

Lastenheft „4 Gewinnt“ in pygame

Sarah Burghardt



20. Februar 2025

Inhaltsverzeichnis

[1. Projektbeschreibung 2](#_Toc191323588)

[1.1 Spielprinzip: 2](#_Toc191323589)

[1.2 Einzelspieler-Modus: 2](#_Toc191323590)

[1.3 Erweiterungen: 2](#_Toc191323591)

[1.4 Benutzererlebnis: 2](#_Toc191323592)

[2. Merkmale 2](#_Toc191323593)

[2.1 User Interface: 2](#_Toc191323594)

[2.2 Steuerung: 2](#_Toc191323595)

[2.3 Spiellogik: 3](#_Toc191323596)

[2.4 KI-Gegner: 3](#_Toc191323597)

[3. Grafiken 3](#_Toc191323598)

[4. Soundeffekte 5](#_Toc191323599)

[5. Erweiterungsmöglichkeiten 5](#_Toc191323600)

[6. Zeitstrahl 5](#_Toc191323601)

[7. Mögliche nächste Schritte 5](#_Toc191323602)

[7.1 Entwicklung des Prototyps: 5](#_Toc191323603)

[7.2 KI-Implementierung: 5](#_Toc191323604)

[7.3 Erweiterungen: 5](#_Toc191323605)

[7.4 Testen: 6](#_Toc191323606)

[8. Beispiel Siege oder unentschieden 6](#_Toc191323607)

# Projektbeschreibung

## Spielprinzip:

* „4 Gewinnt“ ist ein strategisches Denkspiel für zwei Spieler, das auf einem 7x6-Raster gespielt wird. Ziel ist es, vier Spielsteine der eigenen Farbe horizontal, vertikal oder diagonal in einer Reihe zu platzieren, bevor der Gegner dies schafft.

## Einzelspieler-Modus:

* Es gibt eine KI mit verschiedenen Schwierigkeitsstufen, die von einfachen Zügen bis hin zu komplexen Strategien reicht.

## Erweiterungen:

* Optionale Features wie alternative Spielfeldgrößen und Power-Ups sind geplant.

## Benutzererlebnis:

* Eine intuitive Benutzeroberfläche, ansprechende Animationen und Soundeffekte sollen ein angenehmes Spielerlebnis schaffen.

# Merkmale

## User Interface:

Muss-Kriterien:

* + 7x6-Raster
  + Statuszeile
  + Spielende-Anzeige
  + Begrüßungsbildschirm

Kann-Kriterien:

* Highscore-Anzeige
* Verschiedene Designs
* Anzeige der letzten Züge

## Steuerung:

Muss-Kriterien:

* Pfeiltasten zur Spaltenauswahl
* Leertaste zum Platzieren der Steine
* ESC zum Beenden
* R zum Neustart

Kann-Kriterien:

* F10 Speichern des Spielstands
* P Pausieren
* Maussteuerung

## Spiellogik:

Muss-Kriterien:

* Abwechselndes Setzen der Steine
* Überprüfung auf Gewinn
* Spielende bei Sieg oder Unentschieden

Kann-Kriterien:

* Animierte Fallbewegung der Steine
* Soundeffekte
* Anpassbare Spielfeldgröße

## KI-Gegner:

Muss-Kriterien:

* Mehrere Schwierigkeitsstufen
* Grundlegende KI-Strategien

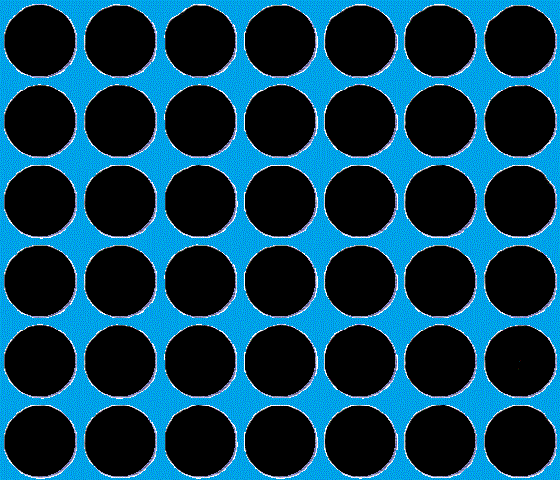
Kann-Kriterien:

