

**Dokumentation**

**Projekt**

**PaperboyClone**

**von Martin Freudenberg und Timo Kramer**

**OSMI**

**Pattern u. Frameworks**

# **1. Beschreibung aus Sicht des Anwenders**

## **1.1 Allgemein**

Der Benutzer findet ein 2D Spiel vor welches er mittels Tastatur bedienen kann. In diesem Spiel steuert der Benutzer einen Zeitungsjungen. Ziel ist es möglichst viele Zeitungen an die Abonnenten auszuliefern. Abonnenten sind durch grüne Briefkästen gekennzeichnet. Durch das ausliefern der Zeitungen erhält der Spieler Punkte, sobald alle Häuser abgefahren wurden endet das Spiel und die Gesamtpunktzahl wird in einer Highscore Tabelle gespeichert.

Um genügend Zeitungen zum verteilen zu haben kann der Benutzer während des Spiels die gelegentlich erscheinenden Zeitungstapel aufsammeln, diese geben ihm zehn zusätzliche Zeitungen. Außerdem erscheinen immer wieder verschiedene Hindernisse auf Straße und Fußweg. Diesen sollte der Spieler möglichst geschickt ausweichen, denn bei Zusammenstoß wird ihm ein Leben abgezogen. Um immer in guter Reichweite zu den Briefkästen zu bleiben kann der Benutzer die Geschwindigkeit variieren.

Sobald seine Leben verbraucht sind endet das Spiel sofort.

## **1.2 Graphische Oberfläche**

Der Benutzer sieht ein Fenster mit verschiedenen Menüs, Spielegrafiken und Schriften. Der Benutzer nimmt die Bewegungen verschiedener Objekte wahr. Das Spiel wird ausschließlich zweidimensional ausgegeben.

## **1.3 Sound**

Es wird sobald das Menü sichtbar ist eine Menü Melodie abgespielt. Weiterhin wird während des spielens eine Spiel Melodie gespielt.

Bei bestimmten Ereignissen wird ein dafür definierter Sound kurz abgespielt. Beispielsweise das Punkte bekommen startet einen Sound. Sobald Zeitungen mit der Umgebung kollidieren oder in Briefkästen landen wird ein Sound abgespielt.

## 1.4 Steuerung

Das Spiel wird mit Tastatur bedient.

Bedienung im Menu:

Pfeil hoch - nach oben navigieren

Pfeil runter - nach unten navigieren

Enter – je nach Kontext, oft wechsel in das nächste Menü

Pfeil links – vorherige Auswahl (Options Menü)

Pfeil rechts – nächste Auswahl

Bedienung im Spiel:

Pfeil hoch - Beschleunigen

Pfeil runter - Bremsen

Pfeil links – nach links lenken

Pfeil rechts – nach rechts lenken

Y – Zeitung nach links werfen

X – Zeitung nach rechts werfen

S – Sound ausschalten

## **2. Beschreibung aus Sicht der Entwickler**

### **2.1 Allgemein:**

Das Spiel wurde in Java mit dem Framework libgdx entwickelt. Bei der Programmierung wurden verschiedene Entwurfsmuster eingesetzt. Zur Datenspeicherung der Settings und der Highscores wurde JSON verwendet. Die Animation des Spielercharakter wurde über „Sprite Sheet Animationen“ umgesetzt. Threads wurden eingesetzt um parallel eine Melodie abspielen zu können.

### **2.2 Asset Verwaltung**

Alle Sounds, Grafiken und Fonts werden zentral in der Asset Klasse abgelegt. Dabei bedient sich diese dem AssetManager von libgdx. Es werden bei Programmstart bestimmte Assets sofort geladen. Diese werden meist schon in den Menüs gebraucht. Sobald das Spiel starten soll wird erst der Ladebildschirm eingeblendet. Dort werden alle Assets in eine Warteschlange getan und dann kontinuierlich geladen. Bei Programm ende wird der AssetManager geleert.

### **2.3 Level erzeugen**

Vor Spielstart wird das Level mittels einer statischen Methode des Levelgenerators erstellt. Dabei werden die Positionen der Häuser und zugehöriger Briefkästen ermittelt. Anschließend wird errechnet wie viele Häuser zu den Abonnenten gezählt werden sollen und diese per zufälliger Verteilung gesetzt. Die Hausgrafik und der Abstand zwischen den Häusern wird per Zufall bestimmt.

### **2.4 Eingabe abfragen**

Die Eingabe während des Spiels dient dazu den Spieler zu bewegen. Über Polling wird überwacht ob eine bestimmte Taste gedrückt wurde und entsprechend gehandelt bei Tastendruck.

