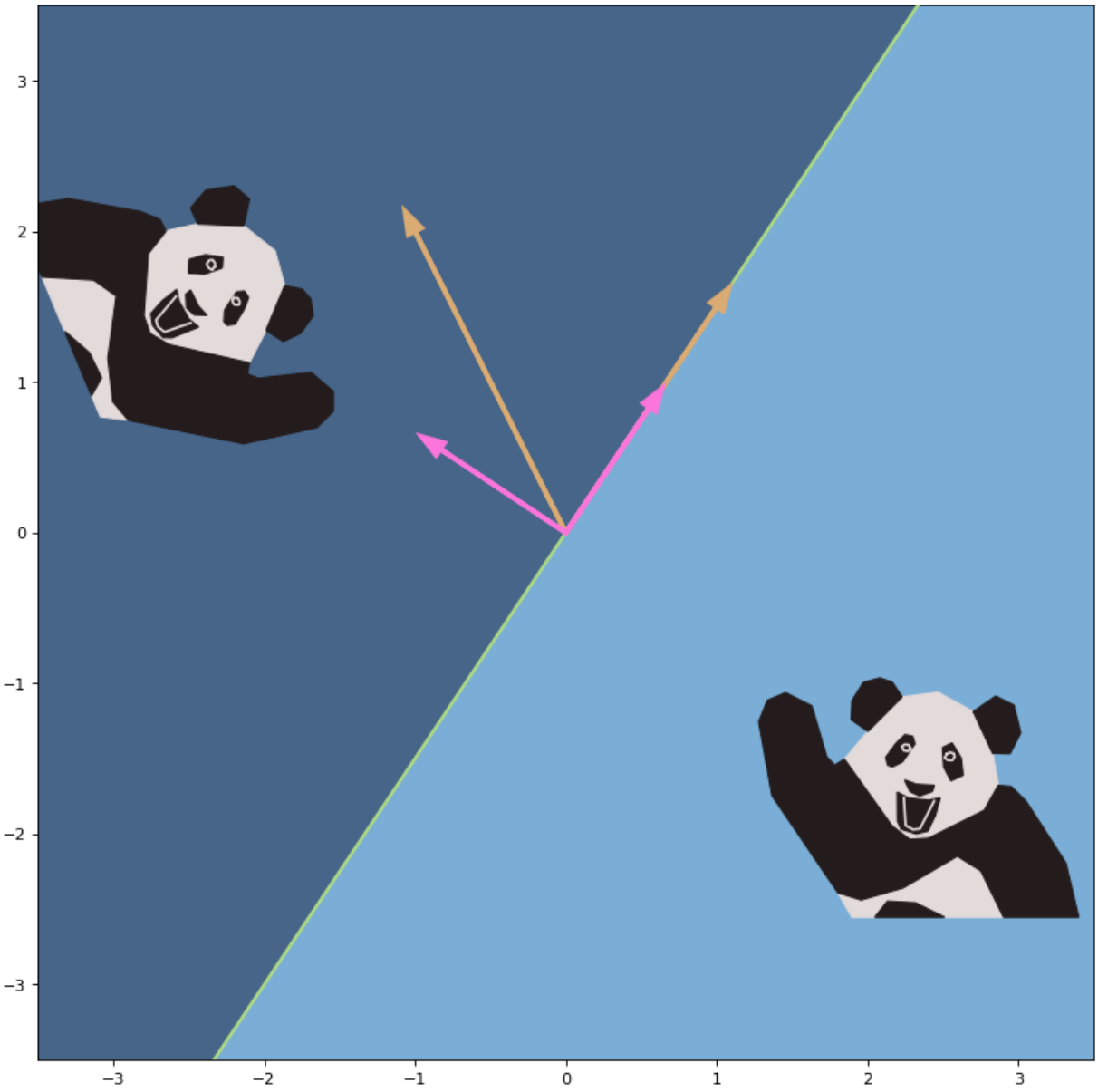
Reflecting Bear

In [1]:
```
import numpy as np
from numpy.linalg import norm, inv
from numpy import transpose
from bearNecessities import *
```

In [25]:
```
# 벡터의 직교화 E 와 선형변환을 위한 T 를 이용한 대칭행렬 계산
def build_reflection_matrix(bearBasis):
    E = gsBasis(bearBasis)
    # 주어진 벡터공간(bearBasis) 기준 y축 대칭이동 시키는 행렬 T
    TE = np.array([[1, 0],
                   [0, -1]])
    T = E @ TE @ inv(E)
    return T
```

In [26]:
```
%matplotlib inline
import matplotlib.pyplot as plt

# 팬더를 대칭시킬 임의의 기준 벡터공간
bearBasis = np.array([[1, -1],
                      [1.5, 2]])

# 선형변환 행렬 계산
T = build_reflection_matrix(bearBasis)

reflected_bear_white_fur = T @ bear_white_fur
reflected_bear_black_fur = T @ bear_black_fur
reflected_bear_face = T @ bear_face

ax = draw_mirror(bearBasis)

# 오른쪽 하단에 위치한 원본 팬더에 대한 그림
ax.fill(bear_white_fur[0], bear_white_fur[1], color=bear_white, zorder=1)
ax.fill(bear_black_fur[0], bear_black_fur[1], color=bear_black, zorder=2)
ax.plot(bear_face[0], bear_face[1], color=bear_white, zorder=3)

# 직교벡터기준으로 대칭시켜 왼쪽 상단에 위치한 팬더 그림
ax.fill(reflected_bear_white_fur[0], reflected_bear_white_fur[1], color=bear_
ax.fill(reflected_bear_black_fur[0], reflected_bear_black_fur[1], color=bear_
ax.plot(reflected_bear_face[0], reflected_bear_face[1], color=bear_white, zor
```



In [ ]: