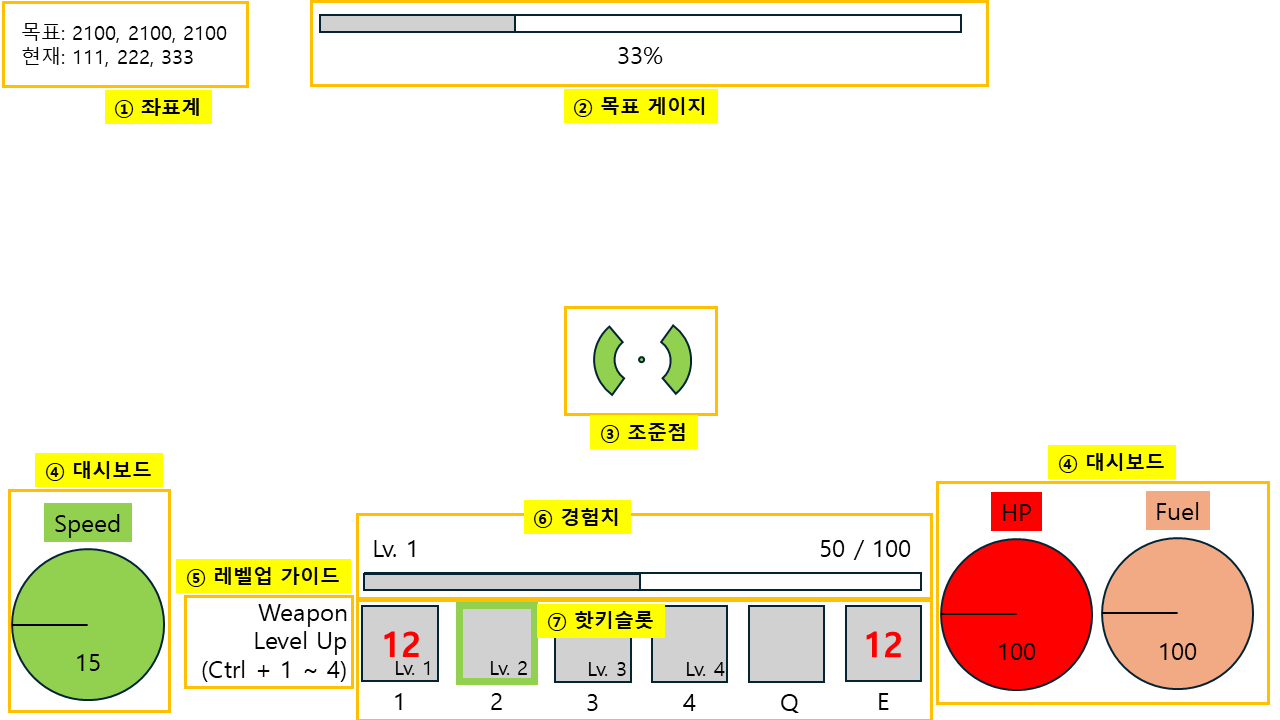
**Retreat Protocol**

**UI 기술 문서**

본 문서에서는 게임에 사용된 UI의 구현 과정에 대해 기술한다. 특별한 기술 없이 유니티에서 제공하는 버튼 컴포넌트만으로도 구현 가능한 타이틀 화면과 함선 선택 화면의 UI는 생략하고, 게임 중 화면 UI에 대해서만, 그 중에서도 특별한 기술을 사용한 UI의 구현 과정만 기술한다.

**게임 중 화면**



**목표 게이지**

현재 게임 진행도(플레이어가 목표에 가까운 정도)를 직관적으로 표시하기 위한 게이지이다.

게임 시작 시 플레이어와 목표 위치 사이의 거리 md를 계산하여 저장하고, 매 프레임마다 현재 플레이어와 목표 위치 사이의 거리 cd를 계산한다. 이후 md와 cd의 비를 계산하여 현재 게임 진행도(백분율)를 산출한다.

게이지는 이렇게 계산한 현재 진행도를 기반으로 표시된다. 0%일 경우 게이지가 완전히 비어 있고, 100%일 경우 완전히 차오른다. 그 외의 경우 현재 진행도에 맞춰 적절히 게이지를 채운다. 현재 게임 진행도를 보다 정확하게 알 수 있도록 게이지 하단에 텍스트(정수)로도 표시한다.

