Lab - AI

Goals

- To learn how to use provided implementation of A* algorithm for pathfinding.

Resources

- Pathfinding project (https://altitude.otago.ac.nz/cosc360/Pathfinding) on GitLab.
- Tutorial on A* algorithm (http://www.raywenderlich.com/4946/introduction-to-a-pathfinding).

Basic challenge

The most fundamental and common problem in many games is pathfinding. The basic challenge is about learning how to use provided pathfinding code.

Pathfinding is one of the most fundamental elements of game AI, and it's quite useful in many types of games. The implementation provided in this lab is derived from Simply A* (https://www.assetstore.unity3d.com/en/#!/content/6385), a free pathfinding package from the Unity Asset

Store. However, whereas Simply A* was designed for 3D pathfinding, with some (somewhat buggy) support for 2D, the code in this lab has been rewritten and simplified specifically for 2D pathfinding. The objective of this lab is not necessarily to explain the details of that implementation, but rather to demonstrate how to use the provided code in you game, so that you can use it for your pathfinding needs.

The grid

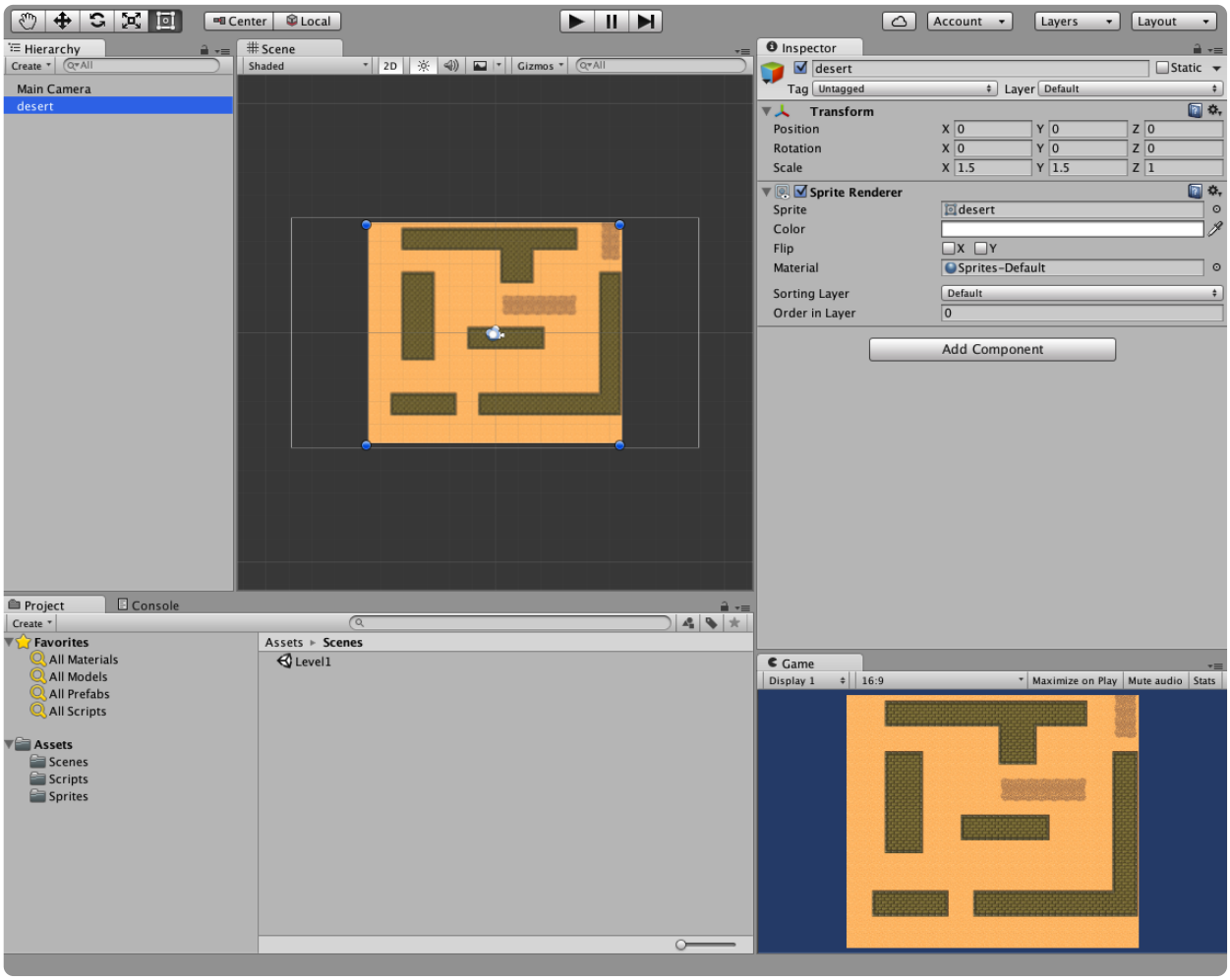
. Fork the Pathfinding project (https://altitude.otago.ac.nz/cosc360/Pathfinding) on GitLab, then clone your fork to your hard-drive.

. Open the *Pathfinding* project, the directory of the cloned repository, in Unity.

. Set your project workspace so that you can see both the "Scene" and "Game" windows at the same time.

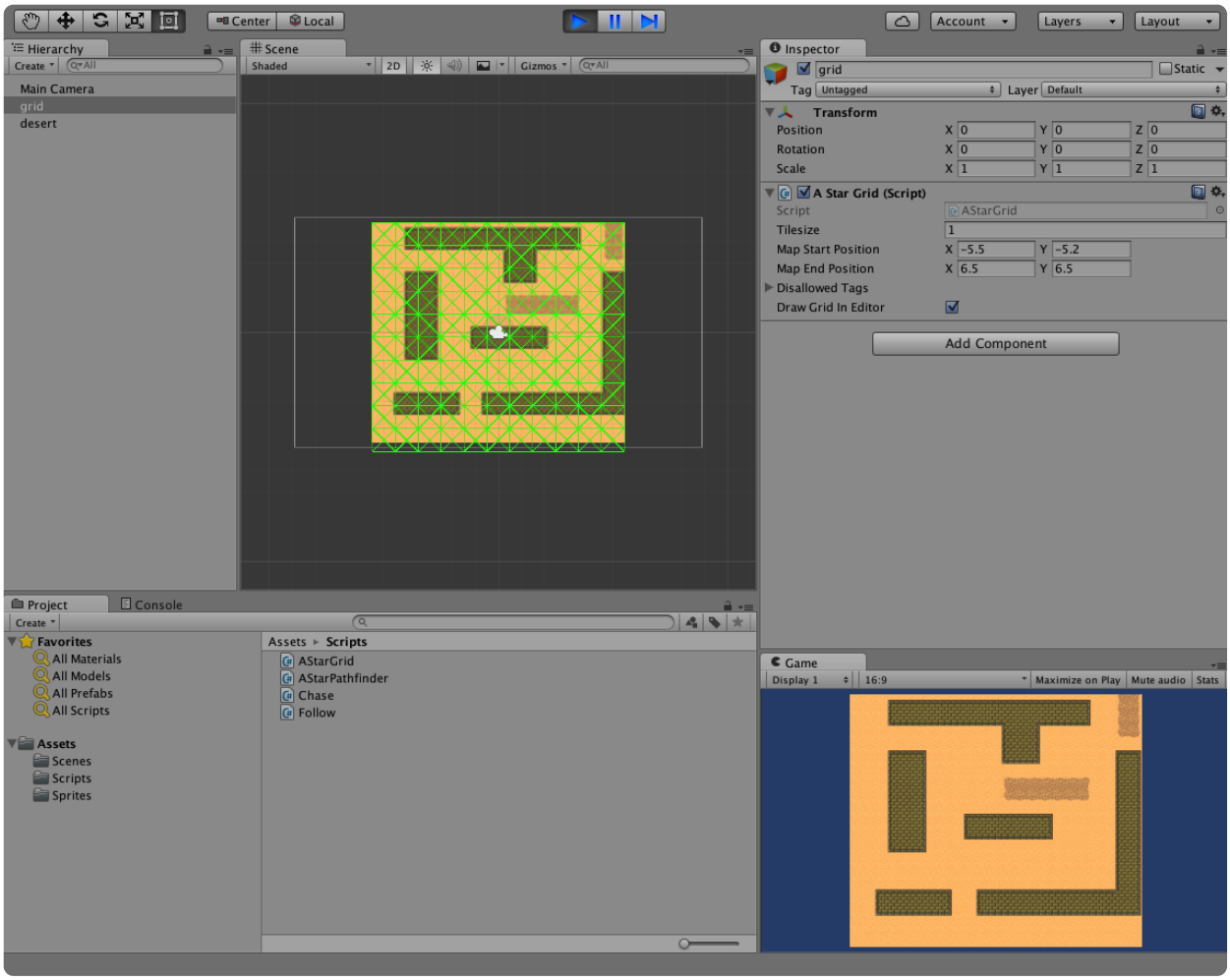
. Save the scene as "Level1".

. From *Assets/Sprites*, drag the "desert" sprite into the **Hierarchy panel** - should create game object of the same name. Set the Transform component of the "desert" game object as shown in the screenshot below.



. Create an empty game object, name it "grid" and add the "AStarGrid" script as its component (the script is already there in your assets). The pathfinding algorithm in this lab operates on a regular grid, and "AStarGrid" creates this grid. You can specify the starting and end point of the grid, as well as the tile size. To make the grid span the "desert" background image, set the attributes of the "A Star Grid" component as follows: TileSize = 1, MapStartPosition = (-5.5, -5.2), MapEndPosition(6.5, 6.5).

. Enable the "Draw Grid In Editor" flag of the "A Star Grid" component and press play. The grid should be drawn in your "Scene" window, as shown in the screenshot below:



The nodes are waypoints that pathfinder uses to make the path. The lines between waypoints give possible connections. At the moment all nodes are connected with their neighbouring nodes.

Details of the implementation of pathfinding are not going to be explained in this lab. However, you can take a look at the code, and read through the comments if you wish to understand it and/or modify it.

