

UNITY INPUT SYSTEEM

Jordy Teunis

Inhoudsopgave

Inleiding.....	1
Methodes	1
Uitvoering.....	2
1. Onderzoek hoe de nieuwe input systeem werkt	2
2. Input systeem in Unity	3
Stap 1: Input systeem downloaden	3
Stap 2: Input systeem toevoegen aan het project	3
Stap 3: Action map toevoegen en invullen.....	4
3. Bewegen doormiddel van het input systeem	4
Validatie.....	6
Reflectie.....	6
Bronnen.....	6

Inleiding

Het doel van de nieuwe input systeem is dat het overzichtelijk te lezen is in de code, dat het makkelijk aan te passen is en dat het makkelijk te begrijpen is voor andere teamleden. De nieuwe input systeem is een nieuwe feature in Unity die erg handig is om de input van de speler op te vangen, daarnaast is het ook erg makkelijk aan te passen door de input interface die automatisch wordt geschreven door Unity. Om dit systeem goed onder de knie te krijgen heb ik het onderzocht en mee geoefend en een beweegscript gemaakt, zodat het een snel en handige tool wordt om te gebruiken.

Methodes

1. Analyseren: Onderzoek hoe de nieuwe input systeem werkt
 - Type: Literatuur- en audiovisuele onderzoek
 - Beschrijving: Onderzoeken hoe de input werkt, welke vereisten er voor zijn en op welke manier het kan worden geïmplementeerd in een Unity project.
 - Bronnen: Unity Documentation en YouTube
2. Ontwerpen: Input systeem in Unity
 - Type: Input systeem toevoegen aan een Unity project
 - Beschrijving: Het Input systeem integreren in een Unity project zodat er een bewegen script gemaakt kan worden voor de speler
3. Realiseren: Bewegen doormiddel van het input systeem
 - Type: Project met werkend input system
 - Beschrijving: script toevoegen zodat de speler kan bewegen doormiddel van het nieuwe input system.

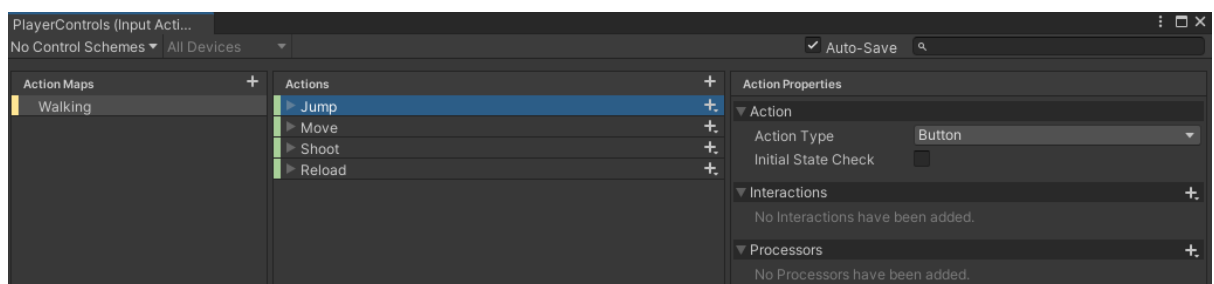
Uitvoering

1. Onderzoek hoe de nieuwe input systeem werkt

In de informatievevideo van Dman en in de Unity Documentation zijn er een aantal dingen nodig om het input systeem te gebruiken. Als je het input systeem wilt gebruiken moet je het eerst downloaden doormiddel van de Unity Package manager, dan wordt het oude systeem automatisch inactief gezet.

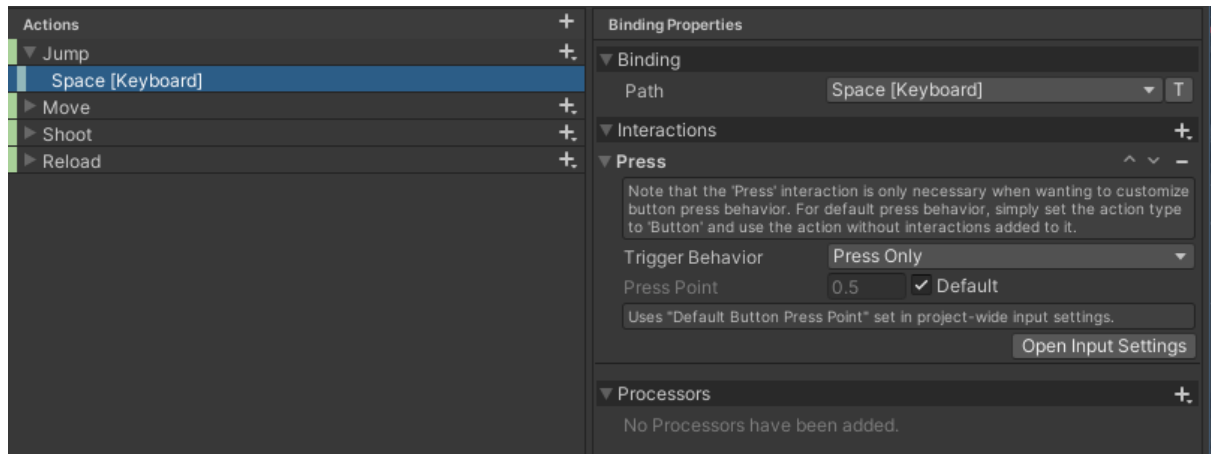
In het nieuwe input systeem zijn er verschillende action maps die je kan gebruiken voor verschillende staten waarin de speler zich bevindt. Als de speler bijvoorbeeld aan het zwemmen is kan je een andere action map gebruiken dan wanneer de speler aan het lopen is. In de action maps moet je actions aanmaken die je verschillende eigenschappen kan geven. Een action is vaak een specifiek event dat een bepaalde interactie of input van de speler afhandelt, zoals een sprong, aanval, of beweging.

