

\* 유니티 인증 어소시에이트 : 게임개발자 시험 준비자료.

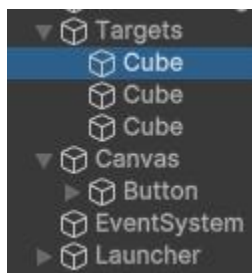
## 시험 출제영역

영역	해석	영역	해석
Animation	unity애니메이션	Asset management	에셋 관리
Audio	소리	Employment preparedness	취업
Game art principles	실제 3D모델링 등	Game design principles	게임 기획
Industry awareness	산업 의식?	Lighting	광원
Materials and effects	질감	Physics	물리학
Programming	전반적인 프로그래밍	Project management	프로젝트 관리
Services	unity엔진 외 부분	Navigation, PathFinding	이동경로 분석
User Interface	UI		

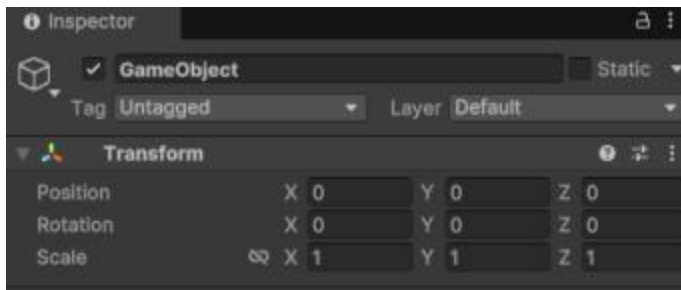
### a. Unity 에디터 및 프로젝트 관리

- Unity 에디터 인터페이스 이해.
- 프로젝트 설정 및 에셋 임포트/구조화.
- 씬 관리 및 씬 간 전환 구현.
- 연습문제

1) 다음 Targets의 자식객체는 무엇인가



- 2) 로그, 에러, 경고 등 보여주는 창 [Console]
- 3) Scene을 저장하면 포함되어 함께 저장되는 것은?  
: Hierarchy Object, Inspector 내용 변경값, Object 위치정보
- 4) Transfrom Component의 중요 기능요소는?



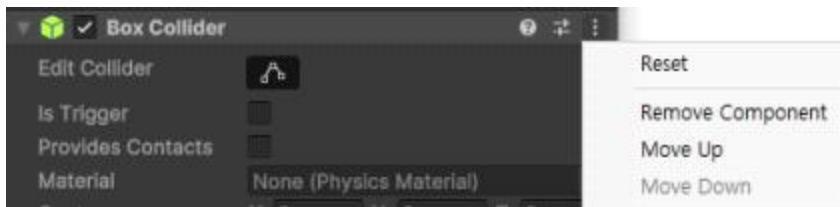
: GameObject name, Tag, Layer, [Transform] position, Rotation, Scale

5) Scene 창에서 뷰를 회전시키려면

: 옵션 / Art키를 누른상태에서 왼쪽버튼으로 드래그 (공전)

: 마우스 우클릭버튼으로 드래그 (자전)

6) 컴포넌트를 초기화 하는 방법?



: 컴포넌트 오른쪽 세로 ... 클릭 후 Reset

7) 선택한 오브젝트에 Scene 뷰창을 이동시킬 수 있는 단축키는?

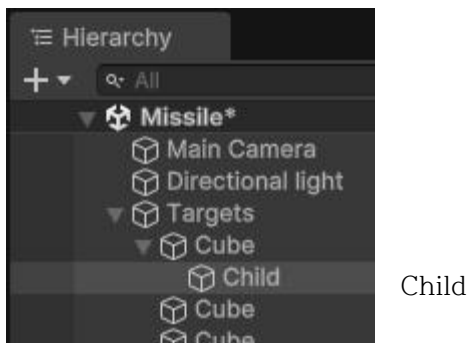
: f - Scene view를 이동

z - 피벗, 센터 모드 전환

8) 오브젝트간에 부모 자신 관계의 계층구조를 만들 수 있는 창은?

