Making of Gelbeeck Continued

Node Movement Part 4

In GameBoard Script

```
public class GameBoard : MonoBehaviour {
    private static int boardWidth = 28;
    private static int boardHeight = 36;

    public GameObject [,] board = new GameObject[boardWidth, boardHeight];
    // Use this for initialization
    void Start () {

        Object[] objects = GameObject.FindObjectsOfType(typeof(GameObject));
        /*FindObjectsofType will search the whole scene for any type of objects
        and place them in the Object[] array*/

        foreach(GameObject o in objects){//looping through only gameobjects
            Vector2 pos = o.transform.position; // using global position not local position.

            if(o.name != "PacMan"){
                board[(int)pos.x, (int)pos.y]=o; // add the gameObject to this position of the array
            }else{

                 Debug.Log{"Found PacMan at: "+pos}; //pointless
            }
```

In PacMan Script

```
private Node CurrentNode; //stores PacMan's current NodePosition.

    // Use this for initialization
    void Start () {
        Node node = GetNodeAtPosition(transform.localPosition);

        if(node != null){
            CurrentNode = node;
            Debug.Log(CurrentNode);
        }
    }


Node GetNodeAtPosition (Vector2 pos){
        GameObject tile = GameObject.Find("Game:").GetComponent<GameBoard>().board[(int)pos.x,(int)pos.y];
        //Find the GameObject named "Game" in the scene and get it's GameBoard component,
        //pass a position into it's array then store it into tile

        if(tile != null){
            return tile.GetComponent<Node>();
            //return the Node Component
        }
        return null;

    }
```

Part 2

```
Node CanMove(Vector2 d){
        Node moveToNode = null;

        for(int i = 0; i< CurrentNode.neighbours.Length;i++){
        //looping through each array element and checking if it's direction matches the direction
        //PacMan wants to go.
            if(CurrentNode.validDirections [i]==d){
                moveToNode = CurrentNode.neighbours[i];
                break;
            }

        }
    void MoveToNode(Vector2 d){
        Node moveToNode = CanMove(d);

        if(moveToNode != null){
```

```
            transform.localPosition = moveToNode.transform.position;
            CurrentNode = moveToNode;
            }
        }
```

Underneath every keypress, add the line MoveToNode(direction) to Make PacMan move to that Node.

```
void CheckInput(){
        if(Input.GetKeyDown(KeyCode.LeftArrow)){
            direction =Vector2.left;
            MoveToNode(direction);
        }
```