In de action geef je de specifieke eigenschap voor het event. Hierin kan je de input van de "W" toets op een bepaalde action zetten. De "W" kan verschillende types hebben, het kan bijvoorbeeld een button waarde hebben waarmee je een input ophaalt wanneer die toets is ingedrukt. Of het kan een value waarde hebben, hierbij worden er waarden opgehaald en verstuurd wanneer er op de toets wordt gedrukt.



Figuur 1: Unity input systeem.

In een action kan je een binding property aan maken die je aan een toets moet vastmaken zodat de action weet bij welke toetsen hij beheert. Hierin kan je de toets een interactie geven die aangeeft wanneer deze wordt geactiveerd, bijvoorbeeld “Press Only”, deze interactie wordt geactiveerd wanneer er één keer op de toets wordt gedrukt. “Hold” wordt pas geactiveerd wanneer de toets een bepaalde tijd wordt ingedrukt, dit is handig voor bijvoorbeeld speciale aanvallen waarbij je een langere tijd dezelfde toets moet indrukken om de aanval te activeren.



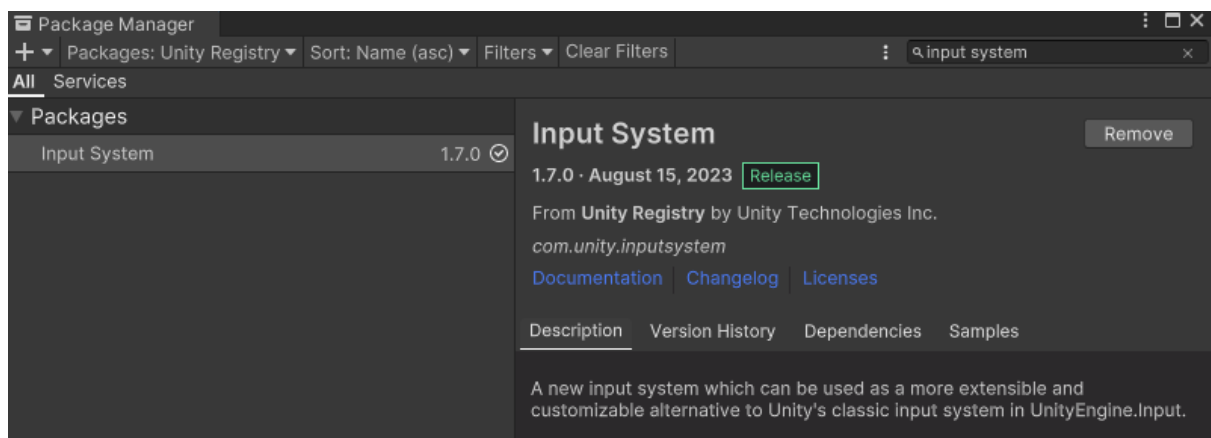
figuur 2: Unity input systeem binding properties.

2. Input systeem in Unity

Stap 1: Input systeem downloaden

Om het input systeem toe te voegen moet er als eerst het Input systeem gedownload worden doormiddel van de Unity Package manager.

- Zet de Package manager op Unity Registry, zoek in de taakbalk op input system en druk op download.



Figuur 3: Unity Package manager.

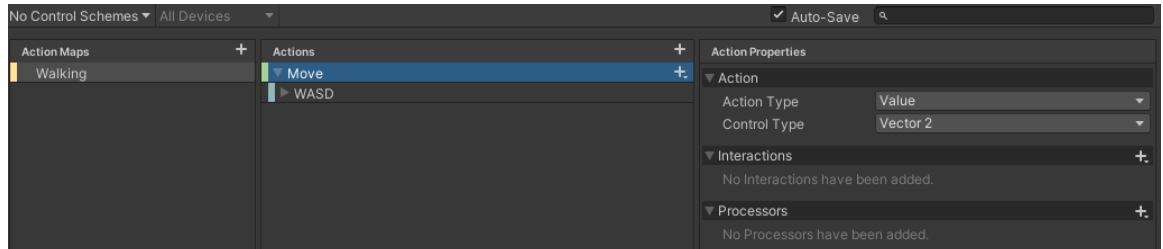
Stap 2: Input systeem toevoegen aan het project

- Nu de nieuwe input system is gedownload kan je het aanmaken door een map aan te maken die de naam “Input” krijgt. Hierin kan je door de rechtermuisknop in te drukken en op “create” te klikken en daarna op “Input Actions”. Geef je input action de naam “PlayerControls”.