: hierarchy

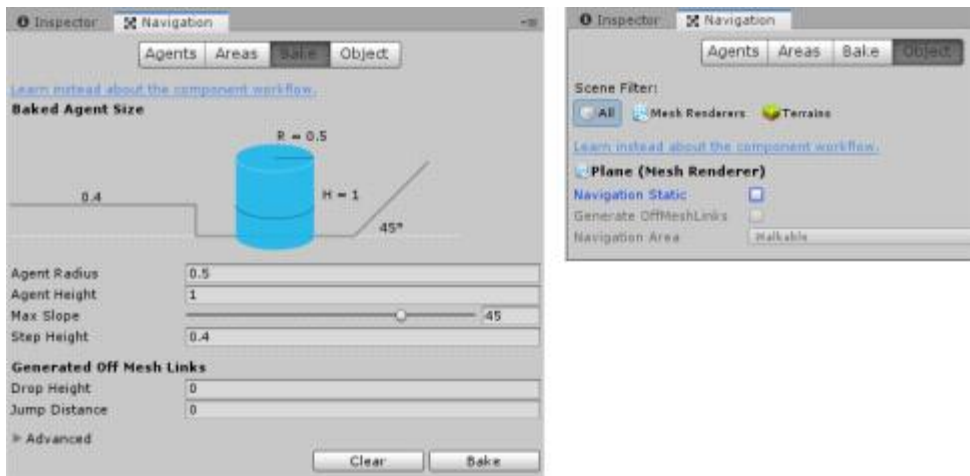
8-1) Cube 오브젝트의 자식 오브젝트를 선택해주세요.



9) Static navigation으로 표시되는 NavMesh를 생성하는 작업은?

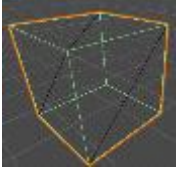

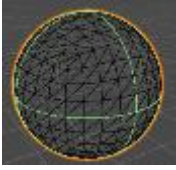


: Baking



10) NavMesh의 설명 : Agent가 이동할 수 있는 표면의 영역을 정의

11) 콜라이더 설명

		
Box Collider	Capsule Collider	Sphere Collider

12) Hierarchy 창에 나타날 수 있는 항목

: GameObject

13) 컴포넌트에서 기어 아이콘을 누르는 경우 나타나는 항목은 : 컴포넌트별 커맨드 메뉴

14) 네비게이션에서 max slope 역할은? : max slope 이상 경사도를 못가게 하도록 설정

15) 새로운 가상 입력축을 매칭하기 위한 것은 : Input Manager

16) 가상 입력축(버티컬 축)에 대한 설명으로 적합한 것은?

: 가상의 축값을 읽고 여러 기기에서 똑같은 좌표만큼의 기능을 할 수 있다.

17) Scene 로딩하기 위한 방법



18) 모든 게임 오브젝트에 기본적으로 존재하는 컴포넌트 : transform

## b. 프로그래밍 (C#)

- C# 기본 문법: 변수, 조건문, 반복문, 함수 등.
- 객체지향 프로그래밍(OOP): 클래스, 상속, 다형성 등.
- Unity API: Transform, Physics, Input 등 주요 API 활용.
- 이벤트 시스템: UnityEvent 및 사용자 정의 이벤트 활용.
- 연습문제