The tiles of the grid are squares and the grid is created so that there is an integer number of tiles. As a result, the grid doesn't cover the background perfectly. If you're using this pathfinding script in your game later on, you'll probably need to spend a bit more time, making sure the grid covers the background right. For this lab however, this doesn't need to be perfect, so just leave it as it is.

Chasing

Once you have the pahtfiniding grid setup, here's how to get a game object to move on it.

. Drag the "packman" and "ghost" sprites from *Assets* / *Sprites* to the **Hierarchy panel** - corresponding name sprite game object should get created. Set the "pacman" Transform attribute to: Position = (-4, -4, -1), Scale = (2.5, 2.5, 1), and "ghost" Transform to: Position = (3.5, 2.5, -2), Scale = (2.5, 2.5, 1).

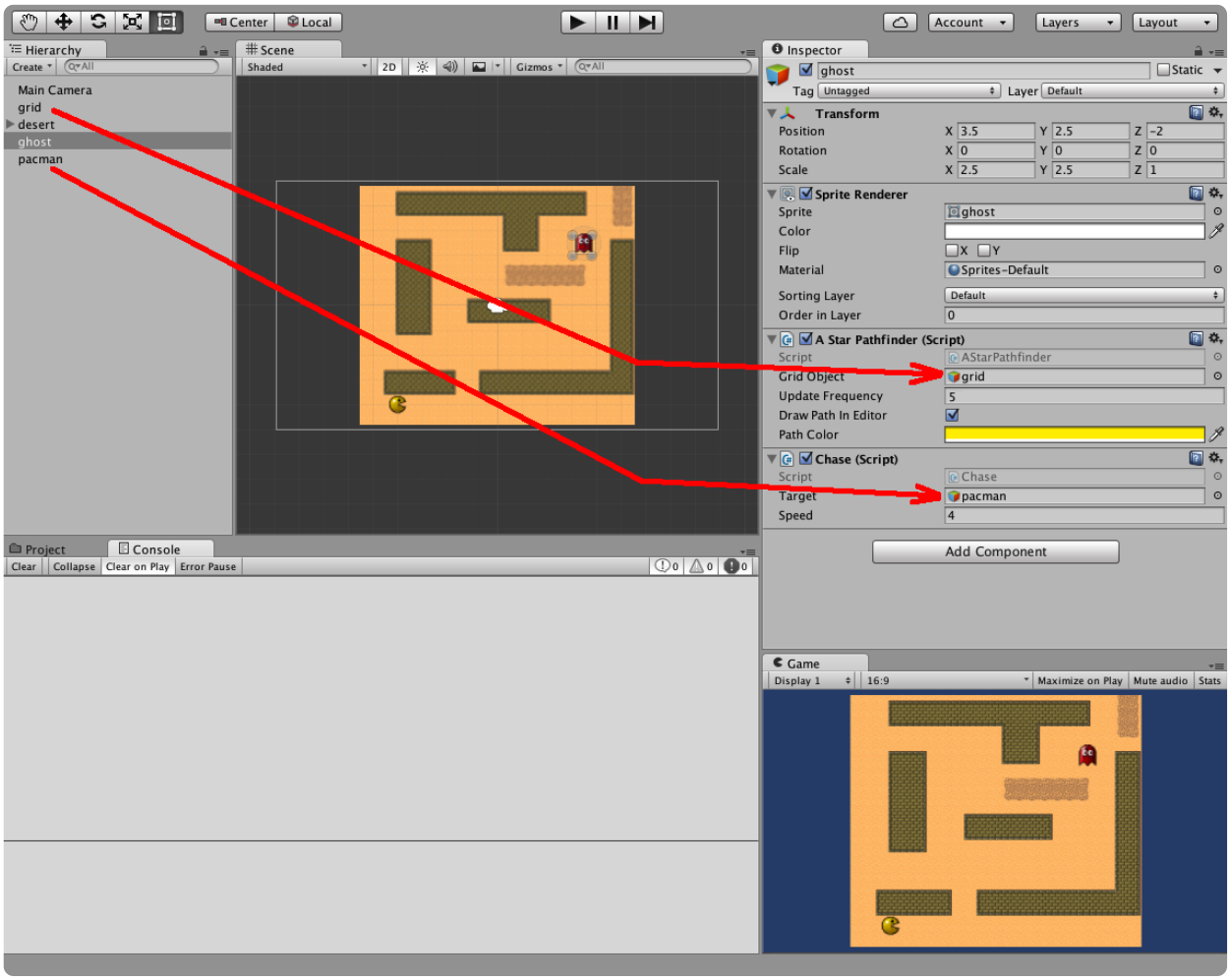
. Add the "AStarPathfinder" script component to the "ghost" game object. This script is responsible for finding the path and moving an object along it at specified speed. The movement is done by setting of the position

of the object's transform - it should be relatively easy to tweak the script to do movement using forces, if that's what's required for your game.

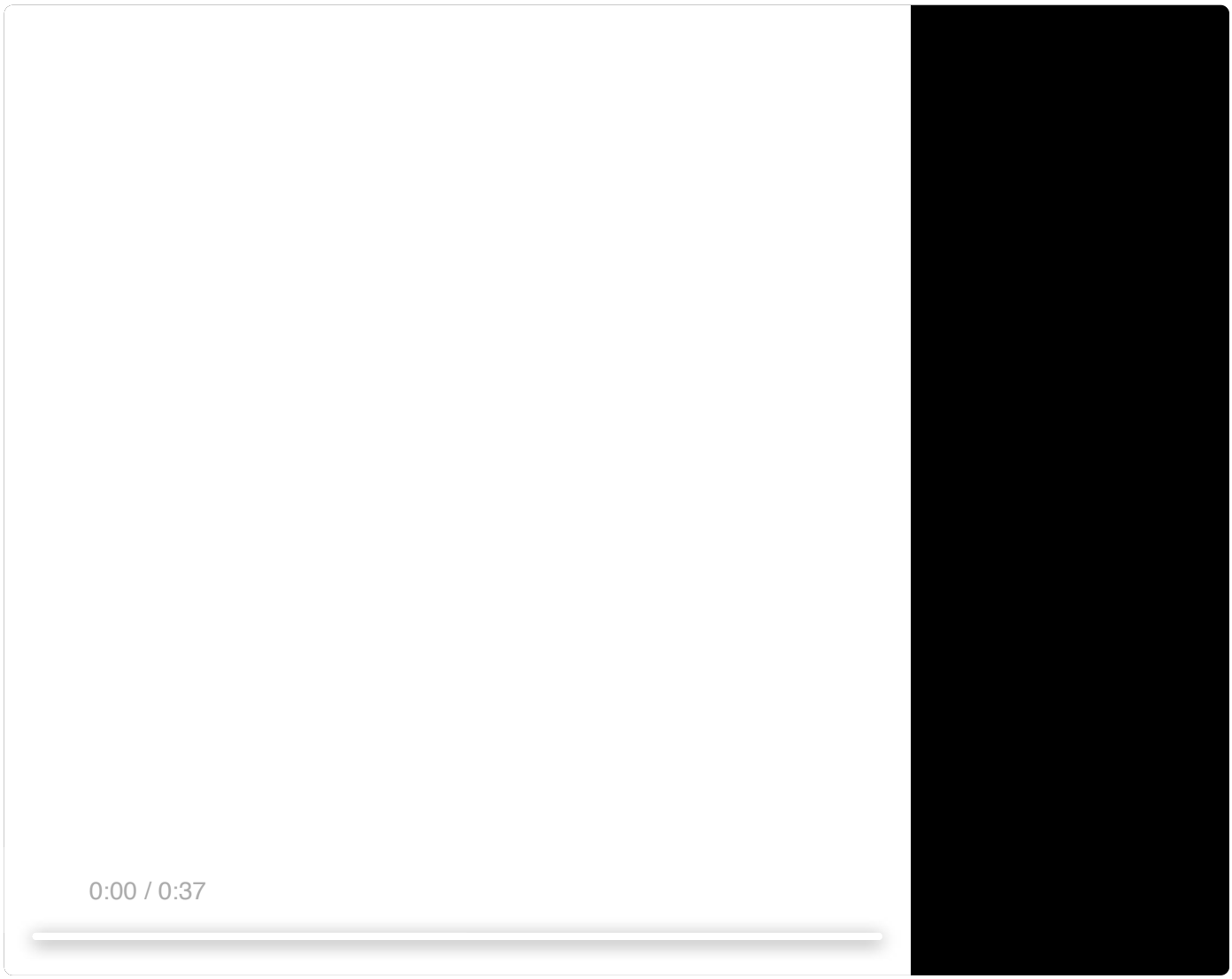
9. The "AStarPathfinder" works on a grid, and so you need to associate it with a grid game object. Drag the "grid" game object to "AStarPathfinder"s *Grid Object* attribute.

10. The "AStarPathfinder" script updates the path at certain frequency. Pathfinding takes some computer resources, and it can often be updated at an interval that is slower than the frame rate. The *Update Frequency* attribute of the *AStarPathfinder* controls this frequency. It specifies number of updates per second. The default is 5 - leave this number as it is.

