Evil Clutches: Stages 7, 8, and 9

This tutorial is a continuation of the EVIL CLUTCHES game build. Do not continue if you have not completed STAGES 1 through 6. In STAGES 7, 8, and 9, you will be introduced to **conditional statements**. Computer programs are written logically. In a **logic system**, the outcomes of events are based on the conditions that exist. In writing we can use words like *IF* and *THEN* to express conditional statements. An example of this could be written this way: IF you destroy a demon, THEN points are added to your score. All of your programming properties are based on logic, although they can be much more sophisticated than this example. If you think about most games, the things that happen are usually a result of other things that happen. Any event or action can be the cause of other events or actions. This is what we mean when we say **cause and effect**. A good programmer can use logic to design a game so that rewards and penalties are caused by specific events and actions. Another example might be: *IF* I destroy a dragon baby, *THEN* I will be penalized and lose points. **Conditional actions** can be created to cause other events and actions. Some can happen randomly based on a preset number of chances written into the programming, or what in math is referred to as **probability**. This tutorial includes **conditional statements** and **actions**, **cause and effect**, as well as **probability** in the programming properties. In later tutorials, you will deepen your understanding of **logic systems**. Add the *new vocabulary* definitions to your Tutorial Guide.

BE SURE TO PAUSE TO COMPLETE HYPOTHESIS AND EVALUATIONS WHEN PROMPTED.

( ) DO NOT USE THE TEST BUTTON IN THIS ACTIVITY UNTIL THE END!

START HERE: Make sure you are in advanced mode and that you open your file ***initials_clutches***.

STAGE 7:

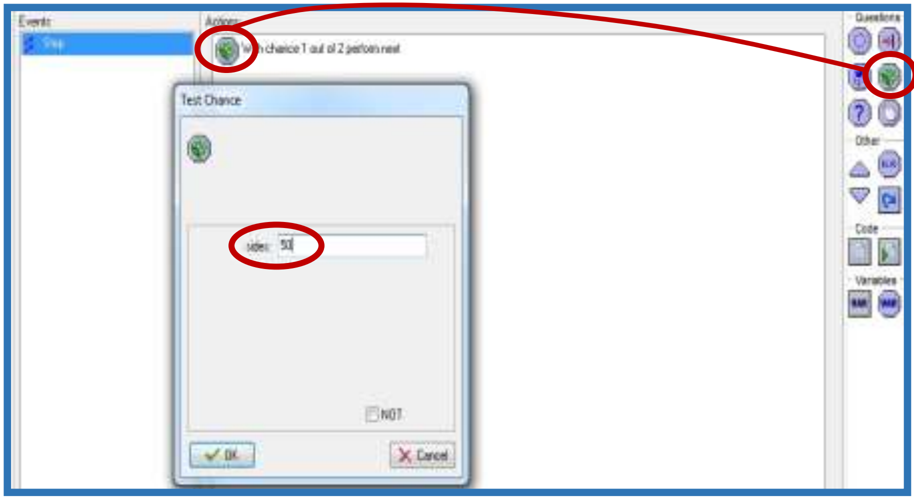
Start your logic system by creating a conditional action with probability

1- Open up the ***obj_boss*** properties. Add a **Step Event**. From the **Control tab** on the right, drag and drop the **Test Chance** action (it looks like a die or dice cube). Make the number of sides on this die **50**.

2- From the **Main1** tab add a **Create an instance of an object** and choose ***obj_demon*.** Use x = **0**, y = **0** and be sure to check the **Relative**. You should recall a similar property for the fireball to shoot it from the dragon.

3- Click **OK.**

NOTE: In your HYPOTHESIS for STAGE 7, be ready to explain what this part of the **logic system** does. At the end of this tutorial, you will also write a **conditional statement** for this stage. Review the Glossary definitions of **conditional statements, conditional action** and **probability**. Read more about these in Concepts Explained to deepen your understanding of the programming before writing your hypothesis.

