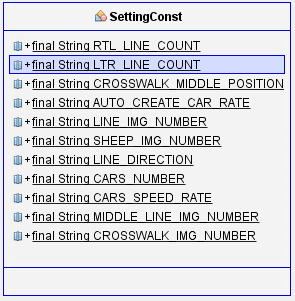
SettingConst.java :‌ کلاس جهت ذخیره ی ثابت های بازی

-ارث بری : ندارد

-فیلد ها :

- شامل فیلد هایی که ثابت های تنظیمات را ذخیره میکنند مانند نام حداکثر تعداد ممکن برای لاین های انتخابی که در فایل xml ذخیره شده

- دیاگرامUml :‌ (رسم شده توسط افزونه ی easy Uml برای netBeans ) :



MenuConst.java : کلاس نگهدارنده ی ثابت های منو

-ارث بری : ندارد

-فیلد ها :

- شامل فیلد هایی که ثابت های منو را ذخیره میکنند مانند حداکثر تعداد ممکن برای لاین های انتخابی

- دیاگرامUml :‌ (رسم شده توسط افزونه ی easy Uml برای netBeans ) :