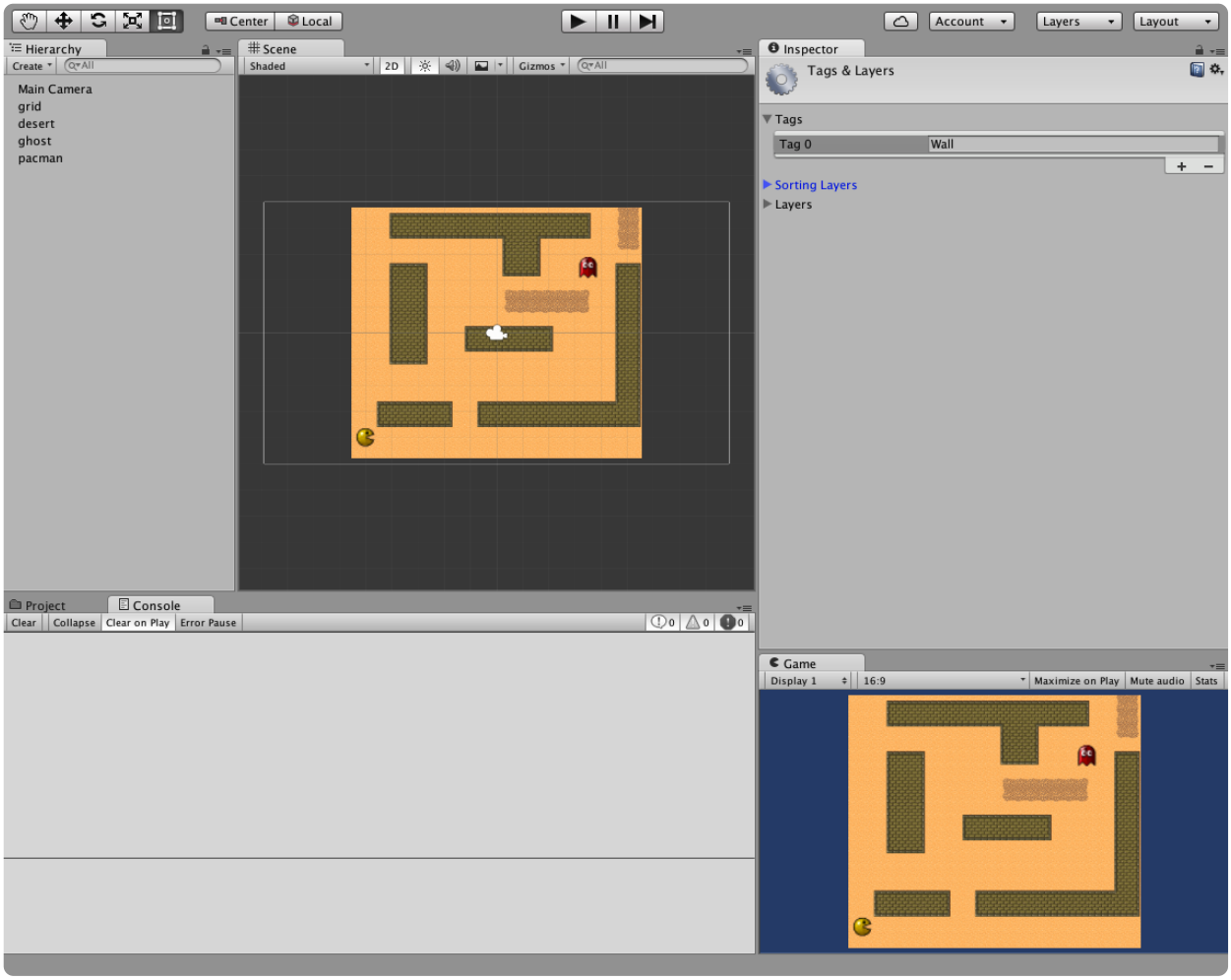
11. Add the "Chase" script as a component of the "ghost" game object. This is a very simple script, which on each update call "AStarPathfinder"'s goTowards method to moves towards a target at specified speed. Make the "pacman" game object the target by dragging it to "Chase" script component's *target* attribute. Set *Speed* to 4.



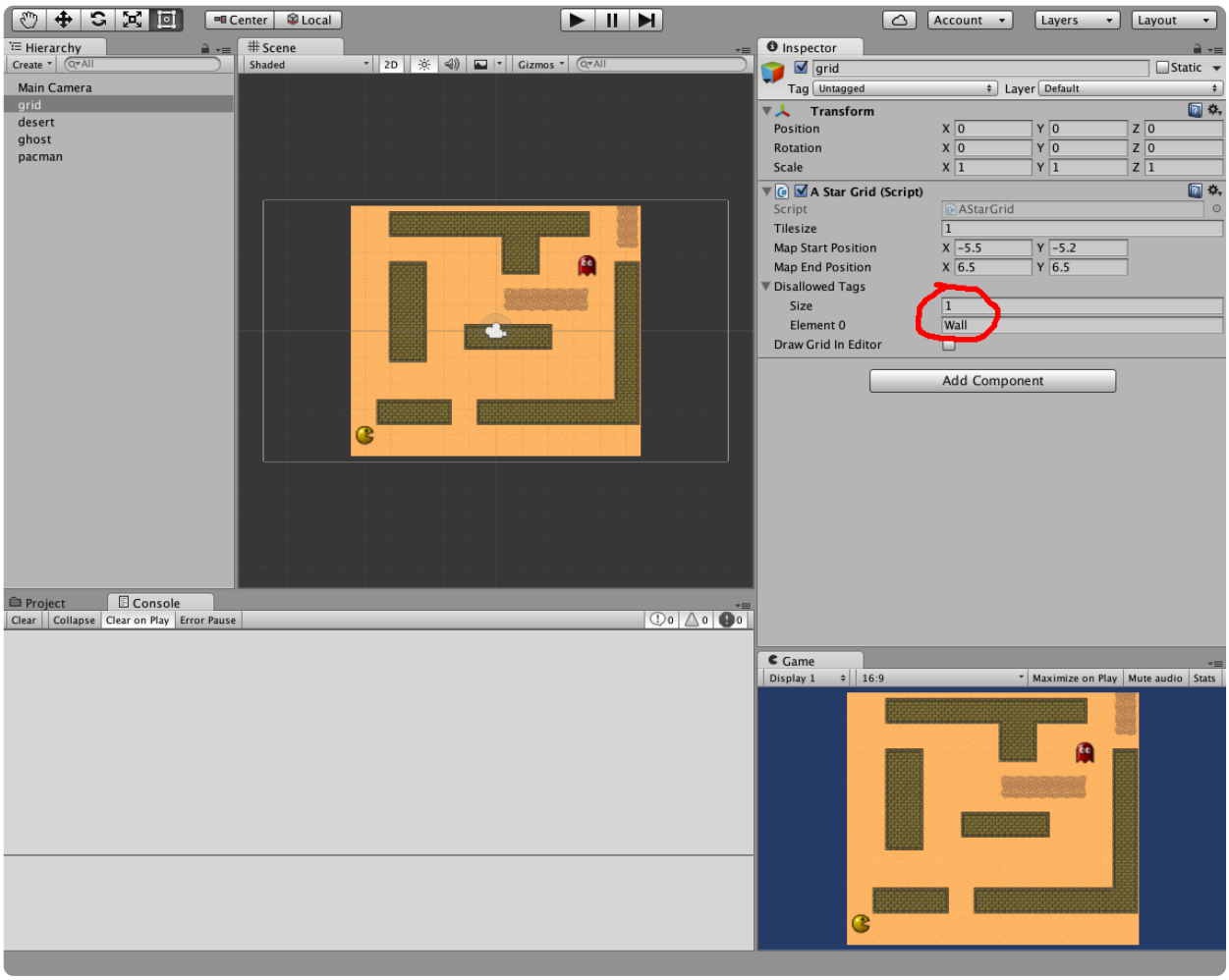
12. Press play. The ghost should travel towards the "pacman" game object.



1. The "ghost" ignores the walls of the background image, because the grid is not aware of those walls. From the main menu select *Edit | Project Settings | Tags and Layers* and create a new tag name "Wall".

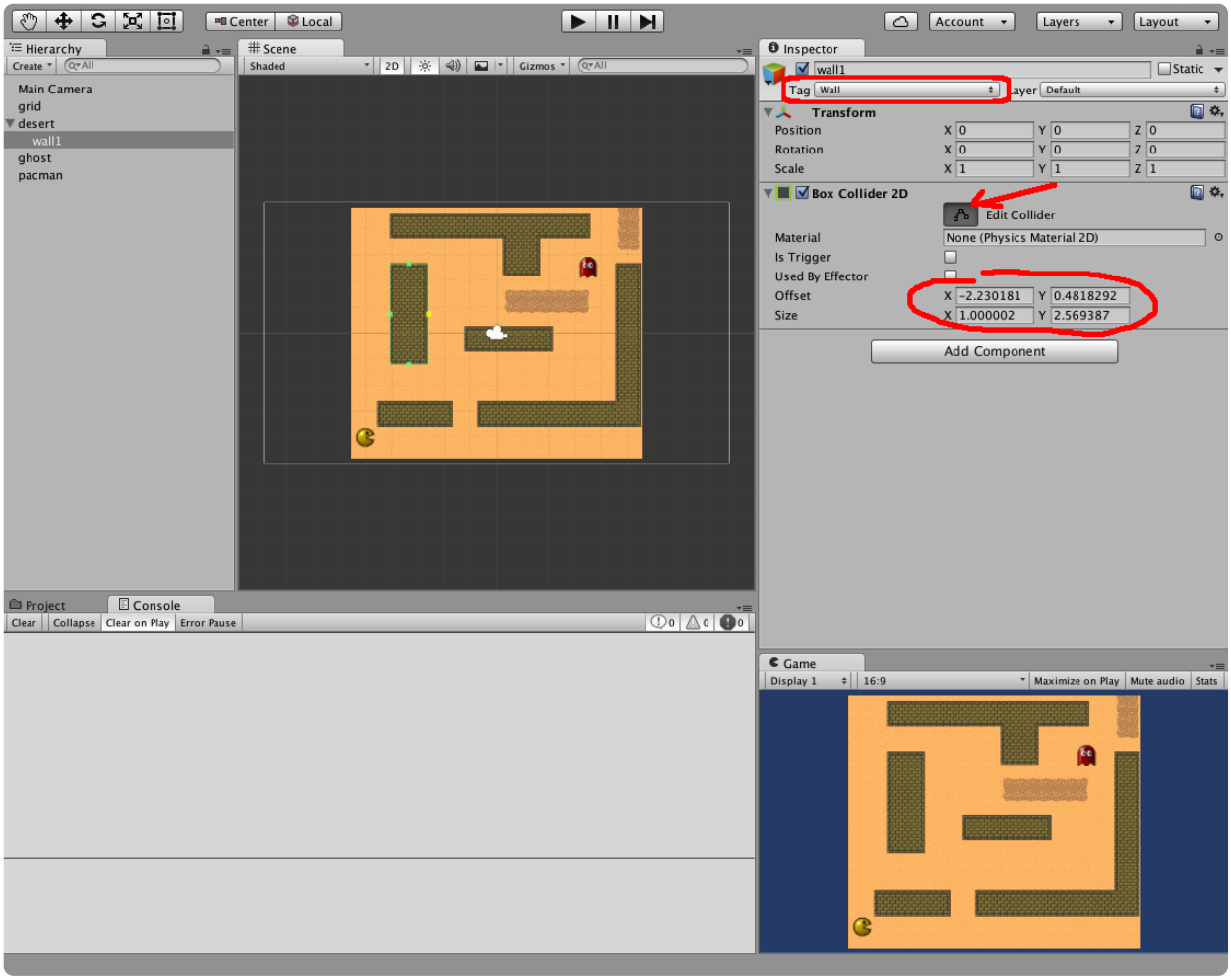


; Select the "grid" object, and specify "Wall" as one of the disallowed tags: expand the *Disallowed Tags* attribute of the "A Star Grid" component, set its *Size* to 1 and set *Element 0* to "Wall".

