



이 다 흥

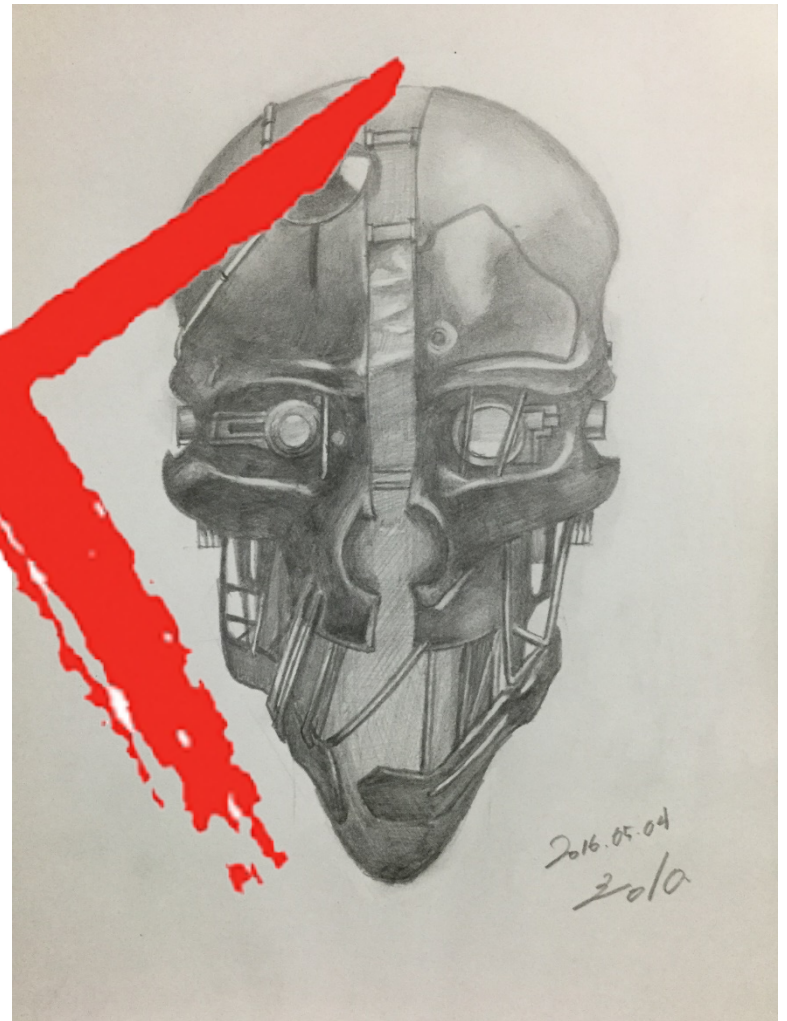
2017103743



경희대학교
KYUNG HEE UNIVERSITY

어떻게 살아왔는지

디자이너



어떻게 살아왔는지

```
209     }
210     dCT = 1.5;
211 }
212 }
213 // 바꾸고자하는 무기, 아이템 트리거,
214 function PickWeapon (code : int, coll2 : Collider){
215     if(weaponCode == 2) return; // 근접무기는 고정
216     if(cT > 0) return;
217     if(Input.GetKeyDown(KeyCode.E)){
218         var change : boolean = true;
219         for(var i : int = 0; i < weaponNum.length; i++){
220             if(code == weaponNum[i]){
221                 change = false;
222                 break;
223             }
224         }
225         if(!change) return;
226         for(var j : int = 0; j < weaponNum.length; j++){
227             if(weaponCode == weaponNum[j]){
228                 Instantiate(weaponItem[weaponCode], Vector3(transform.position.x, transform.position.y - 0.09, transform.position.z), Quaternion.identity);
229                 weaponNum[j] = code;
230                 weaponCode = code;
231                 ChangeMotion ();
232                 Destroy(coll2.gameObject);
233                 break;
234             }
235         }
236     }
237 }
238
239
240 function ItemUse () {
241     if(Input.GetKeyDown(KeyCode.H)){
242         if(hpHealItem == 0) return;
243         hpHealItem--;
244         hp += 50;
245         if(hp >= 100) hp = 100;
246     }else if(Input.GetKeyDown(KeyCode.P)){
247         if(poisonHealItem == 0 ) return;
248         poisonHealItem--;
249         poison = false;
250     }
251 }
```

