

Assignment 5: Game programming

Robert Slob: 5496578
Zino Onomiwo: 4221095

Inleiding

Dit is de documentatie met daarin wat wij allemaal veranderd hebben aan de begincode. We hebben het spel geïmplementeerd zoals dat hoorde, maar hebben ook een paar extra dingen toegevoegd, bijvoorbeeld verschillend soort gedrag van de enemies. Aangezien onze functies uit de naamgeving erg voor zich spreken, behandelen we ze alleen in dit document en hebben we geen commentaar verder in de code gezet.

Voor het laden van png bestanden hebben we de library: [Gloss-Juicy](#) gebruikt.

We hebben ervoor gekozen om de data van verschillende data types niet in de Model file te zetten, maar in aparte files. Dit maakte het makkelijker voor ons om te vinden waar wat stond. We hebben er uiteraard voor gezorgd dat deze files alleen data en een paar functies bevatten, maar niet de modules importeren zoals: Time, Model of View.

We zullen per module behandelen wat er veranderd is.

Main

We hebben 3 sprites ingeladen met behulp van gloss-juicy in de main functie, deze sprites hebben we als parameter mee gegeven aan de initial functie.

Event, Config & Controller

Hierin is niks veranderd.

Player

We hebben een Player-module gemaakt, met hierin het datatype voor de speler. Dit type bestaat natuurlijk uit een locatie, snelheidsvector en richting, waarmee aangegeven wordt in welke richting de speler momenteel gedraaid is. Verder hebben we hier ook nog de spelersafbeelding aan toegevoegd en een shootTimer plus een bool die aangeeft of je mag shooten. Deze laatste twee zijn er zodat je tactisch je kogels moet gebruiken. We hebben ervoor gekozen om de sprite van de player in de Player data te zetten omdat er maar één speler is. We hebben ook de de Locatie klasse geïmplementeerd in de speler.

Model

We hebben aan het datatype World een aantal extra velden toegevoegd:

player, starField, particles, bullets, multipliers, enemies, enemySpawnTimer, multiplierTimer, enemyImage, multiplierImage, playerScore, scoreMultiplier. We hebben World de sprites laten opslaan zodat elk afzonderlijk object niet zijn eigen sprite heeft, en memory in beslag neemt.

De namen spreken allemaal erg voor zich.

In de functie initial maken we onze initiële (spel)wereld aan, ook hebben we extra parameters toegevoegd voor de sprites. In de functie emptyworld maken we onze huidige wereld leeg, zodat het weer net is alsof je een nieuw spel start.

View

Hierin staan de standaard horizontale en verticale resoluties. Verder is hier de functie die van onze World (samen met de resolutie) de bijpassende afbeelding tekent. Dit doet die grotendeels door voor

alle velden uit World de individuele drawfuncties aan te roepen en hij voegt deze afbeeldingen samen. Verder staan hier nog de draw functies van Star, Player en van een string in.

Enemies

In de enemies module hebben we de data van Enemy geplaatst. Een opvallend veld is de updateEnemy. Deze bepaald het gedrag van de vijand. Op deze manier kan elke vijand zich afzonderlijk gedragen, we hebben twee voorbeeld gedragen geïmplementeerd, namelijk: simpleBehaviour, wat de vijand de speler maakt volgen; en jumperBehaviour, wat maakt dat de vijand de speler bespringt.

De separation functie maakt dat de enemies niet in elkaar komen te zitten door ze uit elkaar te laten vliegen.

Bullets

Hier staat het datatype van een Bullet in, namelijk de locatie, de snelheid en de richting. We hebben verder een functie gemaakt die één Bullet tekent en een functie die een lijst van Bullets tekent. Verder staat hier ook de functie in die de bullets laat bewegen (aangeropen per tijdframe). Als laatste staan hier functies in die er rekening mee houden dat als een Bullet buiten de wereld komt, hij niet meer opgeslagen hoeft te worden.

Multiplier

Aangezien multipliers niet bewegen, staat in het datatype hiervan alleen de locatie van de multiplier. Verder staat in deze module een functie om een multiplier te tekenen en een functie om een lijst van multipliers te tekenen. Als laatste een functie die een Multiplier op een random plek laat spawnen.

Stars

Een ster bestaat uit een locatie en de diepte van de ster. Hier staat verder een functie die een ster op een random plek laat spawnen en een functie die het hele veld aan sterren genereert.

Time

In de timeHandler functie word de nieuwe wereld berekent. We hebben deze functies probeert te splitsen in zoveel mogelijke stappen. Hier worden kogels, vijanden en particles gecreëerd en bijgehouden.

Location

Hier wordt een klasse Location gemaakt, deze klasse is gebruikt voor het generaliseren voor de botsingen tussen spelers, kogels, vijanden en multipliers.

Particles

In de datatype van de Particle staat zijn locatie, snelheid, leeftijd en hoe het getekend wordt. In de randomizedParticle wordt een al bestaande particle willekeurig gemaakt, zodat niet alle particles hetzelfde zijn.

RandomUtils

Hier hebben we de functies geplaatst voor willekeurigheid.