



# **BLOCKPRAKTIKUM MULTIMEDIAPROGRAMMIERUNG**

Henri Palleis  
Aufgabenstellung

# Das Projekt

- Vier Teams à 5 Personen
- Spielspaß!
  - Spielspaß genauso wichtig wie Umsetzung der Funktionalität
- Deshalb: Aufgabe angepasst an die Fähigkeiten/Interessen im Team.

# Ideen

- **Arkanoid / Pong**
  - Kollisionen
  - Spielstände
  - Level



# Ideen

- **Tetris/Candy Crush**
  - Datenstrukturen
  - Level
  - Spielstände



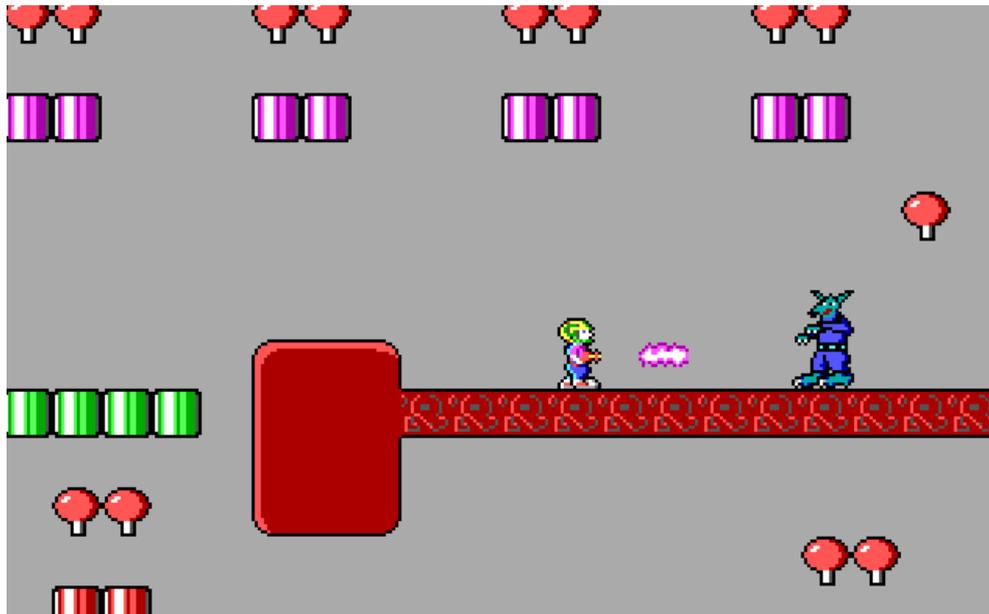
# Ideen

- **Shoot-em-up/Space Invaders/1942**
  - Gegner
  - Schussfunktion
  - Eingeschränkte Bewegung



# Ideen

- **Einfacher Platformer**
  - Schwerkraft, einfache Physik



# Ideen

- **Einfacher Platformer: „Endless Running“**
  - Stetige Vorwärtsbewegung (mit der Zeit schneller)
  - Welt wird automatisch generiert (immer neu)



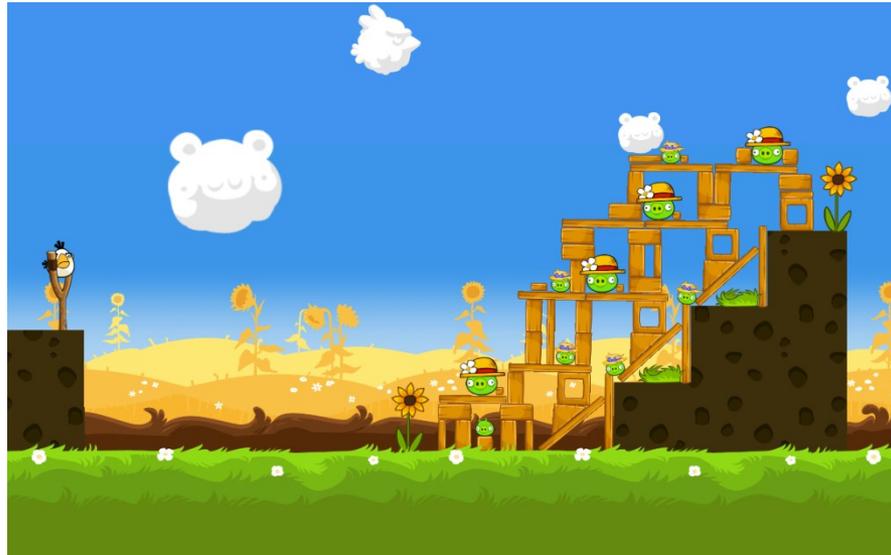
# Ideen

- **Bomberman / Pacman**
  - Tile-basierte Bewegungen
  - Gegner-KI



# Ideen

- **Angry Birds/Physik-Simulationen**
  - Gameplay basiert maßgeblich auf Physik-Engine