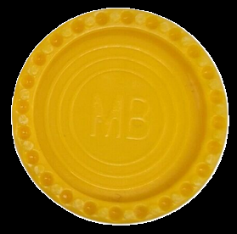
* Erweiterte KI
* Anpassbare KI-Einstellungen

# Grafiken

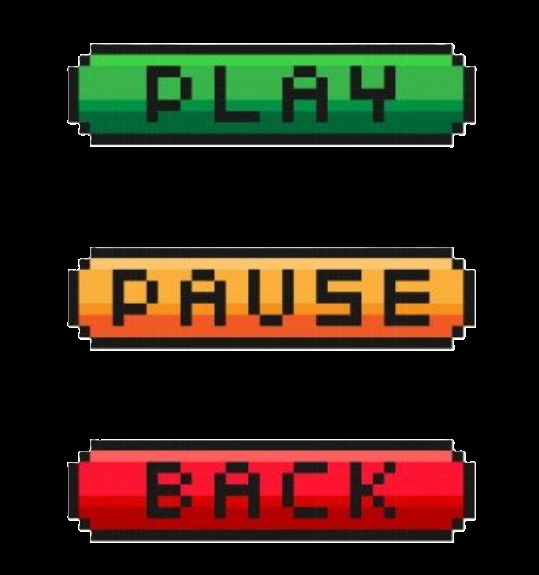
* Spielbrett
* Spielsteine in Rot und Gelb
* Pause-Bildschirm
* Highscore-Anzeige

**Grafiken:**

Spielbrett:

Spielsteine:

Pause:



# Soundeffekte

* Geräusche für das Einsetzen der Steine
* Sieges- und Niederlagen-Sounds
* Optionale Hintergrundmusik

# Erweiterungsmöglichkeiten

* Unterschiedliche Spielfeld-Designs
* Mehrere Spielfeldgrößen
* Power-Ups(z.B. Joker-Stein)
* Anpassbare Spielfeldfarben

# Zeitstrahl

März April Mai Juni

Begrüßungsbildschirm, Spiellogik, Highscore, Optimierung,

Grundfunktionen Grund-KI Animationen, Feinschliff

Test (Erweiterungen)

# Mögliche nächste Schritte

## Entwicklung des Prototyps:

* Beginnen mit der Implementierung der Muss-Kriterien
* Spiellogik und grundlegende Benutzeroberfläche

## KI-Implementierung:

* Entwicklung der KI-Gegner
  + Angefangen mit einfachen Strategien und später evtl. mit komplexeren Algorithmen

## Erweiterungen:

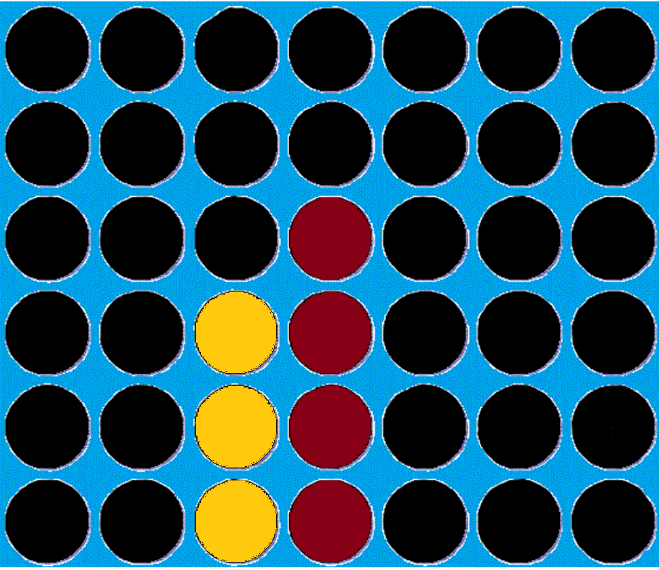
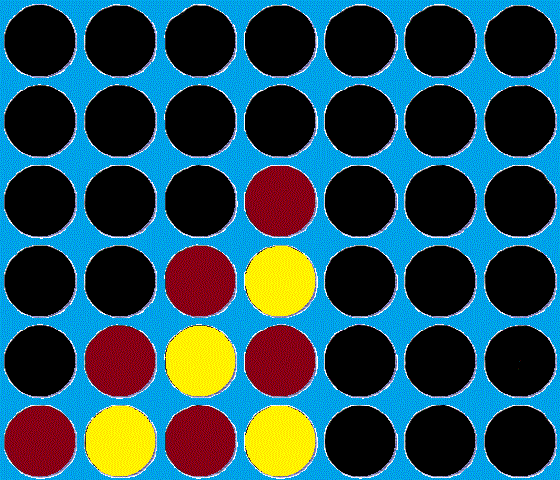
* Nachdem der Kern des Spiels funktioniert, kann man Optionale Features wie Power-Ups und verschiedene Spielfeldgrößen hinzufügen