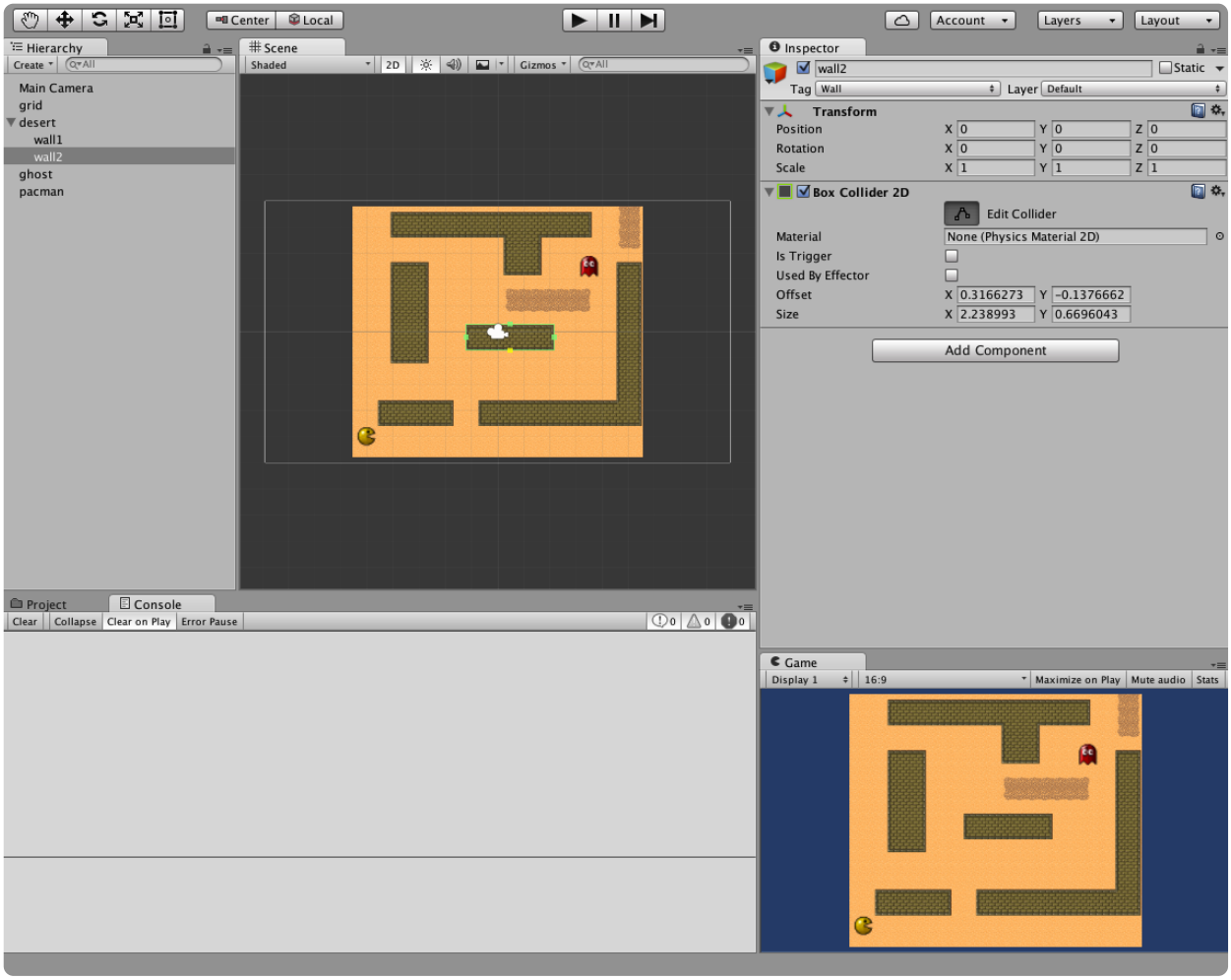


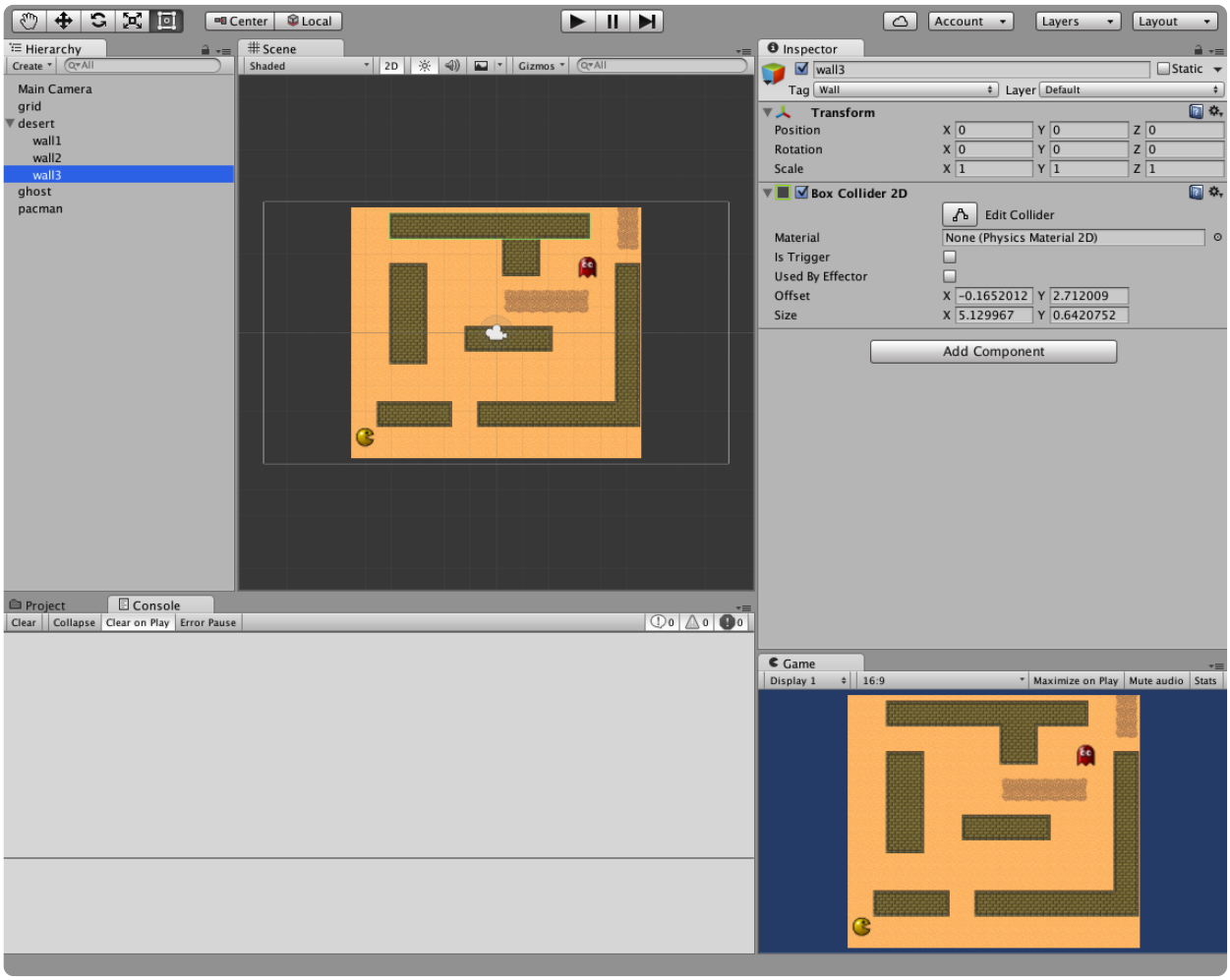
3. Under the "desert" game object create an empty game object, name it "wall1" and set its Tag to "Wall".

