Creating Generic Wars

With Special Thanks to Tommy Gun and CrackedRabbitGaming
Kodu Curriculum: Getting Started

Today you will learn how to create an entire game from scratch with Kodu! This tutorial will teach you how to make a Geometry Wars style twin-stick shooter game. By the end, you’ll have a game that will look like this…

---

**Part 1:**

In this part we’ll create an empty world, learn about land tools, and setting up basic character controls.

Before we begin, learn the parts of the Xbox Controller because we’ll refer to them throughout.

![Figure 1: Parts of the Xbox Controller](image)

Note: While all the buttons needed for each step are specified in the tutorial, remember there is guide on the top left corner of KODU telling you what each button does.
1. Once in Kodu, you should choose “Load World”. Hit A to select.

2. After choosing “Load World”, select “Empty World” to start a New World using the A button. When you load a world, it starts in the play mode. Hit the back button to edit the ‘Empty World’. You will then see a strip of editing tools at the bottom.

3. Terrain is created by painting with the terrain tools. We’ll begin by removing the default terrain.

   Scroll to the ground brush tool and hit A. Highlight the default terrain and hit the left trigger button to remove.

   Now we’re going to add black, circular terrain.

   To select a brush type hit X and scroll to the oval brush. Hit A to select. Then choose a new material by hitting Y. Scroll to the black material as this is the one we’ll be using.

   Place the terrain. An easy way to make terrain is to increase the brush size (using the right arrow on the D-Pad) and just put down a single oval by hitting the right trigger button as seen in the picture to the right.

   - Aside: The other tools for dealing with terrain let you raise and lower the terrain, and smooth or make jagged the elevations.

   - Any water that is added, must have terrain or land under it. By default, each world starts with “glass walls”, so water stays contained and nothing can fall off the edges.
4. We will now add our program objects and characters. Scroll to Object Tools (Kodu icon) on the ToolBar and hit A. This lets you add a new character or object by clicking in a space without an object. So hit A again and then find the Puck as this is the object we'll use. This Puck is going to be the shooter.

The Puck is great for fast games as it flies around without friction but it does bounce of walls and continue to move when you let go of the controls.

5. We’re now going to program the Puck. To edit an object, you must move the cursor to the space where the object is and hover. The object will glow, and then you can hit ☰ to see the code and program. You can change the color of the object using the left and right keys on the D-Pad. We’ll leave our Puck white.

6. After hitting ☰, you should see the screen to your right. Press ☰ to add code in the selected tiles.
We’re going to make our Puck move super fast. Use the following code for tiles: **When** gamepad, L-stick **Do** move quickly, quickly, quickly. Your code should appear as it does on the right.

When you’re done, hit B to close.

Aside: All Kodu code statements start with a When condition, followed by a Do condition to be executed. The choice of tiles at any point is determined by what went before. Conditions are evaluated simultaneously. If there are no tiles in the area, then the action will be done at all times.

7. Now hit the back button to test your code out. You should be able to move your puck using the Left stick.

Once you’re done, hit the back button again to continue programming and editing.

8. Move the cursor to the Puck and hit 🎉

We’re going to add the following code: **When** gamepad, R-stick **Do** shoot, blip

Close the screen by hitting B and then run the game using the back button. You should be able to shoot with the Right stick. When you’re done, hit the back button to continue editing.
9. We’re now going to edit the settings of our object. Move the cursor to our object, the Puck, and hit the X button to open settings when the object glows. You’ll see the screen to your right.

Scroll to ‘Show Hit Points’ and change it to on by hitting so we can see how much health we have.

Then scroll down to ‘Blip Speed’ and increase the speed to 100 using the right button on the D-Pad.

And then change ‘Blips at Once’ to 200 [This way we can shoot more blips faster]

You can also scroll down and increase ‘Glow Light Strength’ to 5.[Hit the back button to return to editing]

10. We can now edit our object once again. Move the cursor to the Puck and hit 🌟. We’re going to make it glow this time.

Enter this code: Do Glow, White. There is no when condition this time as we want it to glow always.

Glow is under the Actions menu in case you can’t find it.
Close the programming screen and test your game by hitting the back button. You should be shooting faster and you should also be able to see the health bar. Yom may need to zoom in using the right bumper.

Once you’re done testing, go back to editing by hitting the back button.

11. We’re now going to set our camera angle. Scroll through the Tools strip to Settings and hit A.

Once in the Settings, scroll to Camera Mode, use the D-Pad to change the selection to ‘Fixed Offset’ and then hit X to save the setting. This sets the angle but allows the camera to move with our object.

Hit the back button to run the game to test it out. Once you’re done, return to edit mode.

---

**Part 2**

In this part we'll add walls, some enemies, AI, a path for a wisp. We will also learn how to use pages.
12. To create walls for our world, we’ll use another land mass to create a perimeter. The perimeter will serve as a guide for our wall placement.