1) 플레이어 오브젝트의 위치를 3,4,5로 지정하려면

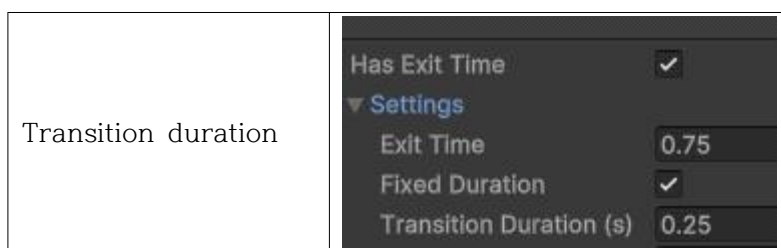
: `Vector3 pos = new Vector(3, 4, 5)`

2) 다른 클래스가 접근 할 수 있는 변수 / 함수 유형은

- : public
- 3) 새로운 스크립트를 생성하는 메뉴 또는 버튼은?  
 : 메인 메뉴 Asset > create > C# script  
 : Project 창에서 오른쪽버튼 > Create > C# script
- 4) 다음 코드의 의도는? Camera mainCam = Camera.main;  
 : 현재 Scene의 메인 카메라 참조를 얻는다.
- 5) 값을 리턴하지 않는 메소드는?  
 : void Start()
- 6) 현재의 게임 오브젝트의 위치를 참조하는 스크립트 코드는?  
 : Vector3 pos = transform.position  
 : Vector3 pos = transform.localPosition
- 7) Awake 메소드가 호출되는 시점은?  
 : 오브젝트가 인스턴스화 될 때. (Scene 시작시), 활성화 (SetActive) 되어 있어야 호출  
 : Start보다 먼저 호출  
 : OnEnable 는 오브젝트가 활성화 될 때 한번 실행됨.
- 8) 메소드의 사용 목적은?  
 : 인스트럭션의 그룹을 지은 것이다.
- 9) 쿼터니언에 대한 설명으로 적합한 것은?  
 : 물체의 회전값을 나타내기 위한 4개의 좌표값을 가진 수학적 요소,

### c. 애니메이션

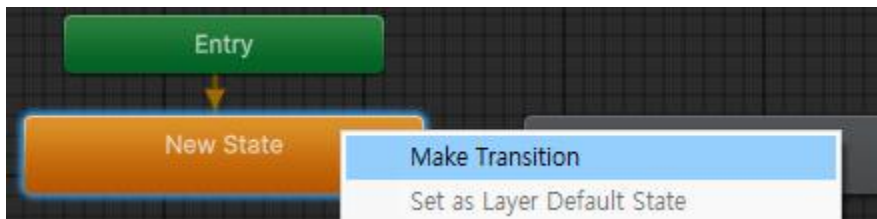
- Animator와 Animation Controller 설정.
  - 트랜지션(상태 전환)과 Blend Tree 사용.
  - 키프레임 애니메이션과 Timeline.
  - 연습문제
- 1) 2개의 애니메이션이 중첩될 때, 그 기간을 설정하려면 무엇을 조절해야할까?



- 2) 에셋 폴더에서 애니메이션 클립과, 컨트롤러를 구분하면?



- 3) 애니메이터 컨트롤러를 오브젝트에 적용시키는 방법은?  
 : 에셋폴더 내, 애니메이터 컨트롤러를 하이어라키 내 게임오브젝트에 드래그 앤 드랍  
 : 적용한 게임오브젝트 선택 후 애니메이터 컨트롤러를 Inspector로 드래그 앤 드랍
- 4) 애니메이터 컨트롤러를 생성하는 방법은?  
 : 메인메뉴 > Asset > create > animator controller  
 : Asset창 > 마우스 우클릭 > create > animator controller
- 5) 하나의 파티클에서 또 다른 파티클이 생겨나는 다중 생성 방법은?  
 : Sub-Emitter
- 6) 파티클 시스템에서 Color가 변하는 속성은?  
 : Color over LifeTime
- 7) Animator Controller 의 설명으로 적절한 것은?  
 : 게임오브젝트를 위한 애니메이션을 준비 및 상태관리
- 8) 애니메이터에서 새로운 상태 (State)로 이동할 수 있는 transition 생성방법



- : 클릭 후 마우스 오른쪽 버튼 클릭 후 Make Transition
- 9) 애니메이터 컨트롤러에서 기본적으로 실행되는 스테이트는 : entry
- 10) 다음 인수값과 매칭되는 항목

보기a	보기b
bool	true/false
int	정수
float	소수점을 포함한 실수
trigger	상태값이 바뀔 때 참조하는 조건

- 11) 유니티 에디터로 3D프로그램으로 작업한 모델링 (Bone, Mesh, Animation 포함) 에셋을 가져오기 위한 파일형태는 : FBX(국제규격)