## 2.5 Kollisionsabfrage

Um zu bestimmen ob sich bestimmte Objekte überlappen werden für einige Objektpaare Tasks definiert, welche in jedem Frame durch die Listen der jeweiligen Typen laufen und jedes Objekt mit jedem auf Überlappung prüfen. Sobald eine Kollision festgelegt wurde wird eine entsprechende Methode von den kollidierten Objekten aufgerufen. In dieser Methode ist definiert was passieren soll wenn dieses Objekt mit diesem Objekttypen kollidiert.

## 2.6 Entity Verwaltung

Alle Entities verfügen über einheitliches Interface. Über dieses Interface ruft die GameWorld Klasse die definierten Methoden auf. Dies ermöglicht die einheitliche Verwaltung aller Entities. Zu den Entities gehören Haus, Mailbox, Paper, PaperPile, Player, Obstacle. Gespeichert werden diese in einer Map geordnet nach dem Typ in der GameWorld Klasse.

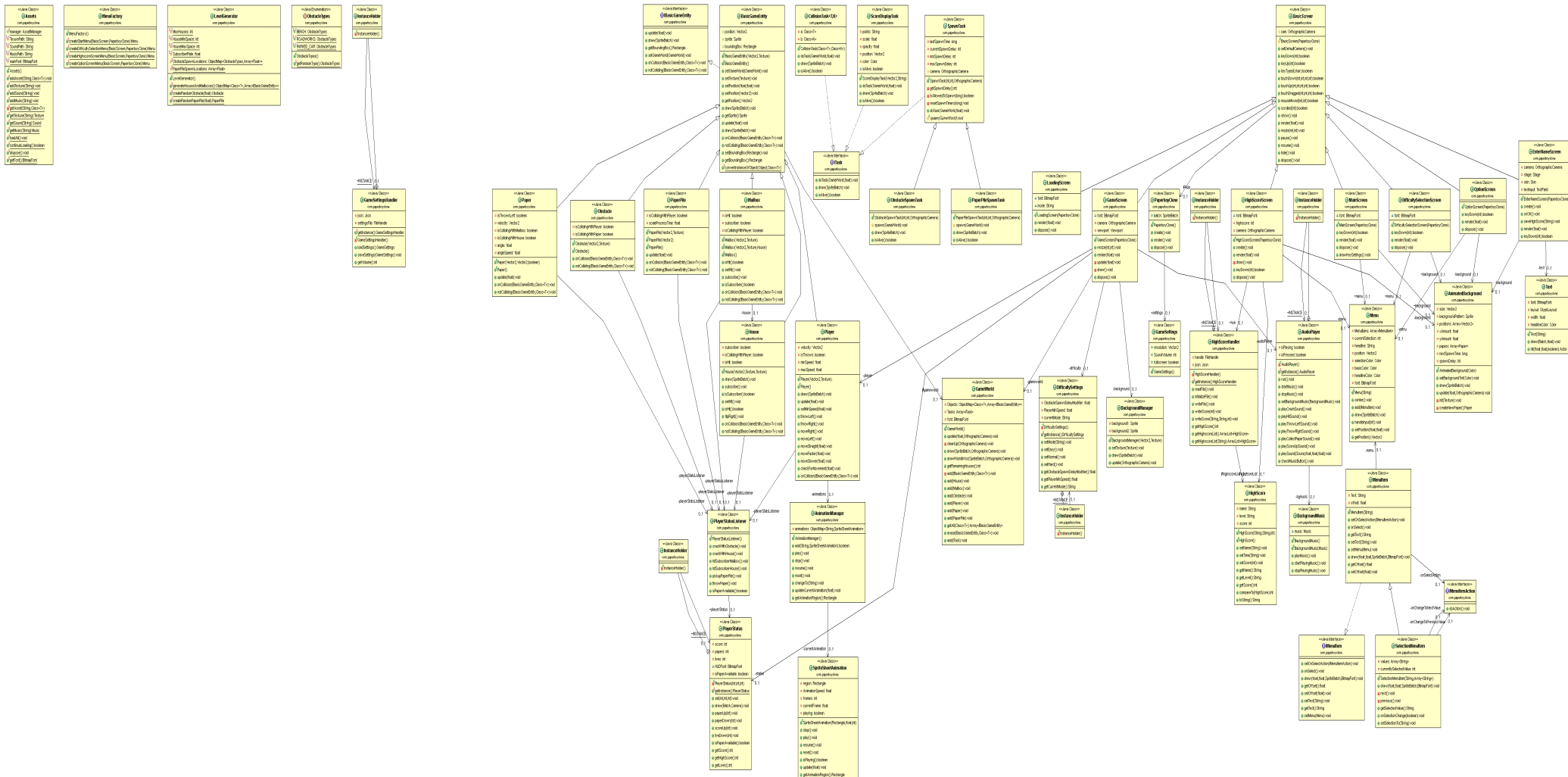
## 2.7 Datenspeicherung

Die Datenspeicherung geschieht über die libgdx Klassen die Methoden bereitstellen um Klassen direkt im JSON Format abzuspeichern. Gespeichert werden die Einstellungen in der settings.json und die Highscores in der PaperboyCloneHighscores.json. Beide werden im Benutzerverzeichnis des Betriebssystems abgelegt.