# Ideen

- **Arkanoid / Pong** (collisions, stable frame rate, score, levels)
- **Tetris** (data structures and how they relate to gaming)
- **1942 / Shoot-em-up** (enemies, bullets)
- **simple platformer / pinball game** if your engine does platformers (gravity-based collisions)
- **Bomberman / Pacman** (tile-based movement, complex enemy AI)
- **Two-player game** of any of the types above (two player inputs)

# Grundsätzliche funktionale Anforderungen

- Anwendung besteht neben dem Spiel selbst aus:
  - Startmenü
  - Impressum
  - (Kurz-)Anleitung (z.B. In-Game-Anleitung)
- Spiel ist eingebettet in eine grafisch zum Spiel passende Website
- Eigenes einzigartiges Spielkonzept

# Weitere Anforderungen

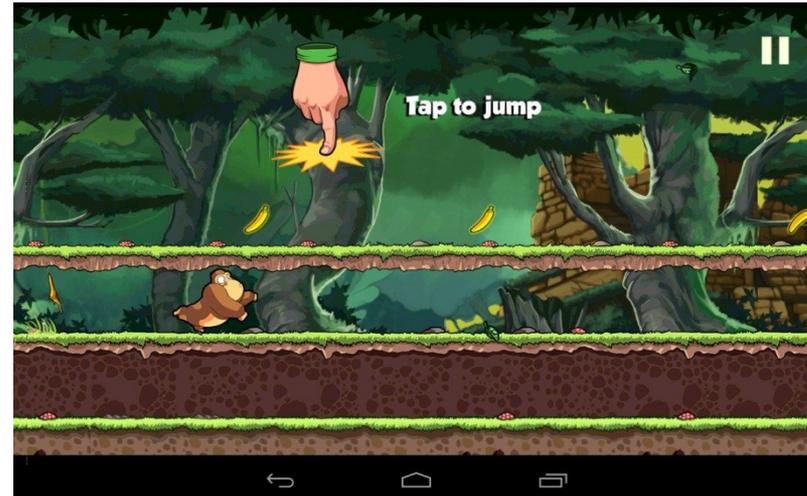
- **Gestaltung:** Alle grafischen Bestandteile des Spiels müssen im Team umgesetzt werden.
  - Sprites für Animationen
  - Objekte, Hintergründe, Texte etc.
  - Wichtig: eigene „Linie“ erkennbar, stimmiges Konzept
- **Sound:** Entweder geeignete Sounds (mit passender Lizenz!) suchen oder eigene Musik und Sounds machen (-> Audiolabor)!



# Nicht-funktionale Anforderungen

- Animation und Interaktion
- Eigenes gestalterisches Thema
- Besonderes Augenmerk auf das Feeling
  - Richtiger Schwierigkeitsgrad
  - Gute Benutzbarkeit
  - Spaß am Spiel
- Flexible Anwendungsstruktur
- Robustheit, Fehlerfreiheit
- Ergebnis soll publizierbar sein
  - Kein urheberrechtlich geschütztes Material verwenden (gilt natürlich auch für Code)
  - Abgeschlossenheit

# Fließendes Spielkonzept



# Weitere Entwicklungswerkzeuge

- GIT-Repository zur Verwaltung aller für das Projekt relevanten Dateien.
- JSHint, A Static Code Analysis Tool for JavaScript (<http://jshint.com/>)
- Grunt, Automation: <http://gruntjs.com/>
- Zur Umsetzung des Test-First-Ansatzes:
  - QUnit: <http://qunitjs.com/>

# Die ersten Schritte

- Teams bilden
- Informationen zu den Originalspielen beschaffen
- Planning Game
- GIT-Repositories und GitHub-Profile anlegen
- Teamaufgaben (vorläufig) verteilen
- ...



# FRAGEN?

# Abschlusspräsentation

- Freitag nächste Woche
  - Jedes Team präsentiert ihr Spiel
    - Zuerst Folien, dann Live-Demo
    - Insgesamt ca. 15-20 Minuten pro Team

# Präsentationsfolien

- Entwicklungsideen
- Ansätze, grundlegende Ideen
- Verwendete Technologien
- Beschreibung der Spielidee
- Schwierigkeiten/Erfolge bei der Entwicklung
- Anwendung der verschiedenen XP-Techniken
- Lustige Bugs
- Einschätzung zur Eignung der verwendeten Technologie für die Spieleprogrammierung
- Sonstiges (Ideen, Verbesserungsvorschläge)

# Abgabe

- Abgabe einer endgültigen Version
  - Erfüllt alle Anforderungen
  - Alle Quelldateien (HTML, JS, Sounds, Grafiken, Schriften)
  - Publizieren einer Spielwebseite
- Eventuell benötigte 3rd-Party-Software
- Fertigstellung bis Ende nächster Woche