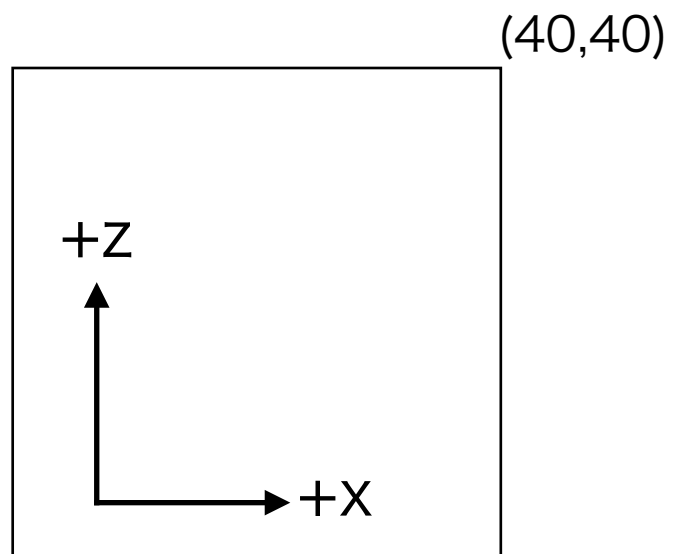
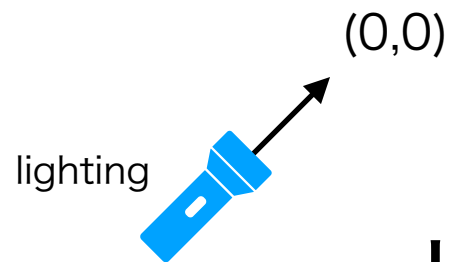
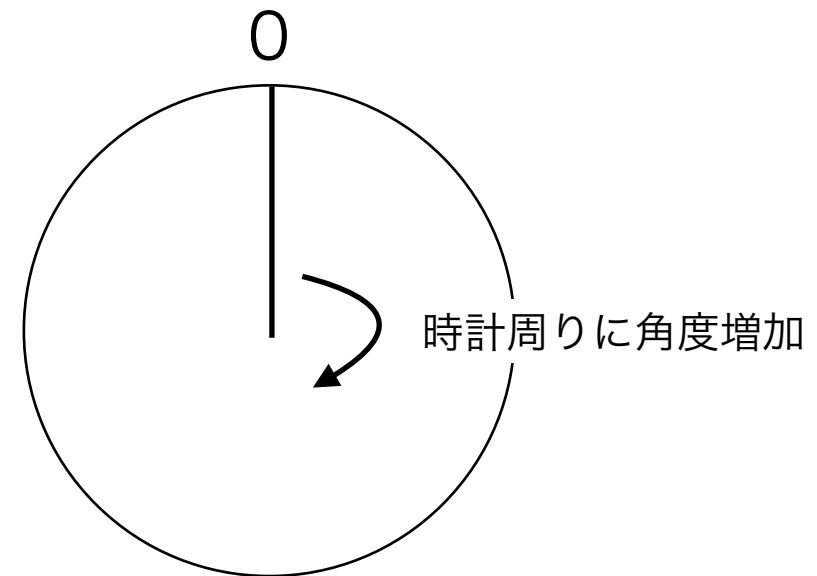