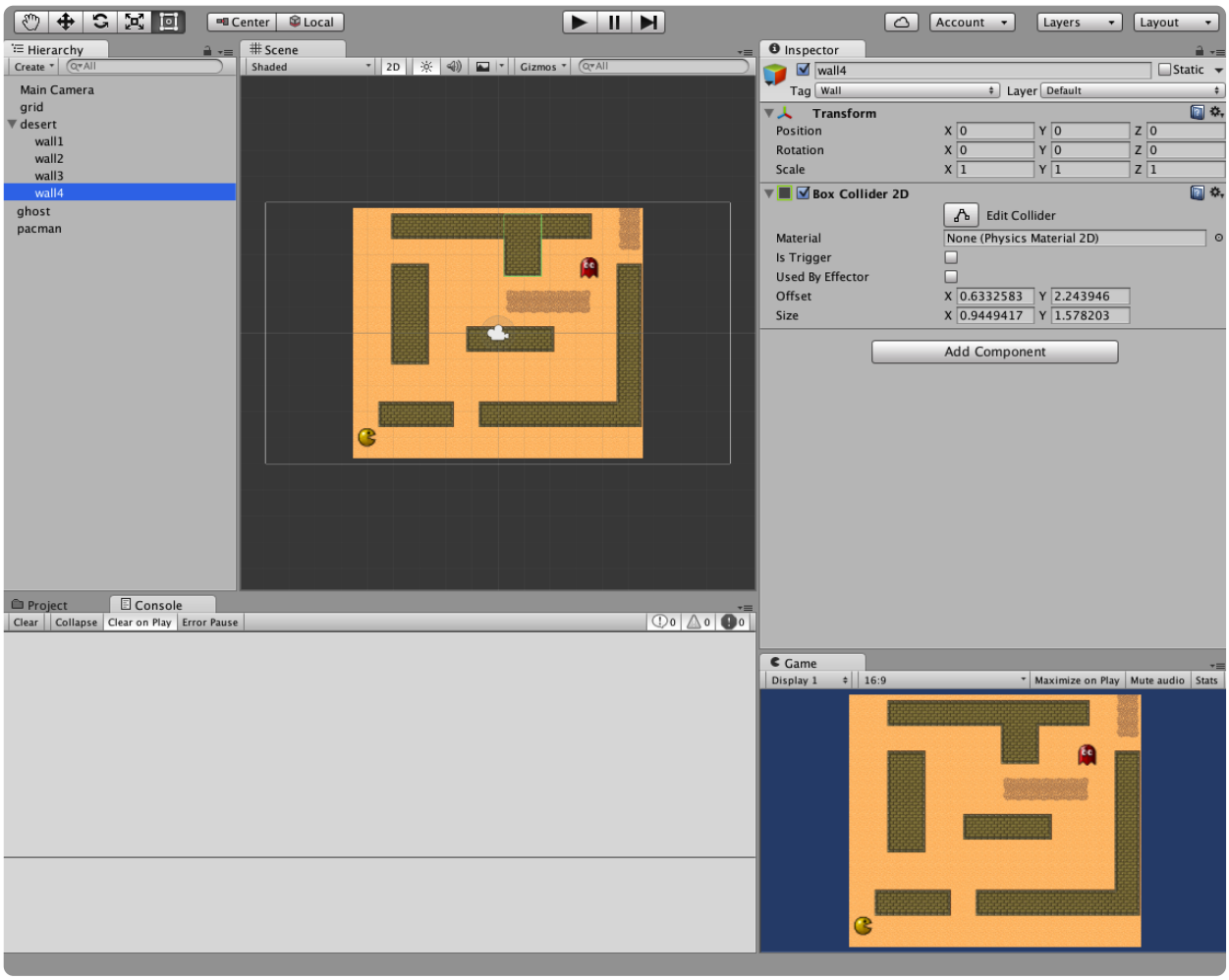
4. Add a BoxCollider2D component to the "wall1" game object. Click the "Edit Collider" button and drag the collider to cover the wall of the "desert" sprite as in the image below:

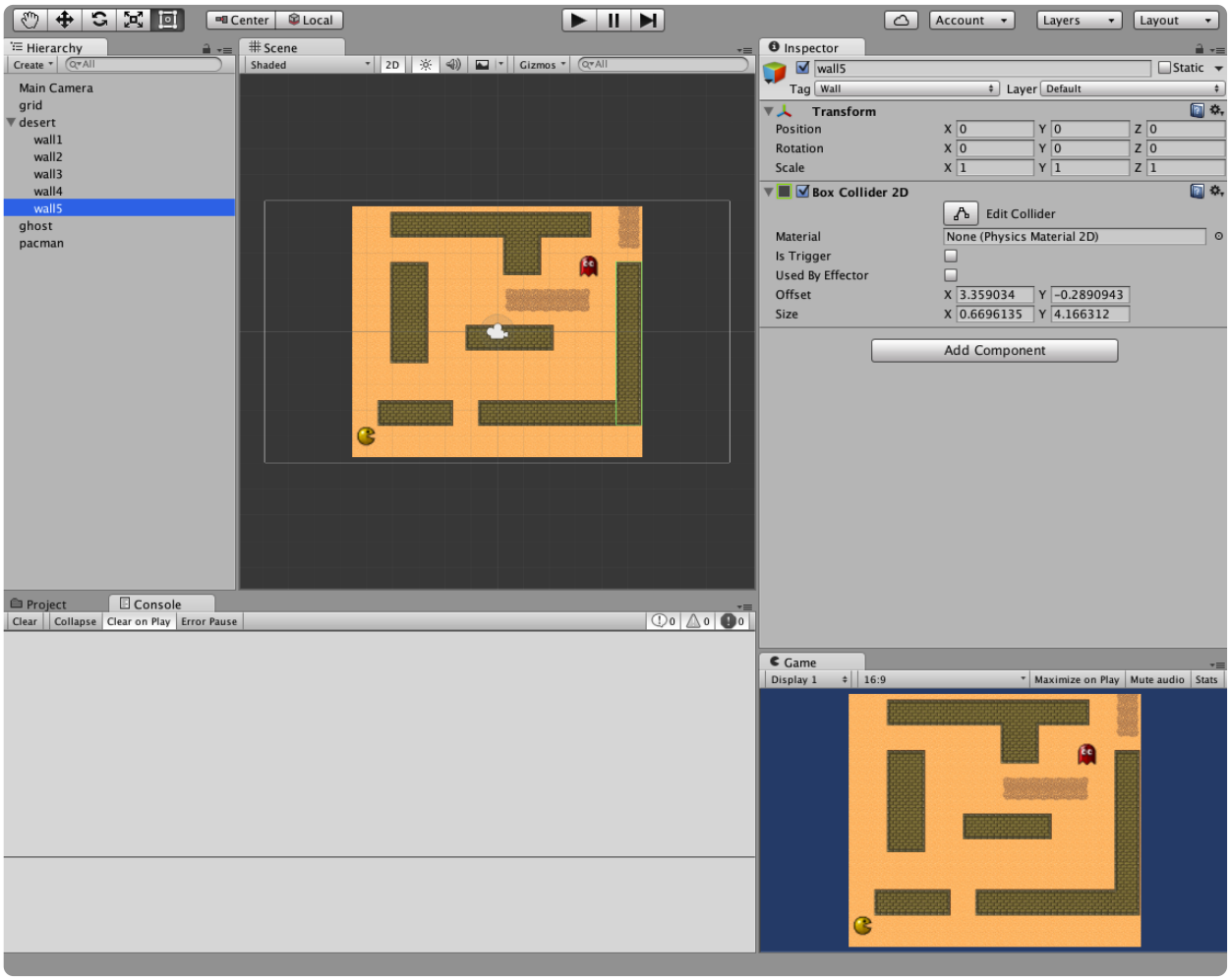


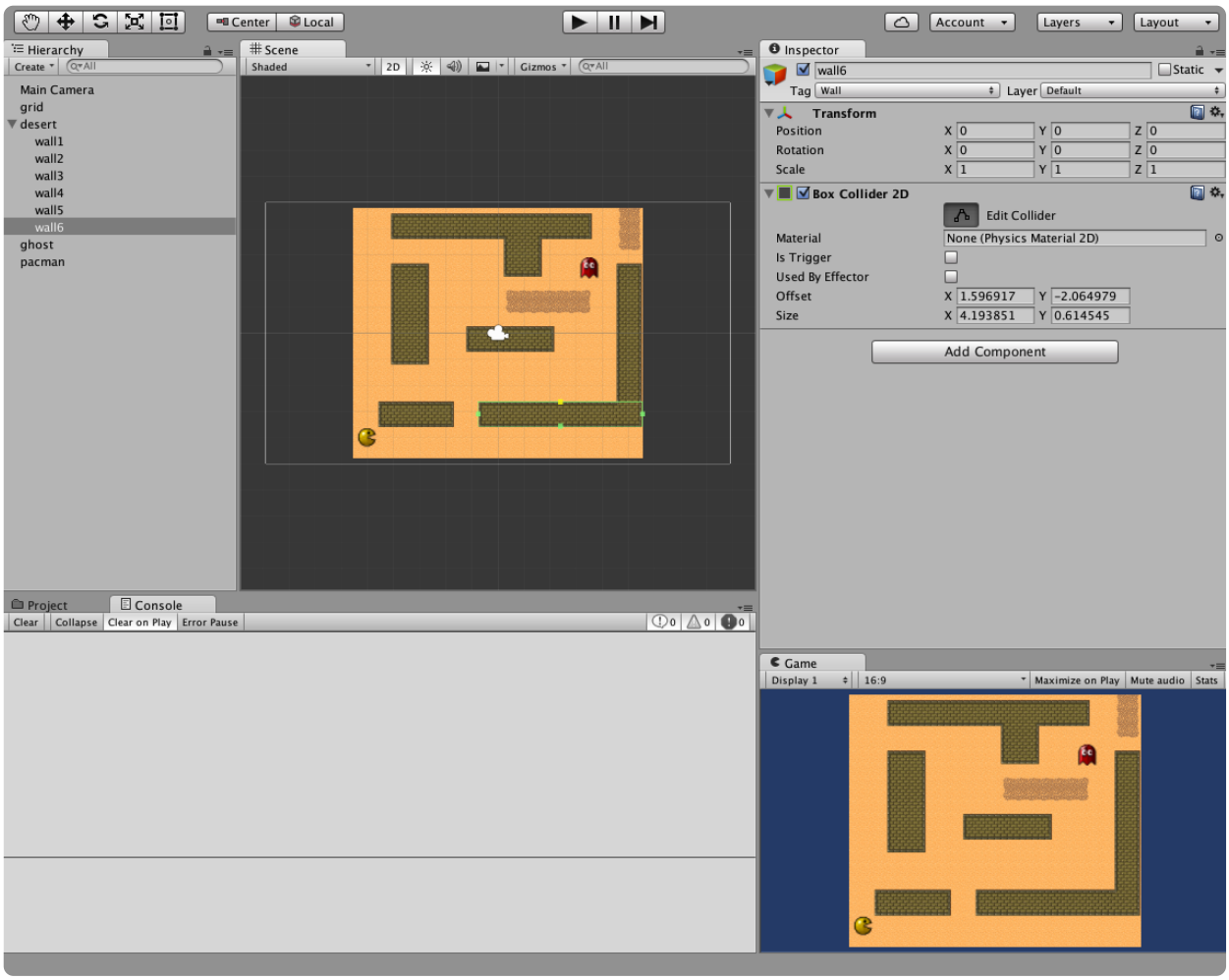
3. Duplicate "wall1", name the new object "wall2" and adjust its collider to cover a different wall. In all create a total of 7 walls, with colliders configured like in the figures below:

