변경

1. **플레이어 UI 변경**
    - 체력 UI : 초상화 → 게이지
    - 변신 지속시간 UI : 게이지 → 아이콘
2. **20190904_1에 있었던 문제점 수정**
    - 에너지파계열 경직 코루틴 함수를 제거
    - OnTriggerEnter2D 함수를 제거
    - 코루틴 함수의 내부 코드를 OnTriggerStay2D 내부로 이동

문제점

1. **에너지파계열 경직코드에서 문제가 다시 발생**
    - 경직 시간에 관해서는 문제가 없으나 움직임 없이 피격당할 시 도중에 데미지 입는 것이 중단됨
    - **움직이면서 피격당할 시** 데미지 중단이 없음
    - 계속 살펴본 결과 OnTriggerStay2D가 도중에 호출되지 않음
    - 코루틴 함수의 내부 코드였던 if문들을 모두 주석처리 해도 동일
2. **20190904_1 VS 20190904_2**

20190904_1

```
void OnTriggerEnter2D(Collider2D coll)
{
  if(coll.CompareTag("Boss_KiBlast"))
    if(CompareTag("Untagged"))
      StartCoroutine(StiffenByKiBlast());
}

void OnTriggerStay2D(Collider2D coll)
{
  if(coll.CompareTag("Boss_KiBlast"))
  {
    if(CompareTag("Untagged"))
    {
      animator.SetBool(hashAss_KB, true);
      Damaged(20.0f * Time.fixedDeltaTime);       //0.02초당 0.4의 데미지 -> 1초당 20데미지 ∴ 총 ㄷ
    }
    else if(CompareTag("Player_Guard"))
    {
      Damaged(5.0f * Time.fixedDeltaTime);        //0.02초당 0.1의 데미지 -> 1초당 5데미지 ∴ 총 데
      animator.SetBool(hashAss_KB, false);
      overStiffenByKiBlast = true;
      leftStiffTime = 0.0f;
      leftFreeTime = 0.0f;
      goku.cantAnythings = false;
    }
  }
}

void OnTriggerExit2D(Collider2D coll)
{
  if(coll.CompareTag("Boss_KiBlast"))
  {
    animator.SetBool(hashAss_KB, false);
    overStiffenByKiBlast = true;
    leftStiffTime = 0.0f;
    leftFreeTime = 0.0f;
    goku.cantAnythings = false;
    goku.stat.health = Mathf.Round(goku.stat.health);
  }
}

IEnumerator StiffenByKiBlast()
{
  do
  {
    if(leftStiffTime >= stiffTime.byKiBlast)
    {
      goku.cantAnythings = false;
      leftStiffTime = 0.0f;
      leftFreeTime = 0.0f;
    }
```

```
    else if(leftFreeTime < stiffTime.duringFree)
    {
      goku.cantAnythings = false;
      leftFreeTime += Time.deltaTime;
    }
    else if(leftFreeTime >= stiffTime.duringFree)
    {
      goku.cantAnythings = true;
      leftStiffTime += Time.deltaTime;
    }
    yield return null;
  }while(!overStiffenByKiBlast);
  overStiffenByKiBlast = false;
  yield return null;
}
```

20190904_2

```
void OnTriggerStay2D(Collider2D coll)
{
  if(coll.CompareTag("Boss_KiBlast"))
  {
    if(CompareTag("Untagged"))
    {
      animator.SetBool(hashAss_KB, true);
      Damaged(20.0f * Time.fixedDeltaTime);       //0.02초당 0.4의 데미지 -> 1초당 20데미지 ∴ 총 D
      if(leftStiffTime >= stiffTime.byKiBlast)
      {
        goku.cantAnythings = false;
        leftStiffTime = 0.0f;
        leftFreeTime = 0.0f;
      }
      else if(leftFreeTime < stiffTime.duringFree)
      {
        goku.cantAnythings = false;
        leftFreeTime += Time.deltaTime;
      }
      else if(leftFreeTime >= stiffTime.duringFree)
      {
        goku.cantAnythings = true;
        leftStiffTime += Time.deltaTime;
      }
    }
    else if(CompareTag("Player_Guard"))
    {
      animator.SetBool(hashAss_KB, false);
      Damaged(5.0f * Time.fixedDeltaTime);        //0.02초당 0.1의 데미지 -> 1초당 5데미지 ∴ 총 데
      leftStiffTime = 0.0f;
      leftFreeTime = 0.0f;
      goku.cantAnythings = false;
    }
  }
}

void OnTriggerExit2D(Collider2D coll)
{
    if(coll.CompareTag("Boss_KiBlast"))
    {
        animator.SetBool(hashAss_KB, false);
        leftStiffTime = 0.0f;
        leftFreeTime = 0.0f;
        goku.cantAnythings = false;
        goku.stat.health = Mathf.Round(goku.stat.health);
    }
}
```