So to add land of a different color/texture around the perimeter of the landmass, use same steps as before, only change the landscape material and the brush size so the new land mass is bigger than the existing land mass.

You should end up with a landscape that resembles the screen to the right.

13. Now scroll to the Objects tool, find the Wall button and hit 🗜️ to build walls. Move the cursor to a point on the perimeter and hit 🗜️ again. Find the ‘Wall’ item in the objects menu. It will be under the button pictured to the right.

Change the color of the wall to orange using the D-Pad.
14. Place the nodes of the wall using the button, following the perimeter to build an octagon. You should end up with a wall as pictured to the right. It doesn’t have to be perfect.

15. The wall is low so place the cursor on the wall and hit X to select the whole path. Hit to pick it up. Then use the up button of the D-Pad to raise the entire wall up. When you’re done, hit again to put the wall down.

Raising the wall is important to keep the enemies we will create from flying outside the arena. You should end up with a wall as seen to the right.
16. Now that we have our wall, you can go back and change the land color to black using the Land Brush tool as before.

17. Now we’re going to create and program our enemies. Use the same process as before to create an object. See steps 4-6 if you’ve forgotten! We will use another Puck object to program as an enemy.

Make sure to change the color of the enemy Puck to red using the D-Pad or else it will look like our shooter Puck.

18. We want to score 20 points when we hit the enemy.

Use the following code to do this: **When** health, 00 points **Do** score, 20 points, yellow.
Hit the back button and test it out. You should see a yellow score on the top right. When you’re done go back to edit mode.

19. You may have noticed the enemy did not die very quickly. We want this to be a fast game so we have to change that.

Move the cursor to the enemy Puck and hit X to go into its Settings.

Scroll to Max Hit Points and change it to 5 using the D-Pad. This will match the damage done by our blip hits so it will only take one shot to kill the enemy Puck.

20. Since we are in settings, we want to change this enemy Puck to ‘Creatable’. This would make it so that we can create as many enemy Pucks as we want, but will only have to change the settings on this one Puck as all the other enemy Pucks will mirror those settings.

Close the settings and test the game.
21. You will notice that you can’t see the enemy anymore. This is because we made that one Creatable. It is the first one and holds the settings. To keep our game organized, move your cursor to the enemy Puck, hit ⚫ to pick it up, move it outside the arena walls and put it down. This way we can always find the enemy Puck when we need to change settings.

22. Now we want our enemy Pucks to move. To do this, we need to first create a path for their movement. The process is similar to creating the wall except select the ‘Plain’ button instead of the ‘Wall button’ and change the color of the path to red. See steps 13-14 if you’ve forgotten!

Here’s a simple path you can create to start
23. Now we need an object that will create our enemies automatically during the game.

We can use an object called a Wisp and program it to do so.

Use the same process as previous to create and program a Wisp object. See steps 4-6 if you’ve forgotten!

24. We want our Wisp to create enemy Pucks randomly at least every 1 second and have them move along our path. Write the following code for the Wisp object

1) **Do**: move, on path, red

2) **When**: timer, 1 second, random **Do**
   create, Puck 2

Note: Create is under the Actions button and our enemy, Puck 2, is under the Creatables button
Hit B to close the screen and run the game using the back button.

You should see the Wisp and the enemy Pucks should be moving. Once you’re done, hit back to go to edit mode.

25. Now we don’t want the Wisp to be seen because then we know where the enemy Pucks are going to show up, giving us an advantage. So move the cursor to the Wisp object and hit X to go into its Settings. Change the Wisp to invisible. If you run the game, you will see that enemies are created but you can’t see the Wisp.

26. We want to make our game more difficult so we’re going to give our enemies intelligence. They are going to come towards us to attack but not always.

Move the cursor to the creatable enemy Puck and hit 📐 to program. Put in the following code:

2) **Do** move, on path, red

3) **When** timer, 5 seconds **Do** switch, page 2

**Note:** The “switch to page x” construct is used for code to be run only after a specific event or condition has been achieved – it is used to create multiple states for a character or object.
27. Hit the right bumper to move to page 2

Write the code

1) **When** hear, puck, white, **Do** move, toward

2) **When** timer, 4 seconds, **Do** switch, page 1

Also add the code: **When** health, 00 points **Do** score, 20 points, yellow

We do this in case the enemy is hit while running this code

28. Now we want to ensure each of our objects is at the same height otherwise there may be problems when shooting. The blips could miss if they are being shot too low or high.

Change the height of each object (Puck 1, Puck 2 and the Wisp)

To do this, move your cursor to the object, press \[A\] to pick it up, use the up/down D-Pad keys to change height and then press \[A\] to put it back down. For ease, just lower them to the ground.