#### d. 에셋 관리

- 에셋의 임포트/익스포트 및 파일 최적화.
  - 리소스 관리: 메모리 사용량 최적화.
  - 연습문제
- 1) 유니티 및 크리에이티브가 제공하는 유/무료 에셋을 구매하고 사용할 수 있게 해주는 것  
 : [에셋스토어]
- 2) Prefab에 대한 올바른 설명은?  
 : 복제해서 쓰기 위한 프로토타입 클론 오브젝트
- 3) Prefabs를 생성하기 위해서는 Hierachy에서 Asset으로 drag & drop

e. 물리 시스템

- Rigidbody와 Collider를 이용한 물리 상호작용.
- Unity의 물리엔진을 사용해 충돌 및 트리거 구현.
- 중력, 마찰력, 반동 등 물리 법칙 적용.
- 연습문제

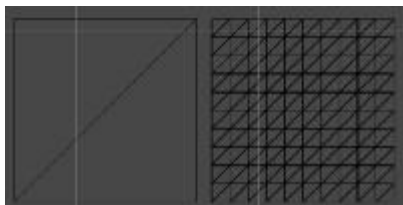
1) Rigidbody의 속성을 연결하세요.

보기a	보기b
Mass	오브젝트의 무게
Collision Detection (연속충돌검사)	Continuous/discrete/continuous Dynamic
Interpolate	부드러운 프레임 적용
Drag	공기의 저항
Angular Drag	회전 시 공기저항

2) 물리엔진에서 Rigidbody에 중력을 적용하기 위해서는 **[use Gravity]** 사용

3) Quad 와 plane의 차이

: 삼각형 개수 차이 (quad가 폴리곤을 더 적게 사용)



[좌 Quad, 우 Plane]

4) 오브젝트에 적용된 콜라이더만 회전시킬 수 있는 방법은?

	<p>Edit Collider Center, Size 수정</p>
--	------------------------------------------

5) Raycast의 역할은?

- : 주어진 광선(ray)의 경로를 따라 collider(충돌영역)가 존재하는지 검색
- : RaycastHit - 충돌체의 충돌 정보를 저장

6) 다음 컴포넌트와 그 속성을 연결하세요.

보기a	보기b
collider	Physics Material

Mesh renderer	Reflector Probe Option
Animator	Avatar
Camera	Depth

- 7) 게임 오브젝트에 Force 및 Torque를 적용하고 싶을 때 추가하는 컴포넌트는?  
: Rigidbody
- 8) Physics.Raycast 메소드의 리턴 결과는?  
: RaycastHit
- 9) 다음 게임 오브젝트가 오브젝트의 콜라이더를 통과하기 위한 속성은  
: isTrigger
- 10) Raycasting에 layer를 사용하는 이유는  
: 충돌체체크의 수를 줄여줌
- 11) 메인 메뉴를 이용해 박스 콜라이더를 추가하려면  
: components > physics 2D > box collider 2D

#### f. 오디오

- AudioSource와 AudioClip 사용.
  - 3D 공간 오디오 설정.
  - 오디오 믹서 및 볼륨 제어.
  - 연습문제
- 1) 유니티 오디오가 지원하는 파일  
: AIF, WAV, MP3, OGG
  - 2) 도플러효과(Doppler Effect)란 무엇인가?  
: 리너스(카메라)의 위치에 따른 무빙 사운드에서 발생하는 주파수 변이
  - 3) 오디오 컴포넌트 중 동굴, 운동장, 숲 처럼 울림 또는 왜곡현상을 쓸 수 있는것은?  
: 리버브 존(Reverb zone)- 음장효과

