

# 가위바위보 기능 추가하기

- 지난 시간에 완성한 캐릭터를 가져온다.
- 캐릭터에 가위바위보 기능을 추가한다.

# 가위바위보 만들기

- Rock 스크립트를 만들어 캐릭터에 연결한다.

```
bool flgRock = false;
```

```
void OnGUI ( ) { if (flgRock == false) {  
    if (GUI.Button(new Rect(10, Screen.height - 110, 100, 100),  
        "가위바위보")) { flgRock = true; } } }
```

# 게임 실행하기



# 가위바위보 상태 관리하기

- 변수로 상태 전환하기
  - 0. 가위바위보 시작
  - 1. 플레이어의 입력 대기
  - 2. 판정
  - 3. 결과

# Rock 스크립트 수정하기

- 다음의 코드를 Update 구문에 추가한다.

```
int modeRock = 0;
```

```
void Update ( ) { if (flgRock == true) { switch (modeRock) {  
    case 0: //가위바위보 시작 break;  
    case 1: //플레이어의 입력을 기다린다 break;  
    case 2: //판정 break;   case 3: //결과 break; } }
```

# 가위바위보 음성 및 상태 준비하기

- 다음의 구문을 Rock 스크립트에 추가한다.

```
public AudioClip voice_rock_start;    public AudioClip voice_rock_loose;
```

```
public AudioClip voice_rock_pon;    public AudioClip voice_rock_draw;
```

```
public AudioClip voice_rock_goo;    const int ROCK = 0;
```

```
public AudioClip voice_rock_choki;    const int GOO = 1;
```

```
public AudioClip voice_rock_par;    const int CHOKI = 2;
```

```
public AudioClip voice_rock_win;    const int PAR = 3;
```

# 가위바위보 음성 및 상태 준비하기

- 다음의 구문을 Rock 스크립트에 추가한다.

```
const int DRAW = 4;           int myHand;
const int WIN = 5;            int unityHand;
const int LOOSE = 6;          int flgResult;
private Animator animator;     float waitTime;
private AudioSource univoice;
```

# 가위바위보 음성 및 상태 준비하기

- 다음의 구문을 Rock 스크립트에 추가한다.

```
void Start ( ) {  
    animator = GetComponent<Animator>( );  
    univoice = GetComponent<AudioSource>( );    }
```



# 가위바위보 액션을 함수로 정의하기

- 다음의 구문을 Rock 스크립트에 추가한다.

```
void UnityRockAction (int action) {  
    switch (action) {  
        case ROCK:  
            animator.SetBool("Rock", true);  
            univoice.clip = voice_rock_start;  
            break;
```

# 가위바위보 액션을 함수로 정의하기

- 다음의 구문을 Rock 스크립트에 추가한다.

case GOO:

```
    animator.SetBool("Goo", true);
```

```
    univoice.clip = voice_rock_goo;
```

```
    break;
```

case CHOKI:

```
    animator.SetBool("Choki", true);
```

```
    univoice.clip = voice_rock_choki;
```

```
    break;
```

# 가위바위보 액션을 함수로 정의하기

- 다음의 구문을 Rock 스크립트에 추가한다.

case PAR:

```
    animator.SetBool("Par", true);
```

```
    univoice.clip = voice_rock_par;
```

```
    break;
```

case DRAW:

```
    animator.SetBool("Aiko", true);
```

```
    univoice.clip = voice_rock_draw;
```

```
    break;
```

# 가위바위보 액션을 함수로 정의하기

- 다음의 구문을 Rock 스크립트에 추가한다.

```
case WIN:
```

```
    animator.SetBool("Win", true);
```

```
    univoice.clip = voice_rock_win;
```

```
    break;
```

```
case LOOSE:
```

```
    animator.SetBool("Loose", true);
```

```
    univoice.clip = voice_rock_loose;
```

```
    break;
```

```
}    univoice.Play ( ); }
```

# 시작 액션 설정하기

- 다음의 구문을 가위바위보 시작 부분에 추가한다.

case 0:

UnityRockAction(ROCK);

← 추가문장

modeRock ++;

← 추가문장

break;

# 가위바위보 버튼 배치하기

- 처음에 작성한 OnGUI 구문에 다음의 구문을 추가한다.

```
void OnGUI ( ) {  
    if (modeRock == 1) {  
        if (GUI.Button(new Rect(Screen.width / 2 - 120, 400, 100, 100), "바위")) {  
            myHand = GOO;  
            modeRock ++;    }  
    }  
}
```

# 가위바위보 버튼 배치하기

- 처음에 작성한 OnGUI 구문에 다음의 구문을 추가한다.

```
if (GUI.Button(new Rect(Screen.width / 2 , 400, 100, 100), "가위")) {  
    myHand = CHOKI;  
    modeRock ++;    }  
  
if (GUI.Button(new Rect(Screen.width / 2 + 120, 400, 100, 100), "보")) {  
    myHand = PAR;  
    modeRock ++;    } }
```

# 가위바위보 버튼 배치하기





# 판정 처리하기

- 가위, 바위, 보 중에 캐릭터가 무엇을 낼지 랜덤하게 결정하는 코드를 추가한다.

case 2: //판정

```
flgResult = -1;
```

```
unityHand = Random.Range(GOO, PAR+1);
```

```
UnityRockAction(unityHand);
```

```
if (myHand == unityHand)
```

```
{
```

```
    flgResult = DRAW;
```

```
}
```

