



Dokumentation
Programmierung des Spieles Brick-Breaker

von
Bernhard Meierhöfer

Prüfer



Philipp-Matthäus-Hahn
Technisches Berufskolleg

Inhaltsverzeichnis

Vorwort	4
1 Die Spielidee	5
2 Hintergrundgeschichte	6
3 Fazit	7
4 Programmcode	8
Statement	i
Abbildungsverzeichnis	I
Programmcodeverzeichnis	II

Vorwort

Die vorliegende Dokumentation bietet einen umfassenden Einblick in die Entwicklung des Spiels Brick-Breaker mit Python. Dabei wird die Pygame-Bibliothek verwendet. Pygame ist eine leistungsstarke Python-Bibliothek für die Spieleentwicklung und dient als Grundlage für die Umsetzung dieses Spiels.

Im weiteren Verlauf wird die schrittweise Umsetzung des Spiels Brick-Breaker mithilfe von Pygame detailliert beleuchtet. Der Leser erhält Einblicke in den Code und erfährt, wie die verschiedenen Elemente des Spiels, einschließlich der Plattform, des Balls und der Bricks, implementiert werden.

Die Dokumentation legt besonderen Wert auf bewährte Entwicklungspraktiken und die Lösung häufig auftretender Herausforderungen beim Programmieren von Spielen.

Abschließend wird auf die Anpassungsmöglichkeiten und Potenziale zur Erweiterung des Spiels Brick-Breaker eingegangen. Dies ermöglicht das Verständnis für die Pygame-Bibliothek zu vertiefen.

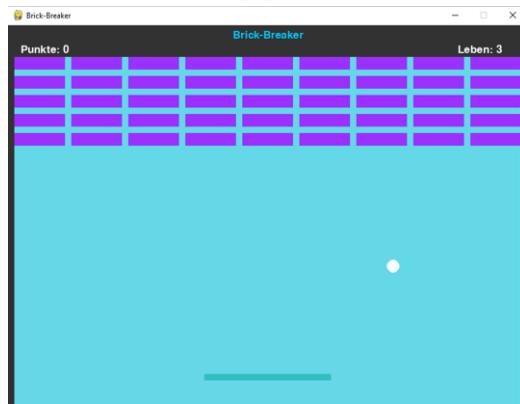


Abbildung 1: Brick-Breaker

1 Die Spielidee

Auswahl

Bei den ersten Überlegungen zur Wahl des zu programmierenden Spiels wurden verschiedene Optionen in Betracht gezogen, darunter Spiele wie Cookie Clicker, Tetris oder Flappy Bird. Bevor die Entscheidung auf Brick Breaker fiel, wurden Kriterien festgelegt, die helfen sollten, ein Spiel mit möglichst wenig Programmierung und Umsetzungsschwierigkeiten zu finden. Dabei sollte das Spiel leicht verständlich für die spielende Person sein und ein einfaches Spielprinzip haben, um eine überschaubare Anzahl von Spielementen zu gewährleisten. Ein weiteres Kriterium war die Leichtigkeit des Codes und die Überschaubarkeit der zu integrierenden Elemente. Aus diesen Gründen bzw. Kriterien wurden Cookie Clicker und Tetris aus der Auswahl genommen, da der Code für das Shopsystem sowie die verschiedenen Blöcke bei Tetris als zu komplex empfunden wurden. Nach sorgfältiger Überlegung fiel die Wahl schließlich auf Brick Breaker, aufgrund seiner vermeintlich einfachen Programmierung und des klaren Spielprinzips sowie Aufbaus, was es als geeignetes Projekt erscheinen ließ.

Ziel des Spiels

Das Ziel des Spiels Brick Breaker ist es, die Blöcke, genannt "Bricks", zu berühren und somit zu zerstören, um Punkte zu erhalten. Sobald alle Blöcke zerstört wurden, gewinnt der Spieler. Dabei muss darauf geachtet werden, dass der Ball nicht unter die Plattform gelangt, da dies zu einem Abzug eines Lebens führt. Nach dem dritten Lebensabzug verliert die spielende Person.

Design

Das Design wurde zunächst schlicht gehalten und im Anschluss an die Programmierung der Spielfunktion angepasst, um ein stimmiges Erscheinungsbild sicherzustellen. Die Farbpalette wurde in kalten Tönen gehalten. Ein anderes Farbkonzept wäre möglich.

2 Hintergrundgeschichte

Breakout

Als Quelle der Inspiration für das Spiel Brick Breaker diente das klassische Arcade-Spiel Breakout, das erstmals von Atari im Jahr 1976 veröffentlicht wurde. Entwickelt von Nolan Bushnell und Steve Bristow, gilt Breakout als Meilenstein in der Geschichte der Videospiele und hat zahlreiche Nachahmer und Neuinterpretationen inspiriert.

Breakout, der Vorgänger von Brick Breaker, erlangte schnell kommerziellen Erfolg und wurde zu einem beliebten Arcade-Spiel in Spielhallen weltweit. Seine einfache, aber fesselnde Spielmechanik und sein innovatives Design machten es zu einem zeitlosen Klassiker, der bis heute von vielen geliebt wird.

