

GravityPong

Technische Dokumentation

Felix Gold

GD1018 24.01.2020

Inhaltsverzeichnis

1.0)	Einführung	2
2.0)	UML Diagramm.....	2
3.0)	CVector2	3
3.1)	GetDistance.....	3
3.2)	Length	3
3.3)	Normalize.....	3
3.4)	Rotate	3
4.0)	CPlayer	3
4.1)	Update.....	3
4.2)	Render	3
5.0)	CBall	3
5.1)	Update.....	3
5.2)	Render	3
6.0)	GameManager	4
6.1)	Update.....	4
6.2)	DisplayScore	4
6.3)	ToggleGameState	4
6.4)	WatchBall.....	4
6.5)	ResetBoard	4
6.6)	ServeBall	4

1.0) Einführung

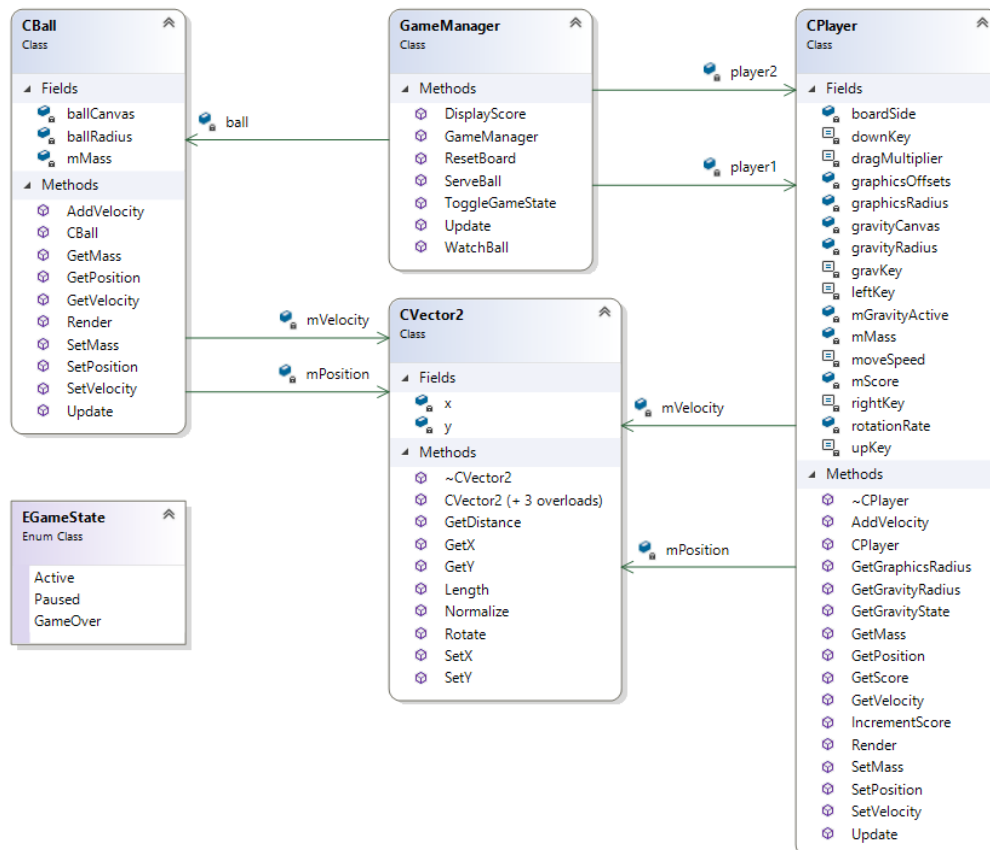
GravityPong ist wie sich bereits am Name ablesen lässt eine abgewandelte Version des Spiels 'Pong', welches auf Gravitation basiert. Hierbei spielt jeder Spieler ein schwarzes Loch, symbolisiert doch einen rotierenden Stern (*), mit der Fähigkeit, den Ball innerhalb eines bestimmten Radius, dargestellt durch einen grauen Kreis, umzulenken. Die Umlenkung des Balls durch Gravitation wird per Tastendruck aktiviert und deaktiviert, um den Ball leichter zu kontrollieren. Sollte ein Ball den Stern eines Spielers berühren, wird der Ball vom schwarzen Loch eingesaugt und als ein Eigentor gewertet. Ziel des Spiels ist es, 5 Punkte zu erreichen.

