|  |
| --- |
|  |
| 센서 및 엑추에이터 활용 실습 |
|  |
| 지능형 인터페이스 |

**교수 : 김상연**

3조

조장 : 김영규

팀원 : 이재영

팀원 : 여창민

센서 및 엑추에이터 활용 실습

지능형 인터페이스

1. 과제 설명
2. 스크립트 코드(주석 포함 및 소스 코드 설명)
3. 고찰
4. 구동 동영상 별첨
5. 과제 설명

이번 과제는 센서 및 엑추에이터 활용 실습이다. 유니티의 에셋을 조이스틱을 제외한 센서와 엑추에이터를 활용하여 자유롭게 구성하는 과제다. 우리는 에셋의 Car 예제를 활용하여 초음파 센서의 값에 따라 자동차를 움직이게 구성하였다.

Car 예제는 키보드의 입력W, A, S, D에 따라 자동차 전, 후 , 좌, 우로 움직인다. 이 Car 예제에 키보드가 아닌 초음파 센서의 값에 따라 거리가 멀면 차가 앞으로 전진하고, 거리가 적당한 중간 값이면 입력이 없도록, 거리가 가까우면 후진하도록 구현하였다. 후진하는 거리는 10이하이고, 입력이 없는 범위는 20이하, 그 이상은 전진하도록 구현하였다.

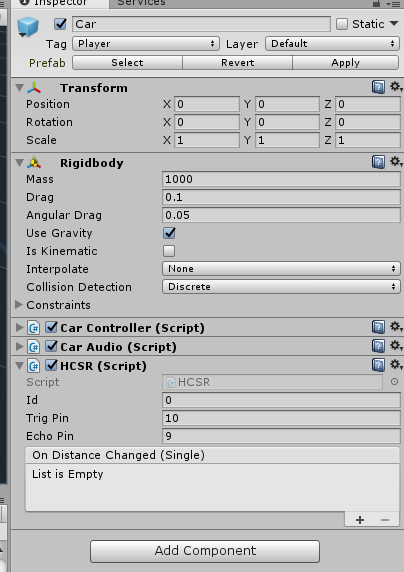
1. 스크립트 코드

HCSR.cs

|  |
| --- |
| namespace Ardunity  {  [AddComponentMenu("ARDUnity/Controller/Basic/Ultrasonic")]  [RequireComponent(typeof(CarController))]  [HelpURL("https://sites.google.com/site/ardunitydoc/references/controller/hcsr04/hcsr04-script")]  public class HCSR : ArdunityController, IWireInput<float>  {  public int trigPin = 10;  public int echoPin = 9;  public static int measuredDistance = 0;  public FloatEvent OnDistanceChanged;  private FLOAT32 \_value = 0f;  //자동차 객체 선언  private CarController m\_Car; // the car controller we want to use  // 차를 움직이는 변수들  private float h = 0;  private float v = 0;  #if !MOBILE\_INPUT  private float handbrake = 0f;  protected override void Awake()  {  // CarController를 받아 m\_Car에 연결시켜준다.  m\_Car = GetComponent<CarController>();  }  protected override void OnExecuted()  {  if (OnWireInputChanged != null)  OnWireInputChanged(\_value);  OnDistanceChanged.Invoke(\_value);  }  protected override void OnPop()  {  FLOAT32 newValue = \_value;  Pop(ref newValue);  if (newValue != \_value)  {  \_value = newValue;  updated = true;  }  }  public override string GetCodeDeclaration()  {  return string.Format("{0} {1}({2:d}, T{3:d});", this.GetType().Name, GetCodeVariable(), id, trigPin);  }  public override string GetCodeVariable()  {  return string.Format("hcsr{0:d}", id);  }  //distacne 부분 수정  public float distance  {  get  {  return (float)\_value;  }  }  void Update()  {  measuredDistance = (int)distance;  v = distance;  //거리Distance에 따라 자동차의 움직임을 조절한다.  if(measuredDistance < 10) { v = -10; }  else if (measuredDistance < 20) { v = 0; }  else { v = 10; }  Debug.Log("Distance : " + distance.ToString() + "cm");  }  //FixedUpdate함수에서 자동차를 조정된 값에 따라 움직이도록 한다.  private void FixedUpdate()  {  m\_Car.Move(h, v, v, handbrake);  #else  m\_Car.Move(h, v, v, 0f);  #endif  }  }  } |

초음파 센서를 사용하여 자동차를 움직이도록 하였다.

Car 예제를 사용하였으며 Car예제의 Car에 들어있는 CarUserControl을 초음파 센서 Reactor인 HCSR로 수정하였다.



초음파 센서의 거리에 따라 차가 전진, 후진, 입력 없음의 3가지 상태로 변화하도록 코드를 구성하였다.

1. 고찰

이번 과제는 지난 주에 실습했던 조이스틱을 제외하고 지금까지 활용했던 센서나 엑추에이터를 활용하는 실습이었다. 우선 과제를 해결하기 위해 여러 에셋을 실행해보고 아이디어 회의를 하였다. 많은 아이디어가 있었지만 자율주행 자동차에 영감을 얻어 자동차를 움직이도록 하였다. 초음파 센서는 예전 실습에서 활용하였기 때문에 쉽게 구현할 수 있었고 지난 주의 실습에서 person과 비슷한 방법으로 CarUserControl을 초음파 센서 Reactor인 HCSR로 수정하여 구현하였다. 센서가 부족하여 좌, 우는 구현하지 못했지만 센서가 충분했다면 자동차 예제도 더 완벽하게 구현할 수 있을 것 같다.