Im Laufe der Jahre wurden zahlreiche Versionen und Variationen von Breakout entwickelt, darunter auch die Brick-Breaker-Spiele. Diese erweitern das grundlegende Konzept von Breakout und führen neue Funktionen und Herausforderungen ein. Durch diese Weiterentwicklungen hat Breakout seine Popularität in der modernen Gaming-Welt erhalten und bleibt ein wichtiger Teil der Videogeschichte.



Abbildung 2: Der Vorgänger Breakout

3 Fazit

Erfahrungen und Erkenntnisse

Im Rückblick auf den Entwicklungsprozess kann festgehalten werden, dass das Erstellen und Programmieren des Spiels einen tieferen Einblick der Programmierung und der Python-Sprache sowie der genutzten Python-Programmbibliotheken ermöglicht hat. die Fähigkeiten die während der Programmierung auftretenden Probleme selbstständig zu lösen und nicht aufzugeben, wenn etwas nicht sofort funktioniert, werden durch das Spiel gestärkt. Das Projekt diente als wertvolle Lernerfahrung sowohl für die allgemeine Programmierung als auch für spezifische Funktionen von Python.

Rückblick auf den Entwicklungsprozess

Rückblickend war der Entwicklungsprozess eine lehrreiche Erfahrung, die Die Wichtigkeit einer strukturierten Herangehensweise an das Programmieren sowie die Fähigkeit mit Herausforderungen und Fehlern umzugehen stärkte. Jeder Schritt im Entwicklungsprozess Förderte die zuvorgenannten Kompetenzen und leitstete somit einen direkten und indirekten Beitrag, das Spiel selbst stetig zu optimieren.

Ausblick auf zukünftige Entwicklungsprojekte

Für zukünftige Entwicklungsprojekte, insbesondere für das Spiel Brick Breaker, ist die Implementierung weiterer Spiel-Elemente geplant. Dazu gehören beispielsweise Items, die aus einem zerstörten Block fallen und dem Spieler verschiedene Eigenschaften verleihen können. Des Weiteren sollen ein Menü und ein Restart-Button hinzugefügt werden. Ein Highscore-System sowie die Integration mehrerer Spiel-Level würden dem Spieler zusätzliche Herausforderungen bieten und so für mehr Abwechslung sorgen.

4 Programmcode

Brick Erstellung

Programmcode 1: Brick Erstellung

```
1 bricks = []
2 for row in range(5):
3     for col in range(10):
4         xPosbrick = col * (brick_width + 10) + left
5         yPosbrick = row * (brick_height + 10) + above
6         bricks.append(pg.Rect(xPosbrick, yPosbrick,
7                                brick_width, brick_height))
```

Im Programmcode der ersten Zeile werden eine leere Liste und Blockreihen erstellt. Insgesamt 5 Reihen und 10 Spalten, durch Zeile sechs werden die Blöcke der Liste hinzugefügt.

Ball Bewegung

Programmcode 2: Ball Bewegung

```
1 # Ball Bewegung
2 xposball += xbewegung
3 yposball += ybewegung
4
5 # Ball Grenzen
6 if xposball > 810 - left: # links
7     xbewegung = xbewegung * -1
8
9 if xposball < 30 - right: # rechts
10    xbewegung = xbewegung * -1
11
12 if yposball < 55 + ballradius: # oben
13    ybewegung = ybewegung * -1
14
15 if yposball > 600: # unten
16    ybewegung = ybewegung * -1
17    Leben -= 1
```

Die Abschnitte zeigen die Bewegung des Balls und überprüfen, ob der Ball an den Rand gestoßen ist. Wenn die Koordinaten des Balls einen bestimmten Punkt erreichen, wird die Bewegung umgekehrt.

Punkte und Leben

Programmcode 3: Punkte und Leben

```
1 # Spielende
2 if (Leben <= 0):
3     xposball = 400
4     yposball = 500
5     xPosplatform = (width / 2) - widthplatform / 2
6     yPosplatform = 550
7     screen.blit(status.render('Game_Over', True,
8         (0,0,0)),(225, 300))
9
10 # Spiel Gewonnen
11 if(points == 45):
12     screen.blit(status.render('Gewonnen', True,
13         (0,0,0)),(225, 300))
14     xposball = 400
15     yposball = 500
16     xPosplatform = (width / 2) - widthplatform / 2
17     yPosplatform = 550
```

Diese Abschnitte zeigen, wie das Spielende und der Gewinnzustand erkannt wird. Wenn das Leben kleiner oder gleich Null ist, wird der Ball zurückgesetzt und es wird "Game Over" angezeigt. Wenn alle Blöcke zerstört wurden, wird "Gewonnen" angezeigt.

Statement

Hiermit erkläre ich, dass ich die vorliegende Ausarbeitung selbstständig und ohne unzulässige fremde Hilfe angefertigt habe.

Die verwendeten Literaturquellen sind im Literaturverzeichnis vollständig zitiert.

Datum: 16.5.2023

Name: Bernhard [REDACTED]

Adresse: [REDACTED]

Unterschrift:

Abbildungsverzeichnis

1	Brick-Breaker	4
2	Der Vorgänger Breakout	6

Programmcodeverzeichnis

1	Brick Erstellung	8
2	Ball Bewegung.....	8
3	Punkte und Leben.....	9