

Flappy Karl

Beni Antesberger, Felix Schmid-Burgk, Mariella Kircher

Inhalt

01

Idee

02

Das Spiel

03

Kommunikation und
Planung
Probleme & Lösungen

04

Code
Probleme & Lösungen

05

Design
Probleme & Lösungen


06

Fazit

01 Idee

- Basierend auf Flappy Bird
- Inspiriert von der Ästhetik kommunistischer Klischees
- Mit Extras (Fellknäuel schießen, besondere Barrikaden etc.)





02

Das Spiel

03 Kommunikation



Probleme

- Aufgabenverteilung unklar
- Fortschritt teilweise unklar



Lösung(en)

- Am Anfang der Stunde kurze Gespräche wer was gerade macht und was man heute schaffen möchte
- Zukünftig: Kanban Board o. ä.

03 Planung



Probleme

- Unklare Anforderungen
- Keine Einhaltung



Lösung

- Nächstes Mal RoadMap kontinuierlich anpassen

Roadmap



Woche 1
Brainstorming,
Planung und
Dokumentation
(Pflichtenheft)



Woche 2
Erster Code und
Aufgabenverteilung



Woche 3
Erstellung der
Präsentation und
weiter Code



Woche 4
Fertigstellung des
Codes und
Feinschliff



04 Design



Probleme

- Passenden Hintergrund finden (bewegende Bilder)
- Alternative Designs zum Original finden
- Arbeit mit Pixeln (Zeichnen)



Lösungen

- Ein GIF macht die Arbeit für dich
- Letztendlich festgelegt

05 Code



Probleme

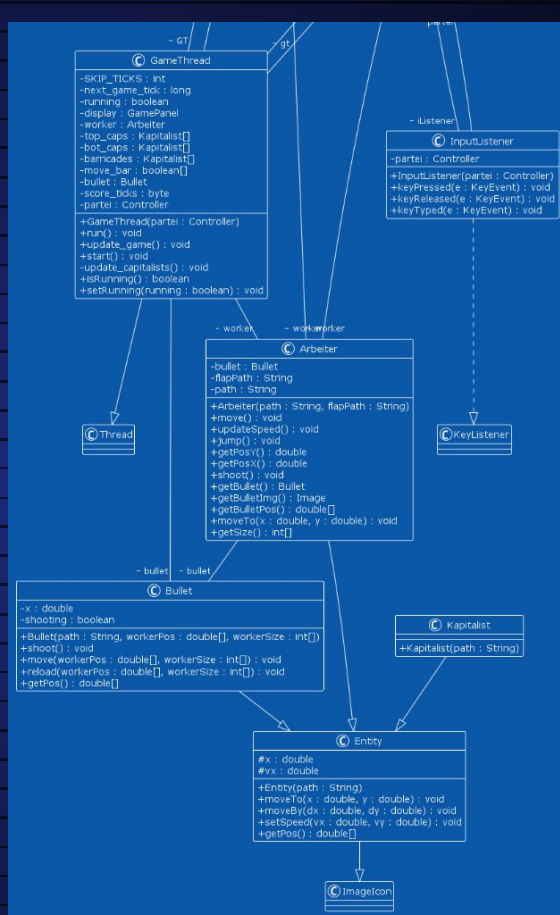
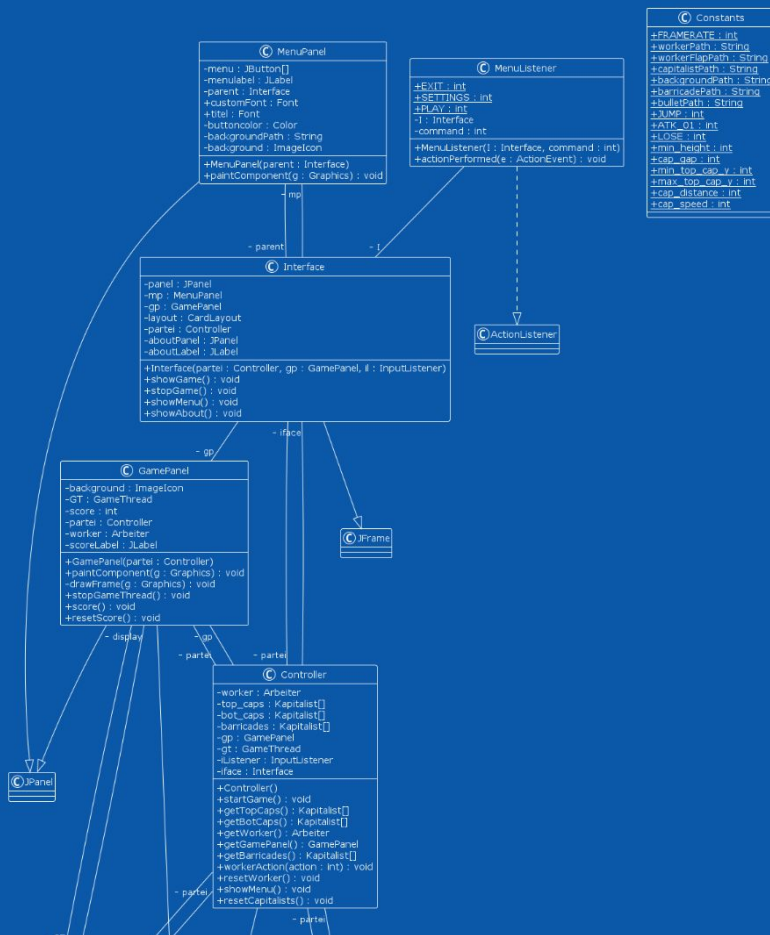
- Aufteilung in MVC
- Kollision
- Bugs
- Wechsel vom MenuPanel zum GamePanel