# 판정 처리하기

```
else { switch (unityHand) {  
    case GOO:  
        if (myHand == PAR) { flgResult = LOOSE; } break;  
    case CHOKI:  
        if (myHand == GOO) { flgResult = LOOSE; } break;  
    case PAR:  
        if (myHand == CHOKI) { flgResult = LOOSE; } break; }  
if(flgResult != LOOSE) { flgResult = WIN; } }  
modeRock ++;  
break;
```

# 결과 표시하기

- 처음에 작성했던 [ case 3: ] 구문에 다음의 코드를 추가한다.

case 3: //결과

```
waitTime += Time.deltaTime;
```

```
if (waitTime > 1.5) { UnityRockAction(flgResult);
```

```
waitTime = 0;
```

```
modeRock++; }
```

```
break;
```

# 결과 표시하기

- [ case 4: ] 구문을 [ case 3: ] 구문 다음에 추가한다.

case 4: //가위바위보 끝내기

    flgRock = false;

    modeRock = 0;

break;

# 버튼에 이미지 설정하기

- GUIStyle을 사용해서 현재 버튼에 이미지를 설정한다.
- Rock 스크립트에 다음의 코드를 추가한다.

```
public GUIStyle btStyleMode;
```

```
public GUIStyle btStyleGoo;
```

```
public GUIStyle btStyleChoki;
```

```
public GUIStyle btStylePar;
```

# 버튼에 이미지 설정하기

- OnGUI 구문에 있는 GUI.Button 코드 일부를 다음과 같이 수정한다.

```
if (GUI.Button(new Rect(10, Screen.height - 110, 100, 100), "가위바위보", btStyleMode))
```

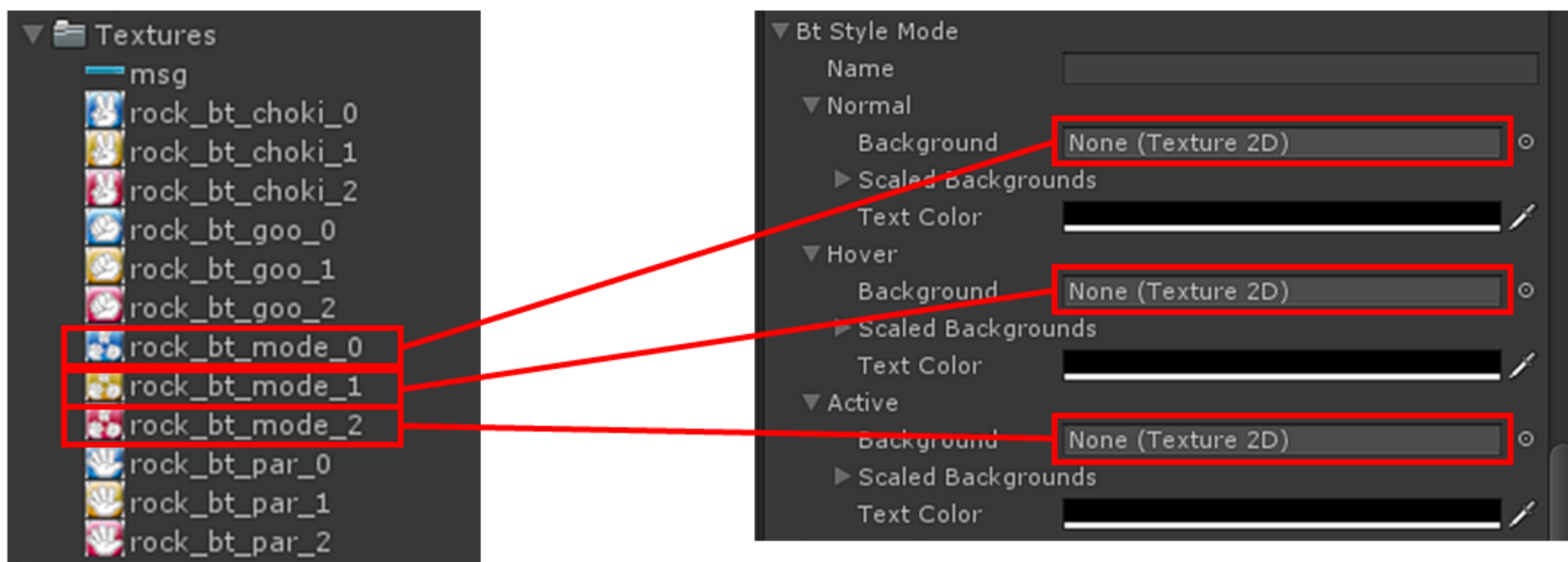
```
if (GUI.Button(new Rect(Screen.width / 2 - 120, 400, 100, 100), "바위", btStyleGoo))
```

```
if (GUI.Button(new Rect(Screen.width / 2, 400, 100, 100), "가위", btStyleChoki))
```

```
if (GUI.Button(new Rect(Screen.width / 2 + 120, 400, 100, 100), "보", btStylePar))
```

# 버튼에 이미지 파일 설정하기

- 첨부된 이미지 파일을 btStyleMode에 연결하고 Image Position을 Image Only로 설정한다. 같은 방법으로 Goo, Choki, Par도 설정한다.



# 입력 대기 처리 추가하기

- 입력 대기 처리를 위해 Update 구문에 있는 [case 1:]에 다음의 코드를 추가한다.

```
animator.SetBool("Rock", false);  
animator.SetBool("Aiko", false);  
animator.SetBool("Goo", false);  
animator.SetBool("Choki", false);  
animator.SetBool("Par", false);  
animator.SetBool("Win", false);  
animator.SetBool("Loose", false);
```