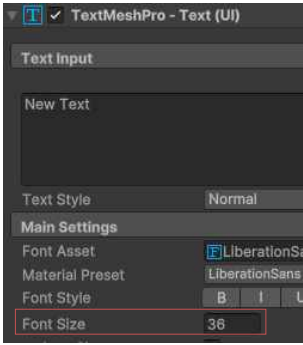
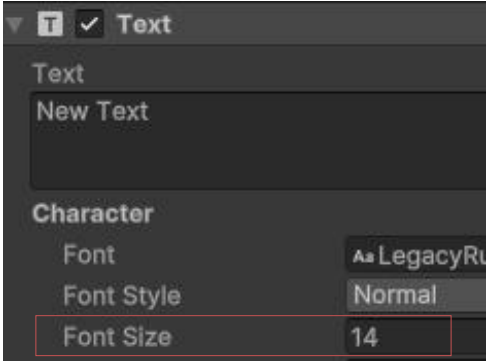


Play On Awake : 게임 시작 시 플레이  
 Loop : 반복  
 Reverb Zone : 왜곡현상 정도\

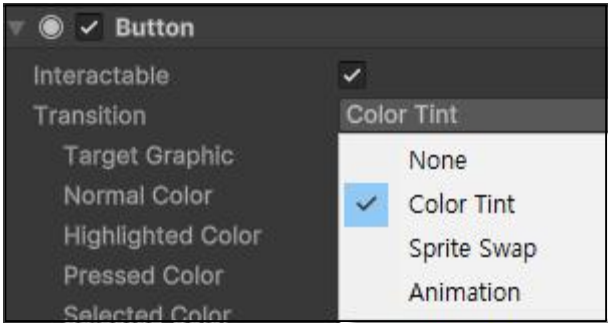
#### g. UI 설계

- Canvas, Text, Image 등의 기본 UI 컴포넌트 사용.
- 버튼, 슬라이더, 드롭다운 등 상호작용 요소.
- UI 애니메이션 및 동적 업데이트.
- 연습문제

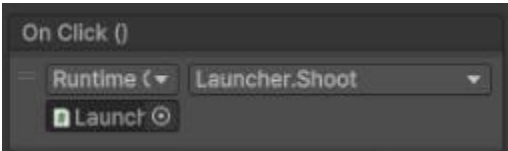
- 1) Rect Transform에서 앵커와 피봇을 미리 지정된 위치로 이동시키는 영역은?  
: Anchor Preset
- 2) Inspector창에서 텍스트의 폰트를 조절하려면 어디를 봐야하나?

TextMeshPro	Text (Legacy)
	

- 3) Button Transition 효과로 선택 가능한 것은?

	<p>Color Tint Sprite Swap Animation</p>
-------------------------------------------------------------------------------------	-------------------------------------------------

- 4) 버튼을 눌렀을 때 호출되는 이벤트는?

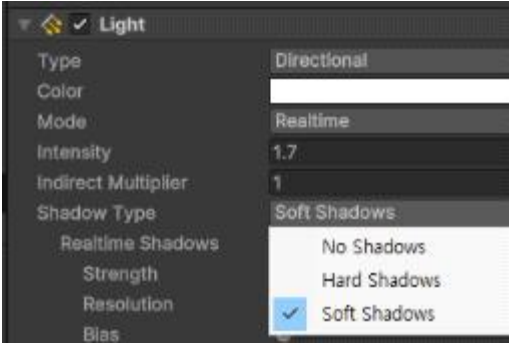
	<p>On Click Event</p>
-------------------------------------------------------------------------------------	-----------------------

- 5) UI 텍스트가 범위 밖으로 나갈 수 있도록 허용한다면?  
: Horizontal OverFlow , Vertical OverFlow

#### h. 조명과 렌더링

- Realtime Light와 Baked Light 차이 이해.
- 그림자 설정 및 최적화.