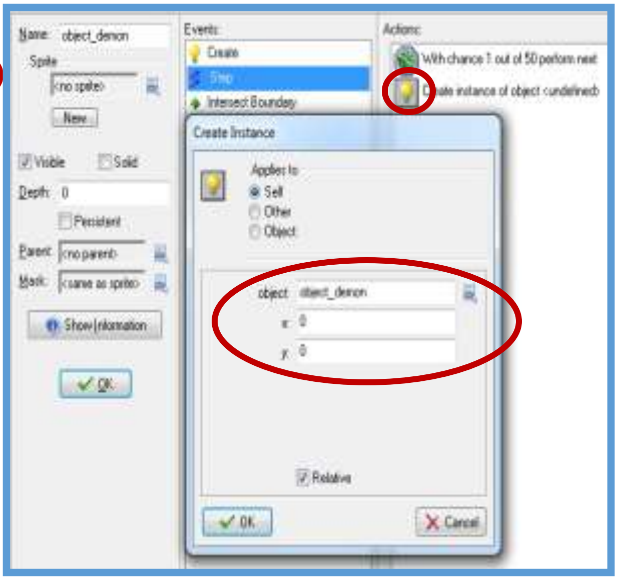

With chance 1 out of 2 perform next

Test Chance

sides: 50

□ NOT

SAVE YOUR GAME FILE. TRY CLICKING THE DISK ICON


Name: object_demon

Sprite: <no sprite>

☑ Visible □ Solid

Depth: 0

□ Persistent

Parent: <no parent>

Mask: <same as sprite>

With chance 1 out of 50 perform next

Create instance of object <undefined>

Create Instance

Applies to: ● Self ○ Other ○ Object

object: object_demon

x: 0

y: 0

☑ Relative

DO THIS ON YOUR TUTORIAL GUIDE

STAGE 7 HYPOTHESIS STATEMENTS: Now it is time to predict the behaviors of the demon object properties applied to the boss object in your logic system.

- **Compare the "With a chance of 1 out of 50" to the rolling of a die 30 times in a second. What effect will this have on the showing of the demon instances?**

Go ahead and fire at a few for fun (▶)! You should now see demons being deployed, or "*spawned*" by the boss. The fire balls should destroy the demons and earn you points. You should see a score in the caption bar (top left) of the game window. If a demon collides with you, the high score table will be show. Test the rest of the high score table by closing it to restart the game. You can write about the functionality of the game programming when you complete your testing and evaluation at the end of this tutorial. If your game is not working as described above, go back through the properties and make corrections before proceeding with STAGE 8.

STAGE 8:

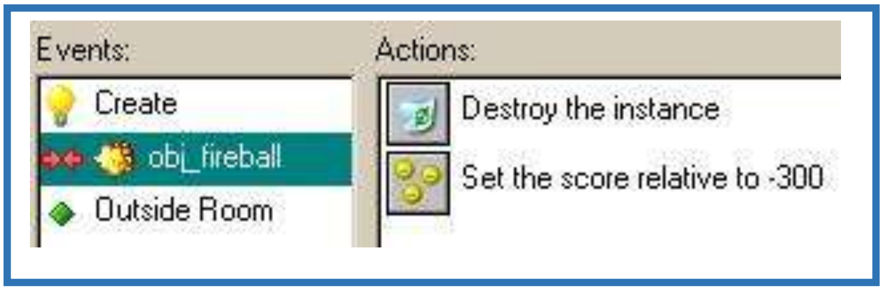
My baby is faster than your baby

1- Create ***obj_baby*** with the ***spr_baby***. Add a **Create Event** and a **Move Fixed** action. Select the **left arrow** for this **Start moving in a direction** action and set the **Speed** to **8**.
NOTE: In your HYPOTHESIS STATEMENTS for STAGES 8 AND 9, be ready to compare speed of demons and babies and why that may be important in the playability of the game, referred to as *balance of play*.

2- Add an **Other: Outside Room** Event and set the Action to **Destroy the instance** of **self**.

3- Add a **Collision Event** with ***obj_fireball*** and set the Action to **Destroy the instance** of **self**.
NOTE: At the end of this tutorial, you will write a **conditional statement** for STAGE 8 Step 3.

4- Add an Action in the **Score tab** where you must **Set the score** to **-300** and check **Relative**.

