# 516.https://stackoverflow.com/questions/70623145/exchange-own-erc20-token-to-another-token-in-smart-contract-for-game

**T:**exchange own ERC20 token to another token in smart contract for Game

**Q:**I want to deploy a smart contract (ERC20) for the game,so the purpose is to keep points.  
  
 ● when someone enters the game, we will ask for some crypto coin (ex. ETH) and give some of our own points  
  
 ● while playing the game, the user will earn some points.  
  
 ● Then that user can get crypto coins (ex. ETH) from that points.  
  
I can write a smart contract to manage points.But I wonder if I can have a function to exchange our points to existing crypto coins (ex ETH) inside of our smart contract.  
  
Does someone know the right way to do it?

1 **Answer**

**A1:**You can calculate the amount of ETH based on a price, and then use the transfer() native function of address payable type.  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
mapping (address => uint256) pointBalances;// 1 point for 100 wei, assuming the points have 0 decimalsuint256 price = 100;function sellPoints(uint256 \_amount) external { require(pointBalances[msg.sender] >= \_amount, "Insufficient balance"); pointBalances[msg.sender] -= \_amount; uint256 weiAmount = \_amount \* price; payable(msg.sender).transfer(weiAmount);}  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]

**C1:**Can't we transfer another token like MIM or ADA?

**C2:**@Talent You can. An easy way is to invoke the .transfer() function of the token contract (not of the address payable type as in the answer), which would send the tokens from your contract address to the specified recipient.