**JoyCastle前端岗位笔试题**

感谢您关注乐城堡科技！如下是测试岗位的笔试题，测试时间两天之内。如无不便，请您使用 github.com 提交答案，回复 git 代码库地址即可。否则，请使用 zip 压缩包以邮件附件形式提交代码。

1. 1)实现 move(GameObjct gameObject, Vector3 begin, Vector3 end, float time, bool pingpong){
}
使 gameObject 在 time 秒内，从 begin 移动到 end，若 pingpong 为 true，则在结束时
使 gameObject 在 time 秒内从 end 移动到 begin，如此往复。

1. 在上题基础上实现 easeIn easeOut easeInOut 动画效果

|  |
| --- |
| using System.Collections;  using System.Collections.Generic;  using UnityEngine;  using DG.Tweening;  public class Move : MonoBehaviour  {  // Start is called before the first frame update  void Start()  {  move(gameObject, Vector3.zero, Vector3.one, 1f, false);  }  private void Update()  {  if (Input.GetKeyDown(KeyCode.A))  {  StopAllCoroutines();  move(gameObject, Vector3.zero, Vector3.one, 1f, false);  }else if (Input.GetKeyDown(KeyCode.B))  {  StopAllCoroutines();  EaseIn(gameObject, Vector3.zero, Vector3.one, 1f, false);  }  else if (Input.GetKeyDown(KeyCode.C))  {  StopAllCoroutines();  EaseOut(gameObject, Vector3.zero, Vector3.one, 1f, true);  }  else if (Input.GetKeyDown(KeyCode.D))  {  StopAllCoroutines();  EaseInOut(gameObject, Vector3.zero, Vector3.one, 1f, true);  }  }  void move(GameObject gameObject, Vector3 begin, Vector3 end, float time, bool pingpong)  {  StartCoroutine(moveIEnumerator(gameObject, Vector3.zero, Vector3.one, time, pingpong));  }  IEnumerator moveIEnumerator(GameObject gameObject, Vector3 begin, Vector3 end, float time, bool pingpong)  {  gameObject.transform.position = begin;  if (pingpong)  {  while (true)  {  gameObject.transform.DOMove(end, time).OnComplete(() =>  {  gameObject.transform.DOMove(begin, time);    });  yield return new WaitForSeconds(time \* 2);  }  }  else  {  gameObject.transform.DOMove(end, time);  }  }  void EaseIn(GameObject gameObject, Vector3 begin, Vector3 end, float time, bool pingpong)  {  StartCoroutine(EaseInIEnumerator(gameObject, Vector3.zero, Vector3.one, time, pingpong));  }  IEnumerator EaseInIEnumerator(GameObject gameObject, Vector3 begin, Vector3 end, float time, bool pingpong)  {  gameObject.transform.position = begin;  if (pingpong)  {  while (true)  {  gameObject.transform.DOMove(end, time).SetEase(Ease.InSine).OnComplete(() =>  {  gameObject.transform.DOMove(begin, time).SetEase(Ease.InSine);  });  yield return new WaitForSeconds(time \* 2);  }  }  else  {  gameObject.transform.DOMove(end, time).SetEase(Ease.InSine);  }  }  void EaseOut(GameObject gameObject, Vector3 begin, Vector3 end, float time, bool pingpong)  {  StartCoroutine(EaseOutIEnumerator(gameObject, Vector3.zero, Vector3.one, time, pingpong));  }  IEnumerator EaseOutIEnumerator(GameObject gameObject, Vector3 begin, Vector3 end, float time, bool pingpong)  {  gameObject.transform.position = begin;  if (pingpong)  {  while (true)  {  gameObject.transform.DOMove(end, time).SetEase(Ease.OutSine).OnComplete(() =>  {  gameObject.transform.DOMove(begin, time).SetEase(Ease.OutSine);  });  yield return new WaitForSeconds(time \* 2);  }  }  else  {  gameObject.transform.DOMove(end, time).SetEase(Ease.OutSine);  }  }  void EaseInOut(GameObject gameObject, Vector3 begin, Vector3 end, float time, bool pingpong)  {  StartCoroutine(EaseInOutIEnumerator(gameObject, Vector3.zero, Vector3.one, time, pingpong));  }  IEnumerator EaseInOutIEnumerator(GameObject gameObject, Vector3 begin, Vector3 end, float time, bool pingpong)  {  gameObject.transform.position = begin;  if (pingpong)  {  while (true)  {  gameObject.transform.DOMove(end, time).SetEase(Ease.InOutSine).OnComplete(() =>  {  gameObject.transform.DOMove(begin, time).SetEase(Ease.InOutSine);  });  yield return new WaitForSeconds(time \* 2);  }  }  else  {  gameObject.transform.DOMove(end, time).SetEase(Ease.InOutSine);  }  }  } |

2. 请从上到下依次打印出一颗二叉树的最左侧节点。

例：

2

/ \

11 23

/ \ / \

10 15 7 14

\

12

/

13

[2,11,10,12,13]

|  |
| --- |
| #include<iostream>  #include<vector>  using namespace std;  struct Node {  char data;  struct Node\* left, \*right;  };  int deepth;  Node\* addNewNode(char data) {  Node\* newNode = new Node;  newNode->data = data;  newNode->left = newNode->right = NULL;  return newNode;  }  void BFS(Node\* root,vector<vector<Node\*>> &v,int deep) {  if (root == NULL)  return;  v[deep].emplace\_back(root);  //cout << "BFS新增" << deep << "层元素：" << root->data << endl;  deep++;  BFS(root->left, v, deep);  BFS(root->right, v, deep);  }  int getDeepth(Node\* root) {  if (root == NULL) {  return 0;  }  int left = getDeepth(root->left);  int right = getDeepth(root->right);  return max(left+1,right+1);  }  int main() {  Node\* root = addNewNode('A');  root->left = addNewNode('B');  root->right = addNewNode('C');  root->left->left = addNewNode('D');  root->right->left = addNewNode('E');  root->right->right = addNewNode('F');  root->right->left->right = addNewNode('G');  deepth = getDeepth(root);  vector<vector<Node\*>> vec(deepth, vector<Node\*>());  BFS(root, vec,0);  //for (int i = 0; i < deepth; i++) {  // for (int j = 0; j < vec[i].size(); j++) {  // cout << vec[i][j]->data << " ";  // }  // cout << endl;  //}  for (int i = 0; i < deepth; i++) {  cout << vec[i][0]->data<<" ";  }  return 0;  } |

1. 请详述使用过或编写过的UI框架，画出结构图或者类图。

感谢您的时间，祝好运！