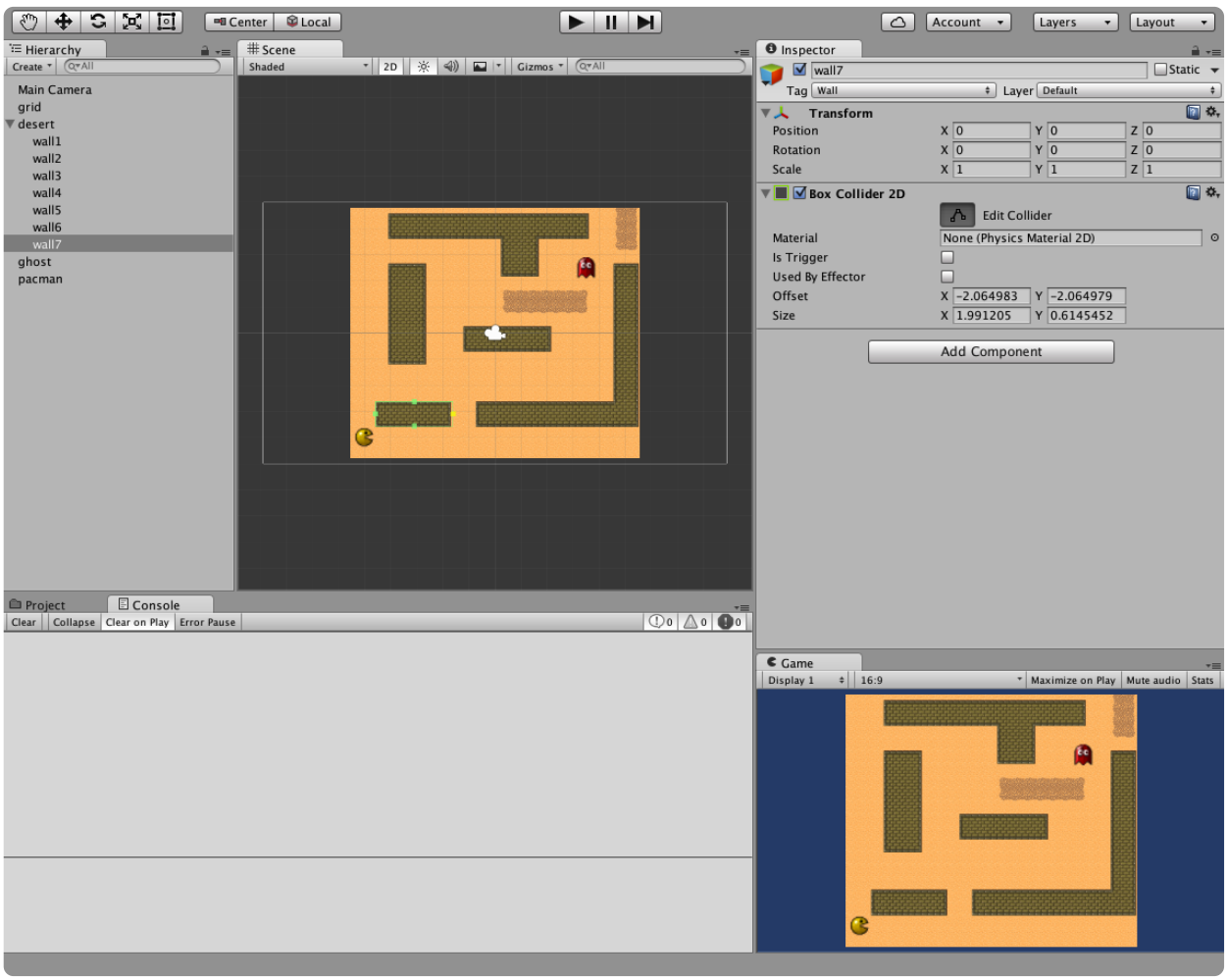

Center
Local
Account
Layers
Layout
Hierarchy
Create
Main Camera
grid
desert
wall1
wall2
ghost
pacman
Scene
Shaded
2D
Gizmos
Inspector
wall2
Static
Tag
Wall
Layer
Default
Transform
Position
Rotation
Scale
Box Collider 2D
Edit Collider
Material
None (Physics Material 2D)
Is Trigger
Used By Effector
Offset
0.3166273
-0.1376662
Size
2.238993
0.6696043
Add Component
Project
Console
Clear
Collapse
Clear on Play
Error Pause
Game
Display 1
16:9
Maximize on Play
Mute audio
Stats


Center
Local
Account
Layers
Layout
Hierarchy
Create
Main Camera
grid
desert
wall1
wall2
wall3
ghost
pacman
Scene
Shaded
2D
Gizmos
Inspector
wall3
Static
Tag
Wall
Layer
Default
Transform
Position
Rotation
Scale
Box Collider 2D
Edit Collider
Material
None (Physics Material 2D)
Is Trigger
Used By Effector
Offset
-0.1652012
2.712009
Size
5.129967
0.6420752
Add Component
Project
Console
Clear
Collapse
Clear on Play
Error Pause
Game
Display 1
16:9
Maximize on Play
Mute audio
Stats

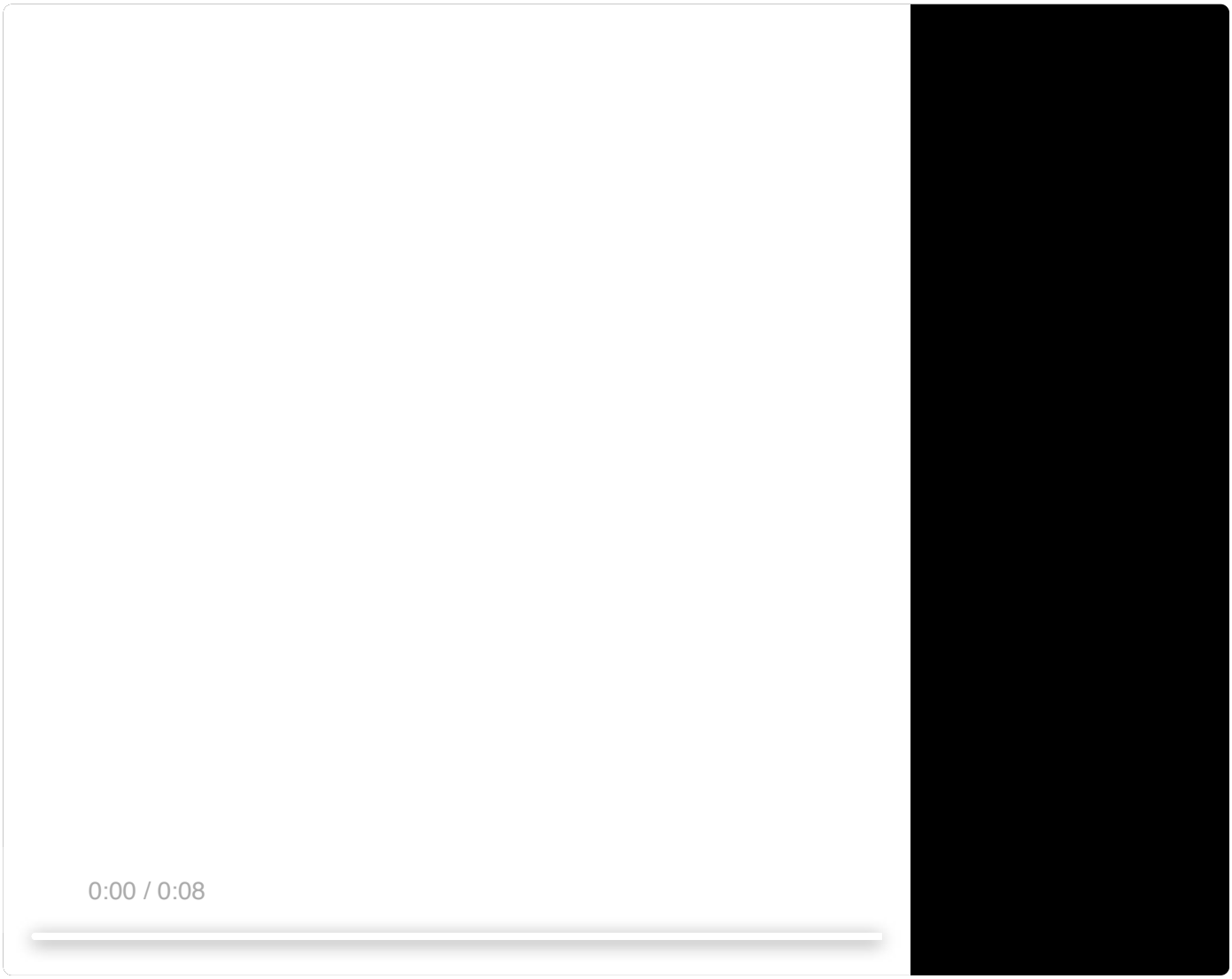

Center
Local
Account
Layers
Layout
Hierarchy
Create
Main Camera
grid
desert
wall1
wall2
wall3
wall4
ghost
pacman
Scene
Shaded
2D
Gizmos
Inspector
wall4
Static
Tag
Wall
Layer
Default
Transform
Position
Rotation
Scale
Box Collider 2D
Edit Collider
Material
None (Physics Material 2D)
Is Trigger
Used By Effector
Offset
X 0.6332583 Y 2.243946
Size
X 0.9449417 Y 1.578203
Add Component
Project
Console
Clear
Collapse
Clear on Play
Error Pause
Game
Display 1
16:9
Maximize on Play
Mute audio
Stats


Center
Local
Account
Layers
Layout
Hierarchy
Create
Main Camera
grid
desert
wall1
wall2
wall3
wall4
wall5
ghost
pacman
Scene
Shaded
2D
Gizmos
Inspector
wall5
Static
Tag
Wall
Layer
Default
Transform
Position
Rotation
Scale
Box Collider 2D
Edit Collider
Material
None (Physics Material 2D)
Is Trigger
Used By Effector
Offset
X 3.359034 Y -0.2890943
Size
X 0.6696135 Y 4.166312
Add Component
Project
Console
Clear
Collapse
Clear on Play
Error Pause
Game
Display 1
16:9
Maximize on Play
Mute audio
Stats