게이지는 두 개의 막대 이미지, 게이지의 배경이 될 이미지 BG와 게이지 내부에서 차오를 이미지 G로 구성된다. G의 x pivot을 0으로 설정하면 x축의 왼쪽 끝을 중점으로 삼게 되는데, 이 상태에서 막대 이미지의 width 값을 조절할 경우, 중점을 기준으로 늘었다 작아지므로 게이지가 차는 연출을 구현할 수 있다.

|  |
| --- |
| void Update()  {  게이지\_출력 함수 호출  }  int 백분율\_계산  {  현재 플레이어와 골 오브젝트 사이의 거리를 계산(Vector3.Distnace 활용)  현재 거리와 최대 거리의 백분율을 계산  if( 현재 거리 값 < 최대 거리 값 ) 백분율 = 0  이렇게 계산한 백분율을 반환  }  void 게이지\_출력  {  백분율\_계산 함수 호출하여 백분율을 반환 받음  백분율의 비율만큼 G의 width를 BG의 n%로 계산  계산한 G의 width를 실제로 적용(Mathf.lerp 보간 활용)  현재 백분율을 텍스트로도 출력(TextMeshProUGUI 활용)  } |

**대시보드**

함선의 속도와 HP, 연료를 직관적으로 표시하는 대시보드다. 속도 대시보드는 화면의 좌측 하단에, HP와 연료 대시보드는 화면의 우측 하단에 표시한다. 대시보드는 바늘과 텍스트로 구성되어 있다. 바늘은 플레이어에게 현재 값을 추상적으로 보여주고, 텍스트는 정확한 값으로 보여준다.

대시보드가 가리키는 변수의 현재값과 최대값의 비를 백분율로 계산하고, 계산한 백분율에 맞춰 적절한 값을 가리키도록 바늘의 각도를 조정한다. 바늘의 각도가 변경될 때, 보간을 사용하여 부드럽게 변경되도록 한다.

대시보드의 바늘은 긴 막대 이미지를 사용하는데, 막대 이미지의 x pivot을 1로 설정하면 x축의 오른쪽 끝을 중심으로 삼을 수 있다. 막대를 회전시키면 중점을 중심으로 회전하기 때문에, 대시보드의 바늘이 움직이는 연출을 구현할 수 있다.

하나의 코드로 서로 다른 값을 가지는 세 가지 대시보드를 출력하기 위해, 오브젝트의 이름을 기반으로, 해당 대시보드가 보여야 하는 변수를 따로 참조하여 출력하게 한다.

|  |
| --- |
| void Start()  {  gameObejct.name을 기반으로 자신이 담당할 변수의 이름을 string 형으로 저장  string 값을 기준으로 변수를 PlayerCtrl 클래스에서 검색 (typeof().GetField() 활용)  검색한 변수의 최대값, 최소값 변수도 함께 검색  검색한 변수들을 float 형으로 변환하여 저장  }  void Update()  {  바늘\_각도\_적용 함수 호출  }  int 바늘\_각도\_계산  {  대시보드로 출력할 변수의 최대값과 현재 값의 비를 백분율로 계산  if( 현재값 < 최소값 ) 백분율 = 0  }  void 바늘\_각도\_적용  {  바늘\_각도\_계산 함수를 호출하여 백분율을 반환 받음  백분율을 기반으로 0에서 180 사이의 각도 값을 계산  계산한 각도 값을 바늘 이미지의 rotation에 적용(Mathf.lerp 보간 활용)  } |

**핫 키 슬롯**

플레이어가 사용 가능한 무기와 스킬의 아이콘, 재사용 대기시간, 레벨, 단축키를 표시한다.

무기와 스킬의 레벨이 1미만인 경우, 아이콘에 (1,1,1,0.5f) 색상을 적용하여 흐리게 표시한다. 레벨이 1이상인 경우, 아이콘에 (1,1,1,1) 색상을 적용하여 또렷하게 표시한다.

무기는 4레벨로 분화되기 때문에, 아이콘의 우측 하단 모서리에 현재 레벨을 텍스트로 표시한다. 스킬은 레벨이 분화되지 않기 때문에, 따로 레벨을 표시하지 않는다.

현재 선택된 무기의 경우, 아이콘 테두리의 색상을 변경하여 강조한다.

무기 또는 스킬이 재사용 대기상태(쿨타임)인 경우, 남은 대기시간을 아이콘의 중앙에 텍스트(정수)로 표시한다. 대기시간은 1초마다 1씩 줄어든다.

|  |
| --- |
| void Update()  {      if( 플레이어의 무기 선택 감지 시 )      {          if( 지금 선택한 무기가 이미 선택된 무기와 다른가? )          {              현재 선택된 무기 = 지금 막 선택한 무기              현재 선택된 무기의 아이콘 테두리를 강조          }          else 아무것도 하지 않음      }      무기\_쿨타임\_표시 함수 호출      무기\_레벨\_표시 함수 호출  }  void 무기\_쿨타임\_표시  {      if( 현재 쿨타임 > 0 ) 현재 쿨타임을 텍스트로 출력 (TextMeshProUGUI 활용)      else 현재 쿨타임을 공백으로 출력 (TextMeshProUGUI)  }  void 무기\_레벨\_표시  {  if( 현재 레벨 < 0 ) 아이콘 색상을 new Color(1,1,1,0.5f)로 변경  else 아이콘 색상을 new Color(1,1,1,1)로 변경  현재 레벨을 텍스트로 출력 (TextMeshProUGUI)  } |