## Coordinate

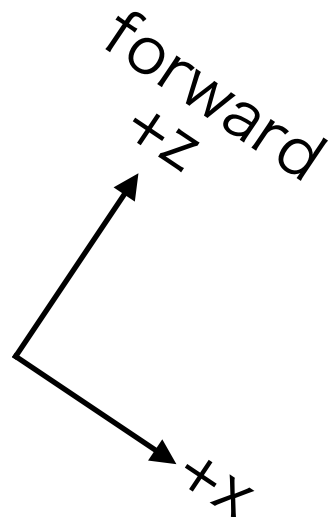


## Orientation

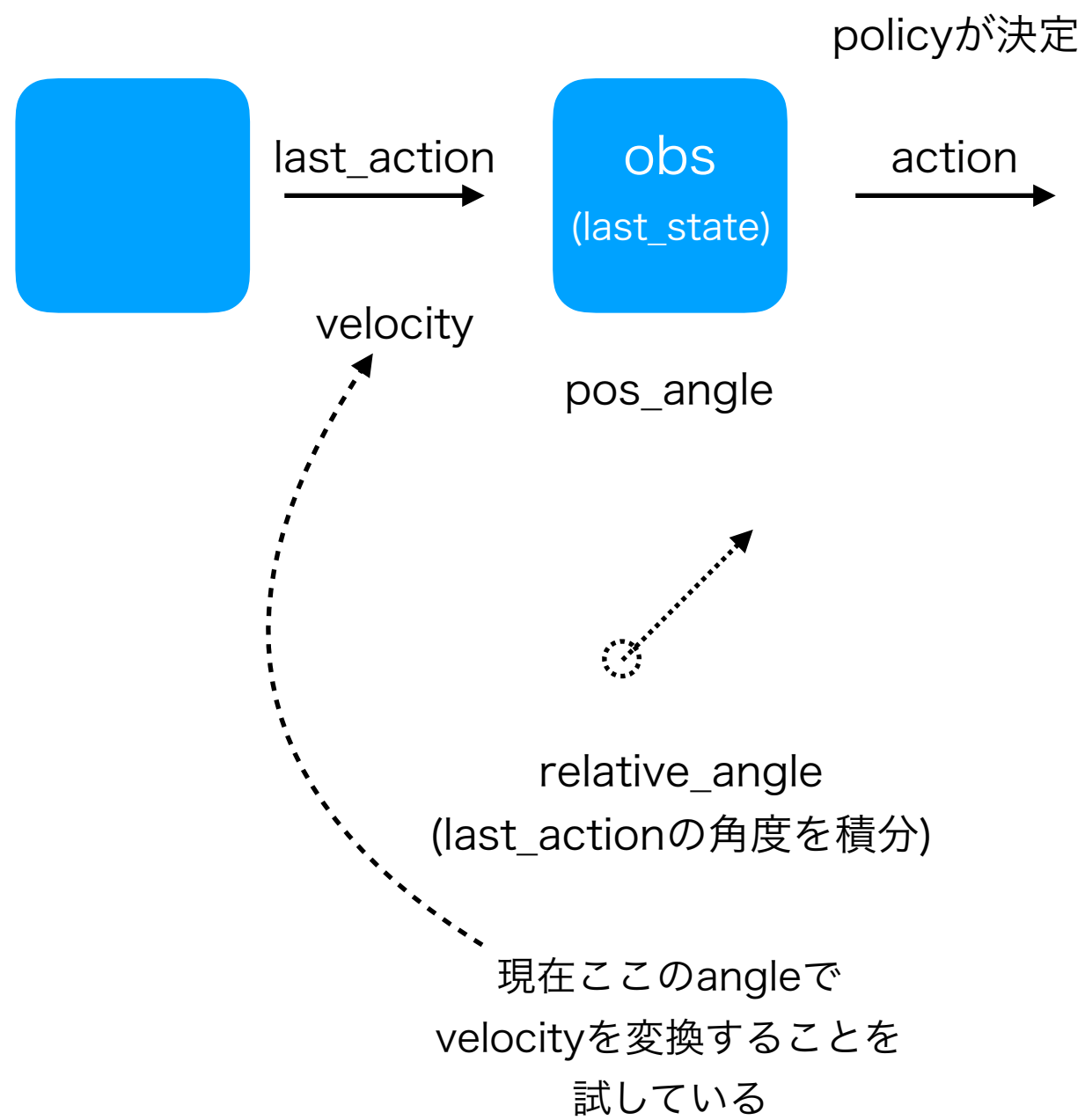


lighting

## Local velocity



※ x方向もあってるか要確認



## display, submit

env = AnimalAIEnv <gym.Env>

- env.reset() 返り値未使用
- Loop
  - 初回 or done後
    - last\_action=[0,0]でenv.step()
    - obs, done, info差し替え
    - estimator, integrator.reset()
  - 積分処理 or 自己位置推定
    - (last\_action, velocity利用)
  - policyがaction決定
    - env.step(action)
    - obs, done, info得る
  - doneなら
    - 次Loop時に[0,0]でenv.step()
  - last\_action = action

基本action決定前にその時点での  
last\_action, velocityで推定/積分

## trainer\_controller

env = UnityEnvironment

- curr\_info = env.reset()
- **last\_action = [0,0]追加**
- Loop
  - **積分 or 自己位置推定ここに挿入**
    - curr\_info.local\_done時にintegrator.reset()
    - local\_done時のlast\_actionそのまま使わず  
last\_action = [0,0]に置き換える
    - local\_done時のvelocityはそのまま使わず  
(0,0,0)に置き換える必要あり
  - action = trainer.take\_action(curr\_info)
    - policyがaction決定
  - new\_info = env.step(action)
    - local\_doneもここに入っている
  - trainer.update\_policy()
  - curr\_info = new\_info
  - **last\_action = action 追加**

local\_done時の次stateはもう次episodeの画面に  
なっている. velocityは(0,0,0)ではないので注意

## UnityEnvのlocal\_done時の挙動