Center
Local
Account
Layers
Layout
Hierarchy
Create
Main Camera
grid
desert
wall1
wall2
wall3
wall4
wall5
wall6
ghost
pacman
Scene
Shaded
2D
Gizmos
Inspector
wall6
Static
Tag
Wall
Layer
Default
Transform
Position
Rotation
Scale
Box Collider 2D
Edit Collider
Material
None (Physics Material 2D)
Is Trigger
Used By Effector
Offset
Size
Add Component
Project
Console
Clear
Collapse
Clear on Play
Error Pause
Game
Display 1
16:9
Maximize on Play
Mute audio
Stats

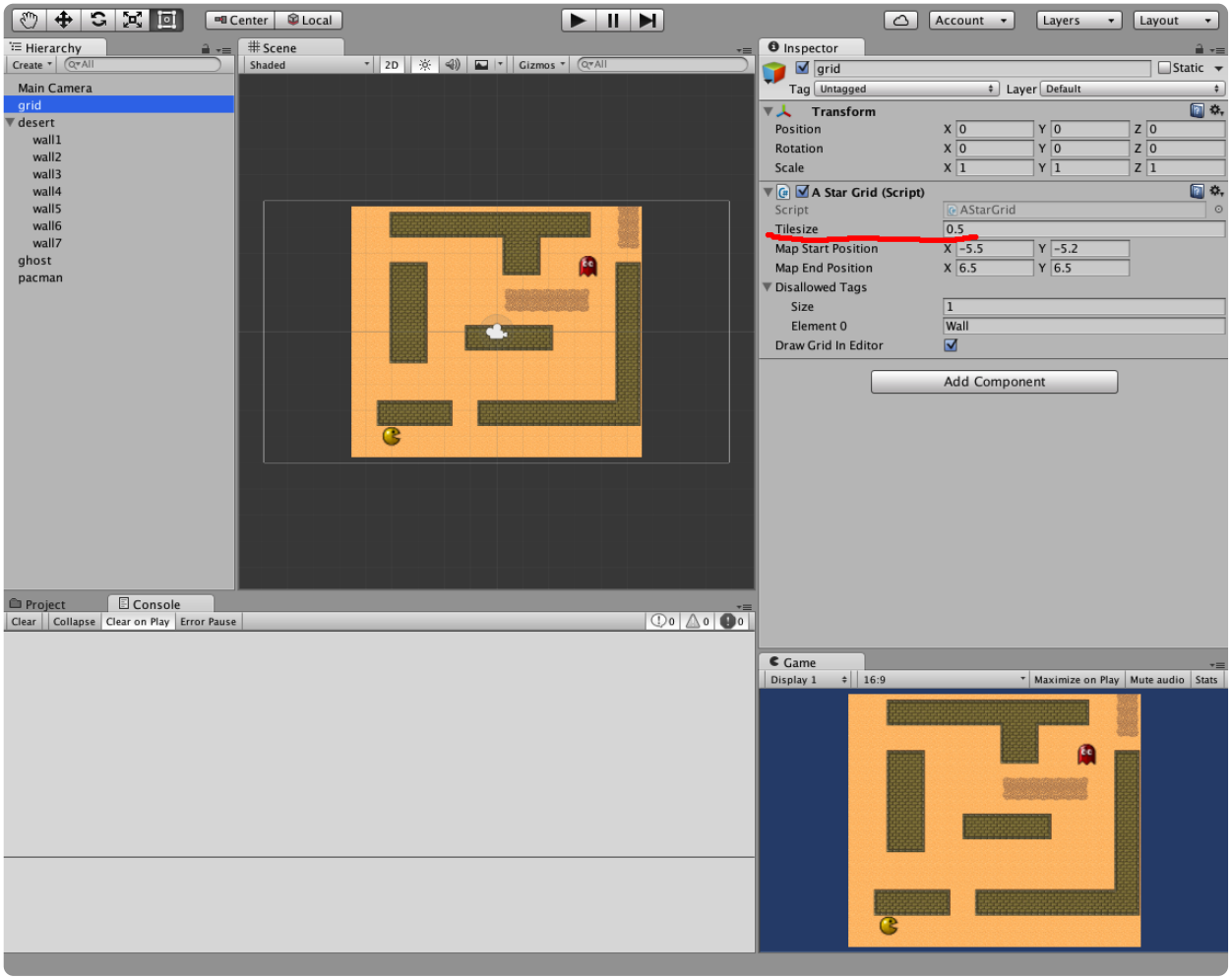

Center
Local
Account
Layers
Layout
Hierarchy
Create
Main Camera
grid
desert
wall1
wall2
wall3
wall4
wall5
wall6
wall7
ghost
pacman
Scene
Shaded
2D
Gizmos
Inspector
wall7
Static
Tag
Wall
Layer
Default
Transform
Position
Rotation
Scale
Box Collider 2D
Edit Collider
Material
None (Physics Material 2D)
Is Trigger
Used By Effector
Offset
Size
Add Component
Project
Console
Clear
Collapse
Clear on Play
Error Pause
Game
Display 1
16:9
Maximize on Play
Mute audio
Stats

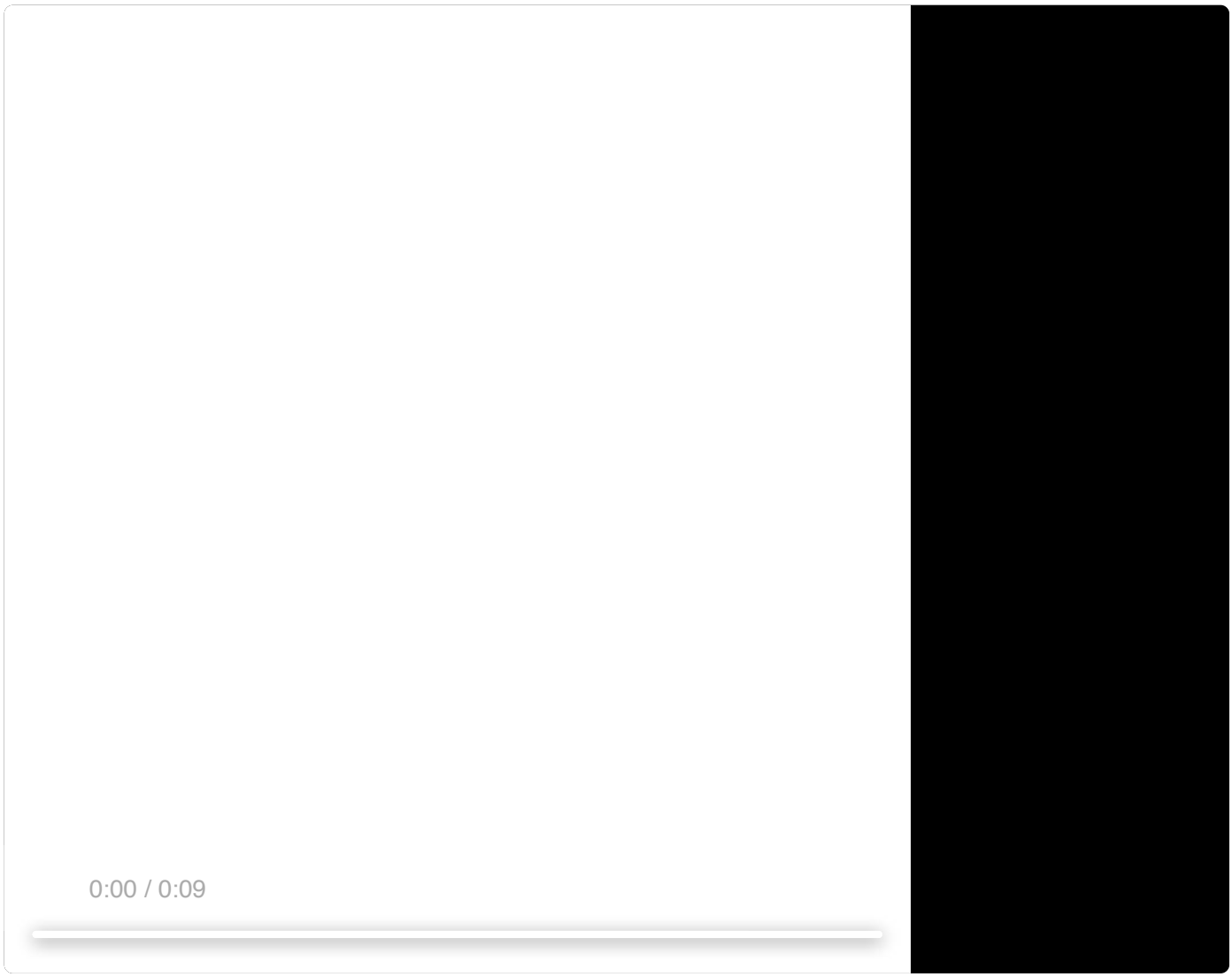
9. If you press play now, the ghost should avoid the walls.



If you take a look at the grid in the "Scene" view, you'll notice that the grid doesn't go over the walls. The "AStarGrid" script will not create a grid over colliders that belong to objects with one of the "Disallowed Tags". In this case, the disallowed tag is "Wall", which is assigned to all 7 walls.

10. Did you notice that the grid is a bit too coarse? There ghost is not taking an obvious path between two walls (when it's close to the pacman), because the grid doesn't have a path there. Select the "grid" object and set the *Tile Size* attribute of its "A Star Grid" component to 0.5. Play the game again. The grid should be finer, with more waypoints, and the ghost should take the path between the walls.