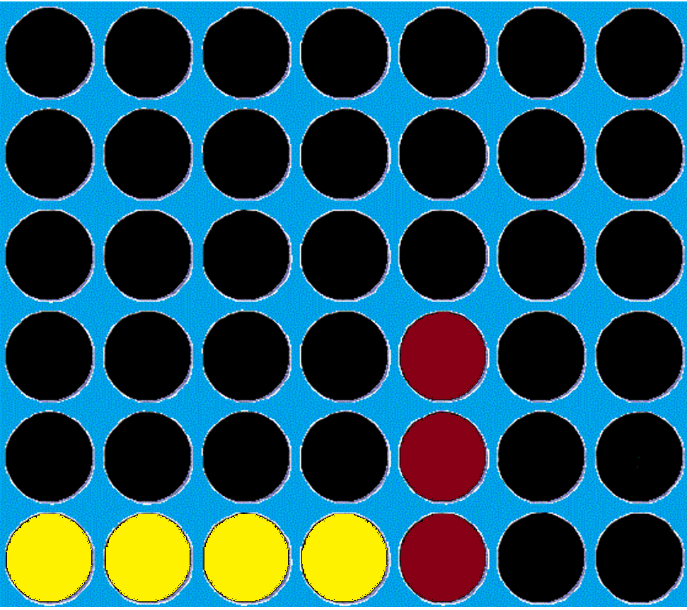
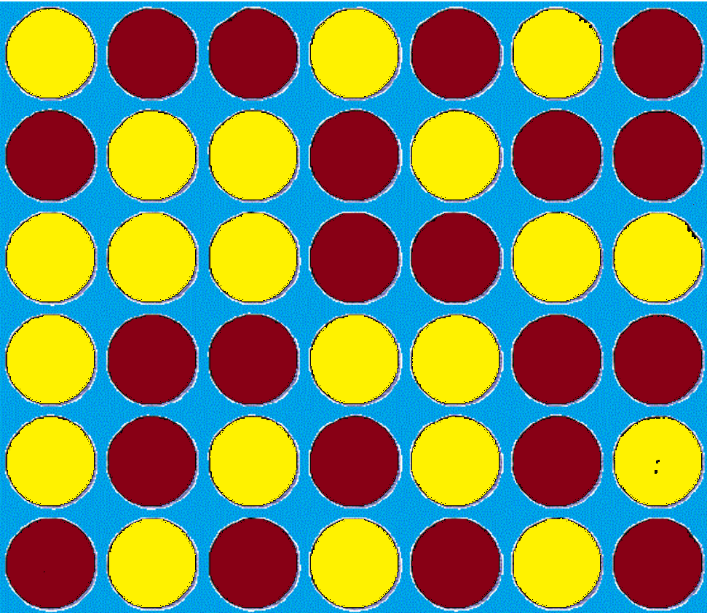
## Testen:

* Mit verschiedenen Personen spielen und Meinung sammeln und ggf. verbessern

# Beispiel Siege oder unentschieden

Diagonal gewonnen (Rot): Senkrecht gewonnen (Rot):



Waagerecht gewonnen (gelb): Unetschieden: