prefab : 코딩 내에서 쓰이는 객체 모아놓은 것.

* 이미지를 scene안에 끌어넣거나 ‘create’해서 Game object를 만들 수 있다. (sprite) 그걸 ‘project’로 끌어넣으면 prefab이된다. (일일이 update를 필요로한다.) -> 게임 저장은 ‘project’에 넣은 것들만 저장이 된다.
* prefab안에서도 하위계층이 존재할 수 있다. 하위를 먼저 prefab으로 만들고 그걸 다시 create에서의 상위계층에 넣어서 하위로 만들고, ‘project’에서 하위를 지운 후 ‘create’에서 하위를 포함한 상위를 ‘project로 옮긴다.
* (1) git add .
* (2) git commit -m “(아무거나// category some file)”
* 위의 두 명령문은 자주할수록 좋다.
* (3) git push origin master(branch 이름)
* (4) git fetch origin (서버 상황을 내컴퓨터에 알림)
* (5) git pull origin master (서버꺼를 내꺼에 내려올 때)
* 처음 작업시 : 오른쪽 마우스 클릭 git bash here을 눌러서 -> 4 >5 > 1> 2> 3 의 순서로 한다.
* 작업할 때 branch 만들기 (또는 옮기기) : git checkout (-b) “branch 이름”
* git branch : branch 몇 개이었는지 알수 있어.
* branch 만든 후에 그것만 update (push)하면 만들어져
* 합칠 때는 github홈페이지에서 pull request - create pull request - merge 전에 모든 조원의 동의가 있어야함.
* Edit-preference- external script editor (비주얼로 바꾸기!)
* 코드만들땐 script에다가// 이미지는 resource //prefab은 prefab만든거는 add component해서 script에 있는거를 update해준다. /
* public으로 선언하면 unity에도 떠서 코드를 변환하지않고도 바로 상태를 볼 수 있다.
* start: Status가 시작될 때 뭘하는지 하는 void가 필요.
* update: 한 frame당 어떤 것을 자동으로 하는 것.
* awake: start보다 먼저 시작된다.
* 60 frame은 1 frame이 1/60 초.
* GetComponentInChildren<Status> 는 하위 prefab을 update하는거랄까.

--------------------------------------------------------------------------------------------

* 글꼴은 네이버나눔고딕으로 하는 거로.