Lösung(en)

- Projekt umschreiben
- `if()`
- `_(ツ)_/`
- ✨ **CardLayout** ✨

Klassendiagramme



Ⓒ GameThread

-SKIP_TICKS : int
-next_game_tick : long
-running : boolean
-display : GamePanel
-worker : Arbeiter
-top_caps : Kapitalist[]
-bot_caps : Kapitalist[]
-barricades : Kapitalist[]
-move_bar : boolean[]
-bullet : Bullet
-score_ticks : byte
-partei : Controller

+GameThread(partei : Controller)
+run() : void
+update_game() : void
+start() : void
-update_capitalists() : void
+isRunning() : boolean
+setRunning(running : boolean) : void



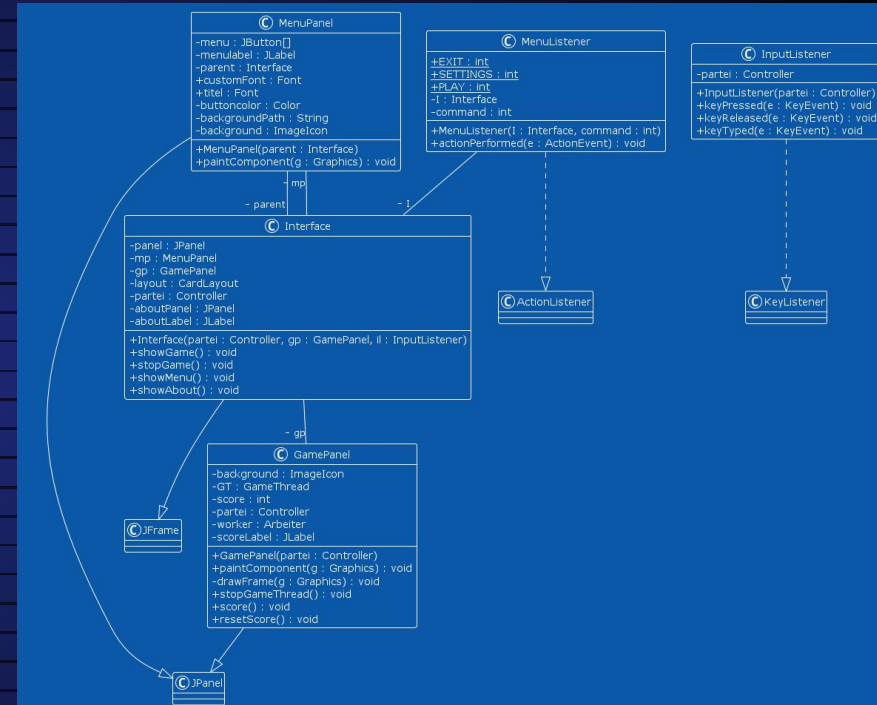
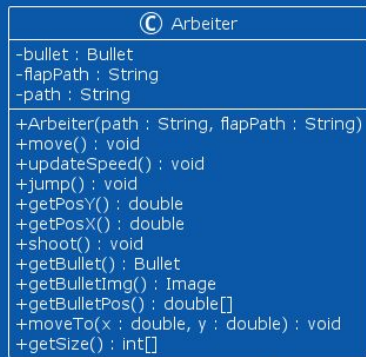
Ⓒ Constants

+FRAMERATE : int
+workerPath : String
+workerFlapPath : String
+capitalistPath : String
+backgroundPath : String
+barricadePath : String
+bulletPath : String
+JUMP : int
+ATK_01 : int
+LOSE : int
+min_height : int
+cap_gap : int
+min_top_cap_y : int
+max_top_cap_y : int
+cap_distance : int
+cap_speed : int

Ⓒ Controller

-worker : Arbeiter
-top_caps : Kapitalist[]
-bot_caps : Kapitalist[]
-barricades : Kapitalist[]
-gp : GamePanel
-gt : GameThread
-iListener : InputListener
-iface : Interface

+Controller()
+startGame() : void
+getTopCaps() : Kapitalist[]
+getBotCaps() : Kapitalist[]
+getWorker() : Arbeiter
+getGamePanel() : GamePanel
+getBarricades() : Kapitalist[]
+workerAction(action : int) : void
+resetWorker() : void
+showMenu() : void
+resetCapitalists() : void



FAZIT