	프로퍼티	기능
	Shadow Type	Hard Shadows 설정을 사용하면 가장자리가 날카로운 그림자가 생성. Hard Shadows는 Soft Shadows에 비해 특별히 사실적이지 않지만 처리가 적고 다양한 용도에서 사용
	Strength	그림자의 어두운 정도가 결정
	Resolution	새도우 맵의 “카메라”에 대한 렌더링 해상도를 설정
	Bias / Normal Bias	그림자의 포지션과 정의를 미세 조정
	Shadow Near Plane	그림자를 렌더링할 때 가까운 평면에 대한 값을 선택할 수 있습니다. 이 거리보다 광원에 가까운 게임 오브젝트는 그림자를 캐스트하지 않음

- 후처리 효과(Post-Processing).
  - Post-Processing은 게임의 시각적 품질을 향상시키기 위해 카메라에 적용할 수 있는 효과 및 필터를 제공하는 기능
- Post-Processing은 후처리 작업 이라는 기술을 사용, 카메라가 영상을 내보내기 직전에 후처리 작업을 통해 다양한 효과와 필터를 적용하여 최종적으로 향상된 영상을 내보내는 기술을 후처리 작업이라 함.



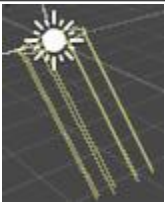



#### - 기본효과 / 필터

Ambient Occlusion	한 장면의 각 점이 앰비언트 라이팅(광원)에 얼마나 노출되어 있는지 계산하기 위해 사용되는 셰이딩 및 렌더링 기법
Auto Exposure	이미지의 밝기 수준 범위에 따라 이미지의 노출을 동적으로 조정
Bloom	이미지의 밝은 영역의 경계에서 확장되는 광원을 생성
Chromatic Aberration	카메라의 렌즈가 동일 지점으로 모든 색을 집중시키지 못한 경우 발생하는 효과

Color Grading	최종 이미지의 색과 명도를 변경하거나 교정하는 과정
Depth Of Field	카메라 렌즈의 포커스 기능을 모방하는 효과
Grain	필름의 작은 파티클에 의해 발생하는 랜덤 광학 텍스처
Lens Distortion	카메라 렌즈의 왜곡을 모방하는 효과
Motion Blur	카메라에 의해 촬영되는 오브젝트가 카메라의 노출 시간보다 더 빠르게 움직일 때 이미지의 블러링을 모방하는 효과
Screen Space Reflections	스크린 공간 데이터를 재사용하여 반사를 계산하는 기법
Vignette	중심과 비교하여 이미지의 테두리를 향해 어두워지는 효과 또는 중심에 비해 채도를 줄이는 효과

- 연습문제

1) 라이트 시스템과 광원의 형태를 연결하시오.

보기a	보기b	그림
directional	cylinder	
spot	cone	
point	sphere	
area	rectangle	
skybox	ambient	

2) 바닥에 그림자를 생성하는 광원은 [directional light]

3) Ambient Occlusion(반사광)의 최대거리 Max Distance 값을 늘리면 발생하는 현상은?  
: 반사광의 계산 범위가 늘어난다.

4) Light Baking이란 무엇인가?


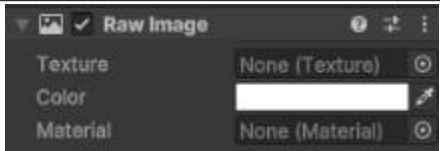
: 게임의 속도를 빠르게 하기 위해 광원에 해당하는 그림자를 미리 그리도록 텍스처를 맵

핑하는 작업

- 5) 다음의 PostProcessing(후작업) 효과 중 조리개(aperture)를 기반으로 blur효과를 나타내는 것은? **[Depth of Field (피사체심도)]**
- 6) 라이트 바운싱을 이용한 부드러운 그림자를 만드는 shadow는? **[soft shadow]**
- 7) 다음 중 셰이더의 속성값과 설명을 연결하세요.

보기a	보기b	그림
emission	표면에 라이트를 생성하는 rgb값	
normal map	기본의 폴리곤에 세부적인 표현을 추가	
metallic	환경의 반사를 추가, albedo를 감소	
softness	마이크로 표면의 라이트의 퍼짐을 추가	

- 8) Light Probes (라이트 프로브) 는 무엇인가?
  - : Scene의 특정위치에서 반사된 light 값을 측정한다.
  - : Scene의 빈공간을 지나가는 빛에 대한 정보를 파악해서 사용함.
  - : 베이킹된 조명 정보를 Scene에 저장
- 9) Image vs RawImage

Image	스프라이트를 렌더링 (오브젝트 형태표현 이미지)	
RawImage	텍스처를 렌더링 (오브젝트 표면, 느낌 표현)	

- 10) 하나의 이미지를 여러개 스프라이트로 분리하는 도구는



11) 재질 (Material)의 albedo가 의미하는 것은?