어떻게 살아왔는지

프

```
63 // Update is called once per frame
64 void Update () {
65     if (nextStep != Step.NONE && step != nextStep) {
66         switch (nextStep) {
67             case Step.CLIMB:
68
69                 break;
70
71             case Step.AIR:
72                 pressBlock.transform.position = new Vector3 (pressBlock.transform.position.x, maxY, pressBlock.transform.position.z);
73                 Invoke ("NextStepIsFall", nextStepTimer);
74                 break;
75
76             case Step.FALL:
77
78                 break;
79             case Step.DOWN:
80                 pressBlock.transform.position = new Vector3 (pressBlock.transform.position.x, minY, pressBlock.transform.position.z);
81                 Invoke ("NextStepIsClimb", nextStepTimer);
82                 break;
83         }
84
85         step = nextStep;
86         nextStep = Step.NONE;
87     }
88
89     switch (step) {
90     case Step.CLIMB:
91         moveVector.y = climbSpeed;
92         //pressBlock.transform.Translate (moveVector * Time.deltaTime);
93         if (pressBlock.transform.position.y >= maxY - 0.05f) {
94             pressBlock.transform.position = new Vector3 (pressBlock.transform.position.x, maxY, pressBlock.transform.position.z);
95             nextStep = Step.AIR;
96         }
97         break;
98     case Step.AIR:
99         moveVector.y = 0;
100         break;
101     case Step.FALL:
102         moveVector.y -= fallSpeed * Time.deltaTime;
```


한권의 책



‘해커와 화가’의 공통점은
둘 다 **창작자** (maker) 라는 것이다.’

‘해커와 화가가 하고자 하는 일은
작곡가, 건축가, 작가와 마찬가지로
좋은 것을 창조해내는 것이다’

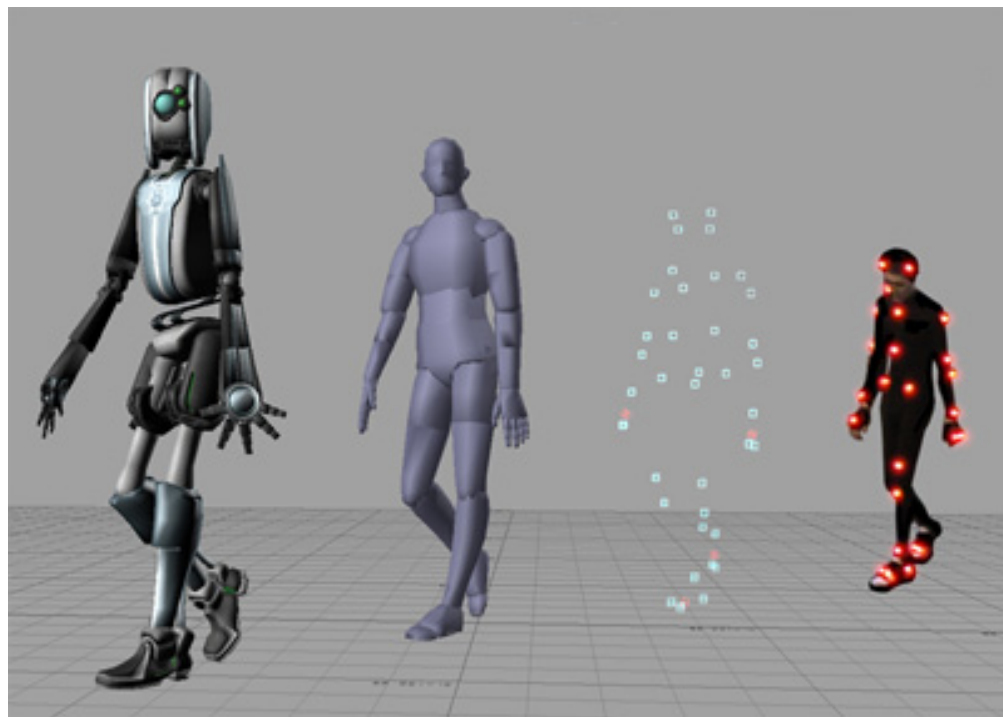
- 해커와 화가

지금은

목표



체계적인 코딩



모델 및 애니메이션 제작

미래모습





Thank you
감사합니다



경희대학교
KYUNG HEE UNIVERSITY