Steuerung:

Spieler 1:

Spieler 1	Spieler 2	Funktion
W	I	Bewegung nach oben
A	J	Bewegung nach links
S	K	Bewegung nach unten
D	L	Bewegung nach rechts
SPACE	RSHIFT	Aktivierung der Gravitation
ESC	ESC	Pausieren/Fortsetzen

2.0) UML Diagramm



Im Folgenden werden die Klassen sowie deren wichtigste Funktionen genauer erläutert. Hierbei werden Getter, Setter und Konstruktoren jedoch nicht mit angesprochen.

3.0) CVector2

CVector2 ist eine Utility-Klasse zur Rechnung mit Vektoren. Sie wird für Dinge wie Positionsberechnung, Geschwindigkeitsberechnung, Richtungsangaben und Ähnliche verwendet.

3.1) GetDistance

Die Funktion GetDistance wird von einem CVector2 aus aufgerufen und erwartet einen CVector2 als Input. Hiermit wird die Distanz zwischen 2 Positionen, z.B. einem schwarzen Loch und dem Ball berechnet. Dies geschieht durch Benutzung der Funktion Length.

3.2) Length

Length berechnet die Länge eines Vektors, indem die Wurzel aus der Summe der Quadrierten Werte des Vektors gezogen wird (Betrag des Vektors)

3.3) Normalize

Die Funktion Normalize normalisiert einen Vektor. Hierbei behält der Vektor seine Richtung bei, seine Länge wird jedoch auf 1 gesetzt.

3.4) Rotate

Die Funktion Rotate dreht einen Vektor. Dies wird z.B. für den drehenden Stern des Spielers benutzt

4.0) CPlayer

4.1) Update

Die Update-Funktion des Spielers verwaltet Dinge wie die Bewegung und die Gravitation des Spielers

4.2) Render

Die Render-Funktion des Spielers rotiert den Stern des Spielers und zeigt diesen an. Sie zeigt außerdem den Kreis an, welcher den Gravitationseinfluss des Spielers symbolisiert.

5.0) CBall

5.1) Update

Die Update-Funktion des Balls verwaltet Dinge wie die Bewegung und den Einfluss der Gravitation auf den Ball.

5.2) Render

Die Funktion Render zeigt den Kreis an, welcher den Ball symbolisiert.

6.0) GameManager

Der GameManager ist eine Utility-Funktion welche zentral verschiedene Aspekte des Spiels verwaltet.

6.1) Update

Die Update Funktion des GameManagers verwaltet die Pausier-Funktion und beendet das Spiel sobald ein Spieler 5 Punkte erreicht.

6.2) DisplayScore

DisplayScore zeigt den aktuellen Punktestand eines Spielers an.

6.3) ToggleGameState

ToggleGameState wechselt zwischen den GameStates Active und Paused.

6.4) WatchBall

WatchBall setzt das Spielbrett auf den Ausgangszustand zurück wenn ein Tor geschossen wird indem die Funktion ResetBoard aufgerufen wird. Außerdem setzt WatchBall das Spielbrett zurück falls sich der Ball oberhalb des Spielfelds befindet (Hierhin wird der Ball versetzt falls er den Stern eines Spielers berührt). Danach wird die Funktion ServeBall aufgerufen

6.5) ResetBoard

ResetBoard Setzt beide Spieler sowie den Ball auf die Ausgangapositionen zurück und setzt alle Geschwindigkeiten auf 0.

6.6) ServeBall

ServeBall bestimmt eine zufällige Richtung zwischen 0 – 360° und “schießt” den Ball anschließend in diese Richtung