: 라이팅 데이터가 없는 재질의 color

: 언릿은 라이트 값을 계산안해서 알베도 값이 100이 됨

12) 폴리곤 모델 에셋의 최적화 방법은?

: 폴리곤 수를 줄임, LOD(Level of detail), draw call(텍스처 수를 줄이는 방법)

13) 다음 재질과 적당한 렌더링 모드를 연결하시오

보기a	보기b
벽돌/돌	불투명 Opaque
아크릴/성애유리	투명 Transparent
와이어 펜싱	컷아웃 Cutout
파워업 아이콘	페이드 Fade

Opaque [오페이크]	디폴트입니다. 투명한 영역이 없는 일반 솔리드 오브젝트에 적합합니다.
Cutout [컷아웃]	투명 영역과 불투명 영역 사이에 하드 에지가 있는 투명 효과를 만들 수 있습니다.
Transparent [트랜스페어런트]	투명한 플라스틱이나 유리처럼 투명한 머티리얼을 사실적으로 렌더링하는 데 적합합니다
Fade [페이드]	오브젝트 페이드 인 또는 페이드 아웃을 애니메이션으로 나타내고 싶을 때 유용합니다.

#### [ Material : Rendering Mode ]

14) lightmap의 사용 목적은?

: 미리 baking했으니 렌더링 속도가 빨라짐

#### i. 게임 디자인 및 플레이 메커닉

- 게임의 흐름(flow) 설계.
- 플레이어 상호작용 요소와 피드백 디자인.
- 레벨 디자인 기본 원리.
- 연습문제

1) 다음 중 기획, 멀티태스킹, 리소스 관리가 중요한 장르는?

: 비행시뮬레이션, 레이싱, 스토리, 타워디펜스, 어드벤처

: 퍼즐, 캐주얼, 빌딩 시뮬레이션, 실시간 전략, 카드, gambling

2) 델타타임의 의미

: 화면 프레임 간의 시간간격

3) 게임 스튜디오 직종 / 역할이 맞게 매칭된 것은?

보기a	보기b
레벨 디자이너	게임 환경의 기획 및 빌드
게임 디자이너	게임 메카닉을 디자인 및 구현
컨셉아티스트	캐릭터의 아이디어 초안 작성
아트디렉터	룩 앤 필의 일관성 확인

#### j. 서비스 및 퍼블리싱

- Unity Ads, Unity Analytics, Unity IAP 연동.

- 게임 빌드 및 플랫폼별 최적화(Android, iOS, PC 등).

- 연습문제

1) 개발 스튜디오 작업중 정보의 노출을 제한하는 계약은? [기밀유지 계약]

2) 콘솔용 하드웨어가 일반적인 PC용 하드웨어와 다른 점은?

: 콘솔이 더 싸지만, 특정 용도로 사용되기 위해 특화되어 빠르다.

3) 제작한 게임 캐릭터, 중요 테마, 독점적인 기술 등은 어디에 해당하는가?

: 지적재산권

4) unity analytic을 이용할 때, 이 프로젝트를 식별하는 방법은?

: 프로젝트ID

5) 디바이스 별 주요 컨트롤 유형을 연결하세요.

보기a	보기b
모바일	터치
콘솔	전용 컨트롤러
PC	마우스, 키보드
VR	VR헤드셋, 전용컨트롤러

6) 다음 중 게임의 분위기에 가장 영향을 많이 끼치는 항목은?

: Scene

7) 다음 중 Cloud Bulid를 지원하는 플랫폼은?

: Web, iOS, android, PC, Linux

8) Reward AD, Simple AD의 차이점은?

: 스킵이 되냐, 안되냐 차이

: 리워드는 스킵이 안됨