## 2.8 Menüs

Die Menu Klasse bietet einen Container für MenuItem und MenuItemSelectionItem. Außerdem ist sie für die Überwachung der Benutzereingabe zuständig. Diese Eingabe Überwachung wird über Events durchgeführt. Für jedes MenuItem und MenuItemSelectionItem kann ein Command definiert werden welcher bei Auswahl dieses Punktes ausgeführt werden soll. Das Command Pattern wurde hierbei eingesetzt. Erzeugt werden Menüs über die MenuFactory um den umfangreichen Initialisierungscode auszulagern.

# 3. Design der Anwendung



## Assets Quellen

The song *8-Bit\_Ninjas\_-12-Shiny\_Spaceship.mp3* was downloaded from *FreeMusicArchive* ([http://freemusicarchive.org/music/8-Bit\\_Ninjas/Party\\_in\\_Space/kzz007-12-8-bit\\_ninjas-\\_shiny\\_spaceship](http://freemusicarchive.org/music/8-Bit_Ninjas/Party_in_Space/kzz007-12-8-bit_ninjas-_shiny_spaceship)) and is licensed under:

Attribution-NonCommercial-ShareAlike 3.0 Unported (CC BY-NC-SA 3.0)  
<https://creativecommons.org/licenses/by-nc-sa/3.0/legalcode>

The Music "MenuMusic.ogg" was downloaded from newgrounds.com (<http://www.newgrounds.com/audio/listen/497160>) and is licensed under:

Attribution-NonCommercial-ShareAlike 3.0 Unported  
<http://creativecommons.org/licenses/by-nc-sa/3.0/legalcode>

## The soundeffects

- "Bike\_Wreck.wav" from 1sticky8 (<http://www.freesound.org/people/1sticky8/sounds/78483/>) is licensed under Creative Commons CC0 1.0 (<http://creativecommons.org/publicdomain/zero/1.0/legalcode>)
- "Hitting a tin can" from ryanconway (<http://www.freesound.org/people/ryanconway/sounds/240800/>) is licensed under Creative Commons CC BY 3.0 (<http://creativecommons.org/licenses/by/3.0/legalcode>)
- "scoreUP.ogg" (<http://www.freesound.org/people/Cabeeno%20Rossley/sounds/126418/>) is licensed under Creative Commons Attribution 3.0 Unported (CC BY 3.0) (<http://creativecommons.org/licenses/by/3.0/legalcode>)

are downloaded from: Freesound.org (<http://www.freesound.org/>)

## The soundeffects

- 961-Arrow-Left-To-Right.mp3 (<http://soundbible.com/961-Arrow-Left-To-Right.html>)
- 962-Arrow-Right-To-Left.mp3 (<http://soundbible.com/962-Arrow-Right-To-Left.html>)

are downloaded from SoundBible.com and licensed under Creative Commons CC BY 3.0 (<https://creativecommons.org/licenses/by/3.0/legalcode>)

## Font:

The font "banana brick" (bbrick.fnt, bbrick.png) is downloaded from (<http://openfontlibrary.org/en/font/banana-brick>) and is licensed under OFL (SIL Open Font License) ([http://scripts.sil.org/cms/scripts/page.php?site\\_id=nrsi&id=OFL](http://scripts.sil.org/cms/scripts/page.php?site_id=nrsi&id=OFL))

## Graphics:

The graphics "house\_1.png", "house\_2.png", "house\_3.png", "house\_4.png" from "Alucus" are downloaded from (<http://www.deviantart.com/art/Pallet-town-tiles-157214973>) and are licensed under Creative Commons Attribution-NonCommercial-NoDerivs 3.0 Unported (<http://creativecommons.org/licenses/by-nc-nd/3.0/>)

"parked\_car1.png" from "dravenx" is downloaded from (<http://opengameart.org/content/basic-topdown-view-car>) and is licensed under Creative Commons Attribution-ShareAlike 3.0 Unported (<http://creativecommons.org/licenses/by-sa/3.0/>) "parked\_car2.png", "parked\_car3.png", "parked\_car4.png" are recolored versions of "parked\_car1.png"

"backgroundPattern.png" from Dax Kieran is downloaded from (<http://subtlepatterns.com/triangular/>) and is licensed under Creative Commons Attribution-ShareAlike 3.0 Unported (<http://creativecommons.org/licenses/by-sa/3.0/>)



## Installationshinweise

Benötigte Java Version: Java 7

Ausführen: paperpoyclone.jar ausführen

Auflösung: Ideal 16:9 oder 16:10 Auflösung,  
bei 4:3 nicht kleiner als 800x600