17. Dezember 2018

SHEEPHARD

CODE DOKUMENTATION

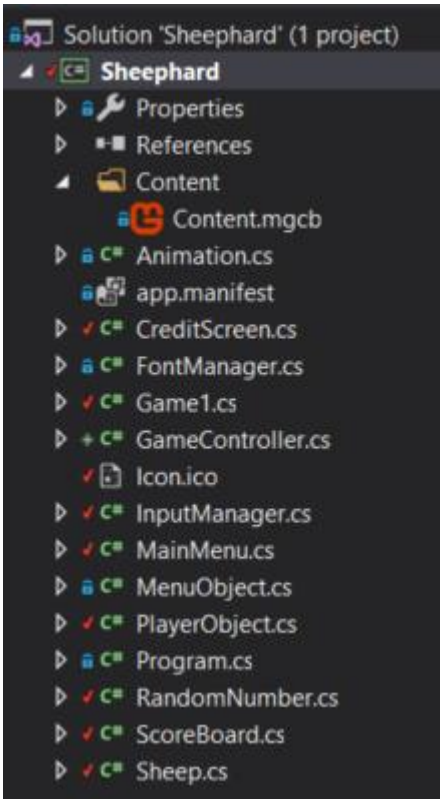
JAN MIKUSCH

FH SALZBURG – MULTIMEDIA TECHNOLOGY

Multimedia Projekt (1)

Jan Mikusch
FH Salzburg – Multimedia Technology

KLASSEN



Class Game1

Game1 ist beim Programmstart für alle Inizialisierungen und dem Laden der Assets zuständig. Die Funktionen Update() und Draw() werden wärend der Laufzeit pro Frame aufgerufen. Diese zwei Methoden rufen die Update bzw. Draw Methode des GameControllers auf.

GameController

Der GameController ist die Zentrale Klasse des Projekts, die Game-Übergreifende Methoden und Variablen besitzt.

Sie speichert die Maximale und derzeitige Rundenanzahl, den Ingame-Countdown, die Spieler ID des zuletzt ausgewählten Schwarzen Schafes, und Referenzen zu den jeweiligen Spielern. Außerdem wird hier der derzeitige Spielstatus (Main Menu, Playing, GameEnd etc) gespeichert.

METHODEN:

Init(): Wird beim Start des Spiels von der Game1 aufgerufen und setzt Variablen auf gewisse Default-Werte.

Update(): Hier wird überprüft ob das Spiel beendet werden soll (oder zum Hauptmenü gesprungen werden soll) und das Ein/Ausschalten des FullScreens wird ermöglicht. Außerdem Deligiert die Funktion je nach SpielStatus an verschiedene Update-Funktionen in anderen Klassen.

Draw(): Zeichnet den Background und Deligiert an andere Draw-Methoden (je nach Game-State)

PlayMusic(): Startet die Hintergrund Musik.

StartGame(): Hier wird der Spielestatus auf Play gesetzt. Die Player Objekte und NPCs werden erzeugt die StartRound() Methode wird aufgerufen.

StartRound(): Erzeugt für jeden Spieler ein neues Schaf-Objekt und Positioniert es mithilfe der SpawnPosition()

SpawnPosition(): Findet eine nicht besetzt Position zum Spawnen der Schafe.

BlackSheepNext(): Berechnet, welcher Spieler das nächste schwarze Schaf ist.

CheckSelected(): Überprüft ob sich vor dem Schwarzen Schaf (Blickrichtung) ein Spieler befindet, der dann als SelectedPlayer gesetzt wird.

ChooseSelected(): Beendet die Runde falls ein Spieler Selectiert wurde.

EndRound(): Verteilt die Punkte am Ende der Runde und setzt den GameState auf Waiting

InputManager

Besitzt Konstanten für die Eingabe über das Keyboard und das GamePad und dient zur vereinfachung der Input-Überprüfung.

Es gibt Methoden für das setzen die States der GamePads und des Keyboards. Diese werden Am Anfang bzw. am Ende in der Update Funktion des GameControllers aufgerufen.