Center
Local
Account
Layers
Layout
Hierarchy
Create
Main Camera
grid
desert
wall1
wall2
wall3
wall4
wall5
wall6
wall7
ghost
pacman
Scene
Shaded
2D
Gizmos
Inspector
grid
Static
Tag
Untagged
Layer
Default
Transform
Position
Rotation
Scale
A Star Grid (Script)
Script
AStarGrid
Tilesize
0.5
Map Start Position
Map End Position
Disallowed Tags
Size
Element 0
Wall
Draw Grid In Editor
Add Component
Project
Console
Clear
Collapse
Clear on Play
Error Pause
Game
Display 1
16:9
Maximize on Play
Mute audio
Stats

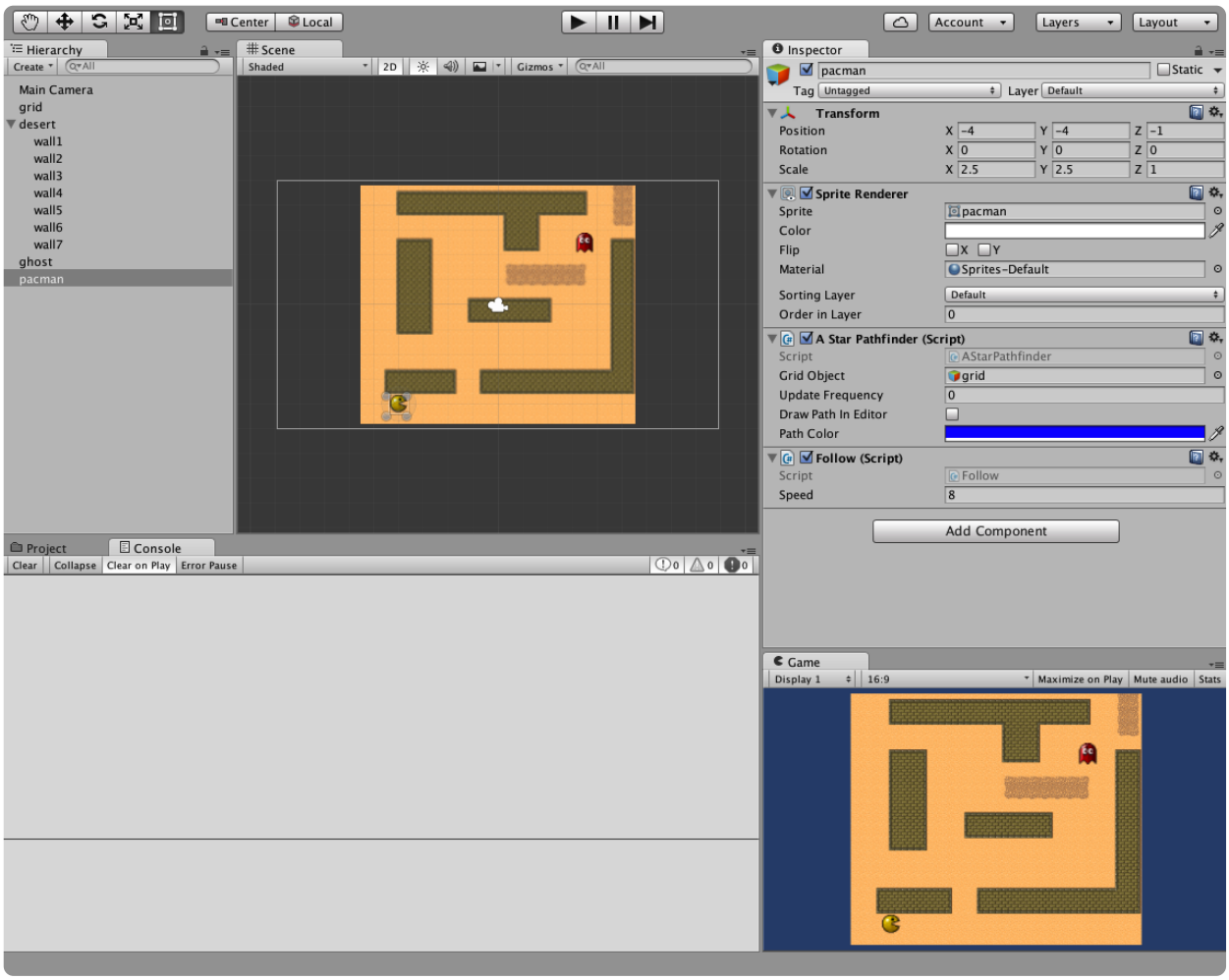


One thing to be aware of, is that the finer the grid, the more time taken for pathfinding - because paths are longer (in terms of the number of nodes), and there are more options to consider.

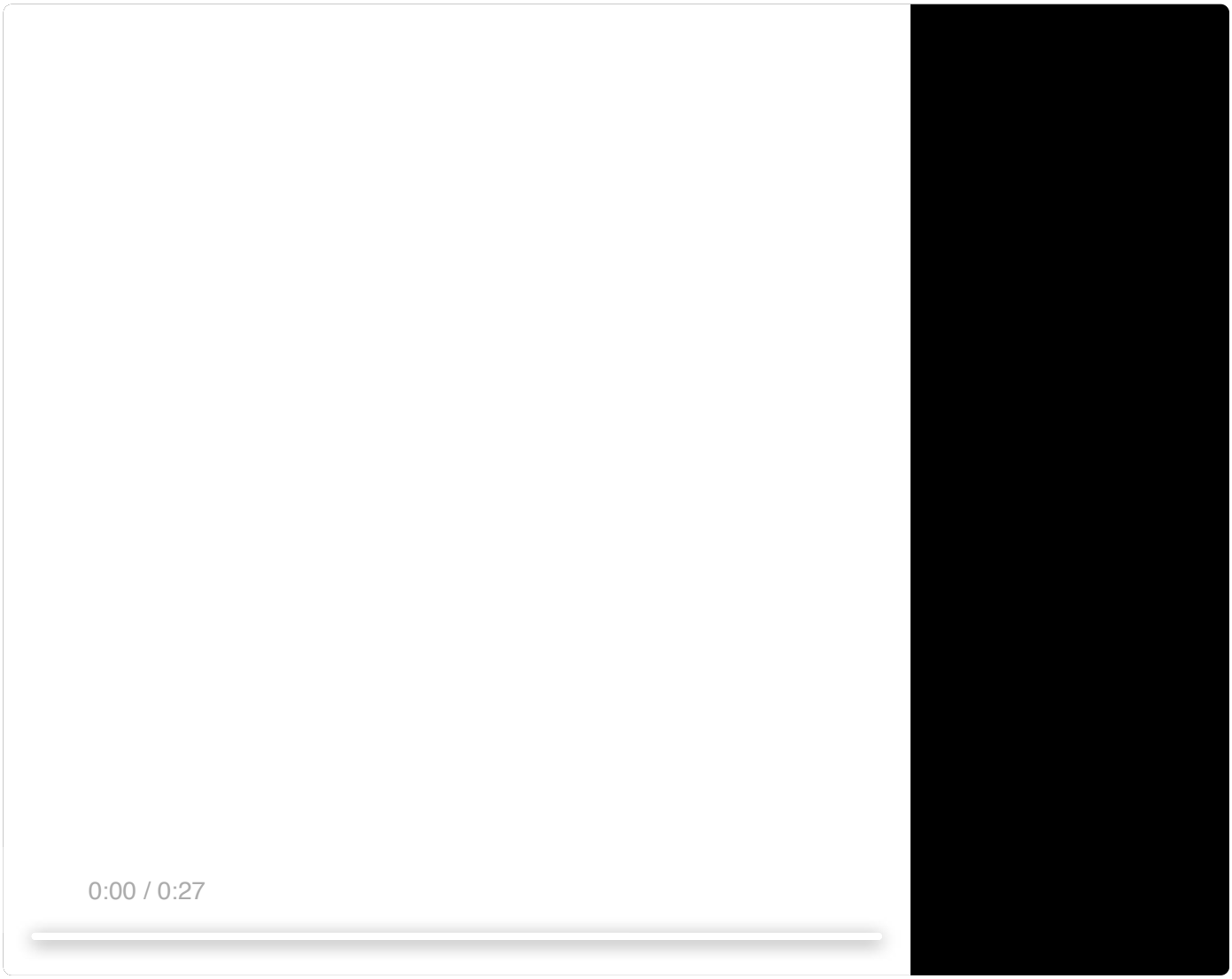
Following

Whereas the ghost chases a specific target, the pacman is going to be mad to find best path to point marked by a mouse click.

1. Select the "pacman" game object and add "AStarPathfinder" game object to it. Drag the "grid" game object over the *Grid Object* attribute of the *A Star Pathfinder* component. Set the *Update Frequency* to 0 - this means the path will be updated as soon as required. This could potentially mean an update every frame, but "AStarPathfinder" updates only its path whenever target destination changes. So, new path won't be necessarily computed per every frame.
2. Add the "Follow" script as a component to the "pacman" game object. This script is a simple script that uses "AStarPathfinder" to find the best path to position denoted by a mouse click, and moves the game object at specified speed. Set the "Speed" of the *Follow* component to 8.



3. Run the game. Click on various locations in the "Game" view - pacman should find a way to get there. If you click somewhere where pacman cannot go (outside the grid or a wall) pacman will not travel there.



That's all for the basic challenge. You can use the code from this lab in you game for pathfinding. The pathfinding algorithm uses regular grid with a Euclidean distance as the heuristic (the guess). The game objects that use pathfinding do not need to be position exactly at the node, but they must be on the walkable part of the grid. The "Chase" and "Follow" scripts demonstrate how to call the pathfinding library to get objects moving. You might need to read through "AStarPathfinder" and/or "AStarGrid" code at some point (it's all commented) and perhaps tweak it bit, to make it work for your game.

Assessment

Show your work to the demonstrator for assessment.

Intermediate Challenge

Learn how to use Unity's built in system for Navigation and Pathfinding (https://docs.unity3d.com/Manual/Navigation.html). It's a little bit more complicated than the system shown in the Basic Challenge, but it's also more powerful - it can work in 3D (and can be made to function over 2D). Create a new scene in the project form the Basic Challenge and redo the pathfinding using Unity's system.

Assessment

Show your work to the demonstrator for assessment.

Master Challenge

- Unity has no built-in framework for behaviour trees. However there are number of unofficial implementations. Find one that is free (please, refrain from the paid toolboxes on the Asset store) and learn how to use it, or develop your own framework for behaviour trees. Create a simple scene demonstrating the use of behaviour trees - the art can consists of placeholders but the scene should showcase a somewhat complex behaviour, such as enemy patrolling in different modes, or AI agent going through a door which may require a key. If you do implement your own behaviour trees, the point is not just to get it working, but have it done in a way that would be usable by other member of your team.

  or

- Devise and complete your own AI Master Challenge - just check with the lecturer or the demonstrator first, whether the scope of the work will be sufficient for the awarded skill points.

Assessment

Show your work to the demonstrator for assessment.