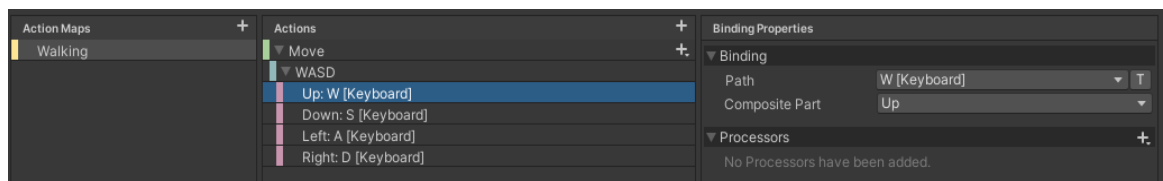
Stap 3: Action map toevoegen en invullen

- Open de PlayerControls input systeem en maak een nieuwe in action map aan die de naam "Walking" krijgt. In Walking maak een Action die de naam "Move" krijgt. Deze action roepen we aan wanneer we willen bewegen. Verander de action type in "Action Properties" naar "Value" en zet de control type op "Vector2". Maak een "Up/Down/Left/Right Composite" Geef deze de naam "WASD" zodat duidelijk is welke toetsen worden gebruikt.



Figuur 4: Unity input systeem met variables.

- Als laatste moeten we de toetsen aan maken in "WASD". Open "WASD" en selecteer "Up <No Binding>" stel bij Path "W" in. Dit is de gewilde toets om vooruit mee te lopen. Stel voor "Down" "S", "Left" "A" en "Right" "D" in. Vergeet niet om de Input op te slaan door op "CTRL S" te klikken of rechts boven Auto-Save aan te zetten



Figuur 5: Unity input systeem met "WASD" variables.

3. Bewegen doormiddel van het input systeem

1. Maak in de Playercontroller een nieuwe "playerInput"-variable deze zorgt ervoor dat de juiste Input wordt gebruikt en een private Vector 2 "moveInput" deze zorgt ervoor dat de Input wordt opgeslagen in een variable.

```
1 public class PlayerController : MonoBehaviour
2 {
3     1 reference
4     public PlayerInput playerInput;
5     3 references
6     private Vector2 moveInput;
```

2. Maak een Awake-methode aan. Deze methode zorgt ervoor dat een nieuwe PlayerInput-instantie wordt gemaakt wanneer het script wordt geïnitieerd.

```
6 private void Awake()
7 {
8     playerInput = new PlayerInput();
9 }
```

3. Maak een FixedUpdate-methode aan. Deze methode die elke “physics frame” wordt aangeroepen.

```
11  void FixedUpdate()  
12  {  
13      MovePlayer();  
14  }
```

4. Maak een MovePlayer functie aan. Hierin staat het script dat de speler daadwerkelijk kan laten bewegen.

```
15  private void MovePlayer()  
16  {  
17      moveDirection = orientation.forward * moveInput.y + orientation.right * moveInput.x;  
18      playerRigidbody.AddForce(moveDirection.normalized * 6 * 10f, ForceMode.Force);  
19  }
```

5. Maak de OnMove functie aan. Deze functie kijkt naar de Move action in de Walking Action map.

```
21  private void OnMove(InputValue value)  
22  {  
23      moveInput = value.Get<Vector2>();  
24  }  
25  }
```

Validatie

Ik heb de code laten reviewen door een teamlid die erop wees dat alles werkte naar wens en dat de speler zich kon bewegen doormiddel van de WASD-toetsen.

Reflectie

Ik heb geleerd hoe ik het nieuwe input systeem gebruik en integreer in een nieuw project, door het nieuwe systeem te downloaden via de package manager en daarna toe te voegen in een folder. Deze input daarna in te vullen en te linken met een script.

Ik zou het vervolg niet anders aanpakken dan hoe ik het nu heb gedaan, want ik heb duidelijk geleerd wat de nieuwe input systeem inhoud en ik kan het toepassen op een project.

Bronnen

DmanGames. (2023, 26 juni). *Unity's new Input System: The Definitive guide* [Video]. YouTube. <https://www.youtube.com/watch?v=ZSP3bFaZm-o>

Input System | Input System | 1.11.2. (z.d.). <https://docs.unity3d.com/Packages/com.unity.inputsystem@1.11/manual/index.html>