Die restlichen Methoden überprüfen, ob bestimmte Tasten gedrück worden sind (beispielsweiße wird im Hauptmenü überprüft, ob der Stick, das DPAD oder die Pfeiltasten auf dem Keyboard nach oben oder unten gedrückt wurden, und durch das Menü zu navigieren).

Random Number

Der Code wurde von (https://scottlilly.com/create-better-random-numbers-in-c/) für bessere zufällige zahlen verwendet und wurde 1zu1 adaptiert. Die Methode Between() erzeugt einen zufälligen int-Wert zwischen den angegebenen Werten

FontManager

Speichert Referenzen für geladene Fonts

MainMenu

Ist für das Hauptmenü zuständig. Besitzt Referenzen für die jeweiligen Bilder, Schafe die am Screen gezeichnet werden sollen.

WICHTIGE FUNKTIONEN:

Update(): Mithilfe des InputManagers wird die Navigation durch das Menü möglich. Durch Wählen von Start wird im GameController das Spiel gestartet. Man kann zu Credits-Screen gelangen, auswählen wie viele Spieler spielen und wie viele Runden gespielt werden. Beim Starten wird außerdem überprüft, ob genügend Controller Connected sind, ansonsten wird eine Meldung ausgegeben.

Draw(): Zeichnet das Menü, die HowTo Texture und die Schafe im Hauptmenü

MenuObject

Beinhaltet Informationen zum anzuzeigenden Text und welche Art von Menü Objekt es ist. (zB für Play oder Credits Menüpunkt)

CreditScreen

Zeichnet die Credits auf dem Bildschirm und kontrolliert, ob der Spieler zum Hauptmenü zurück möchte

ScoreBoard

Zeichnet das kleine ScoreBoard am rechten oberen Rand sowie die übrige Zeit. Wenn das Spiel beendet ist, wird ein großes „Finales“ ScoreBoard gezeichnet.

Animation

Hält in einem Dictionary alle SpriteSheets der Schafe.

WICHTIGE FUNKTION:

DrawSheep(): Zeichnet die Schafe mithilfe der mitgegebenen Attribute. Mithilfe des Y-Werts der Position wird der Z-Index berechnet.

UNTERKLASSE SPRITESHEET

Beinhaltet die Texturen, die Anzahl der Frames, der Rows und der Colums.

PlayerObjekt

Eine abstrakte Klasse für NPCs und Player

WICHTIGE METHODEN

GetRectangle(): Berechnet ein Rectangle, das den Collider des Objekts representiert.

Update(): Ruft die Update Funktion des Sheep-Objektes auf. Es wird die GetMovement Methode aufgerufen, welche bei der vererbung Überschrieben wird Es wird die CollisionDetechtion() Methode aufgrufen und die neue Position des Objektes gesetzt. In weiterer folge wird auch der AnimationsTyp gesetzt.

Draw(): Delegiert an Sheep.Draw()

CollisionDetection(): Überprüft, ob der Character in der Richtung, in die er sich bewegen möchte, an Grenzen stößt. Wenn das passiert, wird die Bewegung in diese Richtung nicht ausgeführt.

CollisionRight/Top/Left/Botton() Es wird kontrolliert, ob der Spieler versucht in einen anderen Spieler oder über den Rand des Spielfeldes zu laufen.

NPC

Erbt von PlayerObject

WICHTIGE METHODEN

GetNpcInput(): Es wird X ms gewartet. Dann wird eine neue zufällige Stance berechnet die der NPC ausführen soll.

CreateMovement(): Erzeugt den Movement-Vektor für den NPC.

Player

Erbt von PlayerObject

WICHTIGE METHODEN

Getnput(): Überprüft die Eingabe setzt dadurch den State (Moving, Running, Eating, …). Bei Moving oder Running wird die relative Bewegung im derzeitigen Frame berechnet.

Sheep

Berechnet in der Update() den Frame, der angezeigt werden soll. In der Draw() Methode wird an die DrawSheep Methode der Animation delegiert.

Die Play() Methode setzt eine neue Animation, sofern sie sich von der derzeitigen unterscheidet.