This may not be a problem because all our objects are pucks but this could be a problem
if you use other Objects when modifying the game

29. You may have noticed the yellow score is on loud, it shows +20 points, when you hit the enemy Pucks. We don't need that so we'll change it

Scroll to settings on the Tools strip

Scroll to ‘Score Visibility: Yellow’ and hit A to change it to Quiet so it will still display the score

Part 3: In this part we'll program a 60 second clock on an invisible rock, add music, and bump some enemies.
30. Now we need a time limit for our game. To do this we need to create an object that will hold that setting.

Create and program any object using the same process as before. We’ll use a rock. For organization, pick it up and place it outside the arena.

Move the cursor to the rock and hit 🨑 to program.

31. Write the following code:

1) **Do** score, 50 points, 10 points, white, once

2) **Do** switch, page 2

This sets the timer to 60 seconds at the beginning.
Now hit the right bumper to switch to page 2 and write

1) **When** timer, 1 second **Do** subtract, 01 point, white

2) **When** scored, 00 points, white **Do** switch, page 3

**This makes the timer count down by one second and when it gets to 0, will end the game**

Now hit the right bumper to switch to page 3 and write

**Do Win**

This ends the game

---

32. Close the programming screen and hit the back button and play the game

You should see the timer on the top right

33. You’ll notice that the white score is on loud as was the yellow score. Follow the same process as in Step 29, to change the score to Quiet
34. Lets program our shooter Puck such that it is hurt when we come into contact with enemy Pucks. Move the cursor to the shooter Puck and hit 🍃 to program. Write this code

4) **When** bump, puck, red **Do** switch, page 2

**Hit the right bumper to move to page 2 and then write**

1) **When** bump, anything **Do** damage, 20 points, me [We can use anything because the only thing we can bump into are the Pucks]

2) **When** bump, anything **Do** boom, it [This will blow up the enemy Pucks if we bump into them]

3) **When** health, 00 points, **Do** end [This means game over cause we died]

4) **Do** switch, page 1

**Note:** damage and boom are under the combat button

**Hit the back button to test**

35. Lets increase our health so we can last longer

Move the cursor to the shooting Puck and hit X to go into Settings. Change ‘Max Hit Points’ to 100 using the D-Pad. Now we can be hit 5 times before we die
36. You can also add music to your game. You can skip this today because there are no speakers in the lab. You need to set this on an object just like the timer. We can use the rock object again. Open the rock’s programming page, switch to page 2 and add the following code

3) **Do** play, action

   *Note:* Play is under the Actions button

37. Let’s make the game more interesting by increasing the speed at which new enemies are created as time goes.

   Move the cursor to the Wisp and hit 🆙 to reprogram

   Write this code:

   3) **When** timer, 20 seconds, **Do** switch, page 2

   Hit the right bumper to switch to page 2 for the Wisp object

   Write this code:

   1) **Do:** move, on path, red

   2) **When:** timer, 0.25 seconds, random **Do** create, Puck 2

   *Note:* Create is under the Actions button and our enemy, Puck 2, is under the Creatables button
38. To create even more enemies simultaneously, we can clone the Wisps. Move the cursor to the Wisp object, hit 
A to pick it up. Then press the right trigger to clone and place the Wisp in any location. Do this twice. Then hit 
A to put the last Wisp down. You should have 3 Wisps in total.

Run the game using the back button to test.

Part 4: In this final part we'll add some lights, and put the finishing touches on our game. We learn how to add a control screen and add glyphs,

39. Change the sky and lighting to appear as nighttime. Select the Settings icon in the main tool strip. Scroll through to find ‘Sky’ and change the setting to black, and then scroll down to ‘Lighting’ and change the setting to Night using the left/right D-Pad keys.
40. We can also make the enemies glow. Move the cursor to the enemy Puck and program. Add the code

4) Do glow, red

41. We can add lights to create a nice effect.

Scroll to the Objects tool. Move the cursor to a location on top of the wall, hit A and add the light. Change the colour of the light using the D-Pad. White is fine.

To add lights all along the wall, move the cursor to the placed light, pick it up by hitting A and place clones using the right trigger. You can add a maximum of 8 lights. Hit A to place the light that was used to close.

The orbs of the lights will be visible. You can make them invisible by moving the cursor to the object and hitting X to change the setting if you have time.

Hit the back button to test.
42. Save and name the game!

The game is fairly complete! You can always play and tweak it as you like.

If you have time, you can create a control screen for instructions. Just add the following code to the rock’s programming and type in the instructions:

Do say, once

The instruction screen will look like that to the right

To get the glyphs to place symbols in the instructions such as for the control buttons. See


As an example, the left stick would be <LS>
The game is fairly complete but you can always tweak it! As examples, you can add new enemies, change the shooting weapons, add a health score, give your shooter the ability to heal, change the movement path so it is more sophisticated as seen on the right...