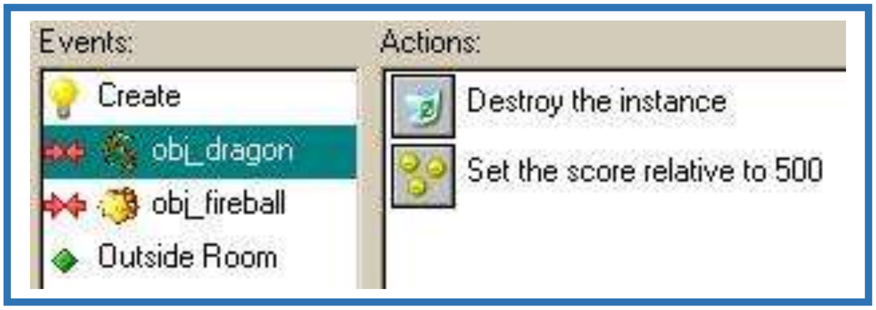


Events:
- Create
- obj_fireball
- Outside Room

Actions:
- Destroy the instance
- Set the score relative to -300

5- Add another **Collision Event** with ***obj_dragon***. Drag an Action for **Destroy the instance** of **self**.

6- From the **Score tab,** drag and drop **Set the score** to **500** and



Events:
- Create
- obj_dragon
- obj_fireball
- Outside Room

Actions:
- Destroy the instance
- Set the score relative to 500

SAVE YOUR GAME FILE

BALANCE OF PLAY

It is important to keep a game challenging, but the player should also have a fair chance of success. *Balance of play* is maintained when **a logic system** both helps a player and works against him at the same time. This concept is sometimes called *balance of power*. Such a system would require **conditional actions** and **probability**. Look for this in Concepts Explained.

DO THIS ON YOUR TUTORIAL GUIDE

STAGE 8 HYPOTHESIS STATEMENTS: Now it is time to predict the behaviors of the baby, demon and fireball objects based on newly added properties.

- **Explain the collision events for each object and explain the action you expect to see. Why is the speed of the dragon babies different from the demons and how do you think it will the balance of play?**

STAGE 9:

So whose baby is faster? Who has more babies?

More probability from conditional action to complete the logic system

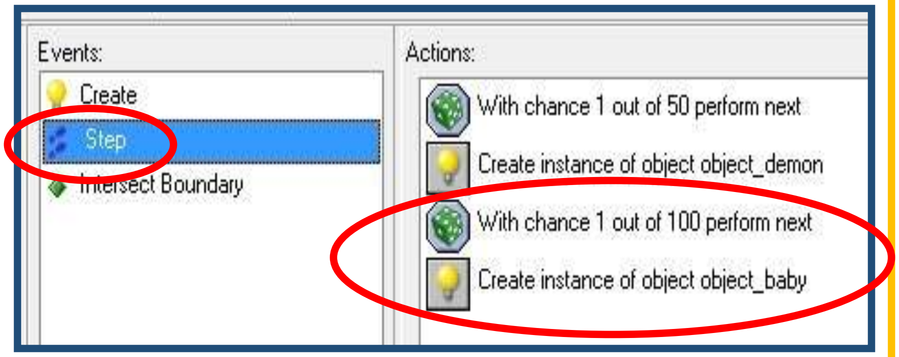
1- Open up the properties window for ***obj_boss*** and drag a **Step Event**.

2- For **With a chance to perform next action** type in **100** for sides. This is the number of sides on the die.

3- From the **Main1 tab**, drag and drop **Create an instance of an object** and select ***obj_baby*** from the drop down. Set x = **0** and y = **0** then check the **Relative box.**

NOTE: If the Boss could speak, you might hear him make the following **conditional statement**: "*IF* I get my 1 in 100 chance *THEN* I will *spawn* one of my babies." If the Dragon and the Boss could both speak, who would have bragging rights to claim the fastest babies? Who produces more babies? Think about this so you can write about it in your HYPOTHESIS for STAGE 9**.**

`


Events:

Create

Step

Intersect Boundary

Actions:

With chance 1 out of 50 perform next

Create instance of object object_demon

With chance 1 out of 100 perform next

Create instance of object object_baby

SPAWNING

Sometimes, a game can be made more challenging when a character or object is produced or "made alive", often without player control. The gaming term to describe this is *spawning*. There are several ways to program for a *spawn* in Gamemaker. Look for this in Concepts Explained and as a *new vocabulary* word in future tutorials.

DO THIS ON YOUR TUTORIAL GUIDE

STAGE 9 HYPOTHESIS STATEMENTS: Now it is time to predict the behaviors of the demon and dragon babies in the logic system you created for *balance of play*.

- **Explain what you expect to see as a result of probability and conditional action properties added to game.**
- **Who has the fastest babies (Dragon or Boss)? Why?**
- **Who *spawns* more in a period of time? Why**

SAVE YOUR GAME FILE

DO THIS ON YOUR TUTORIAL GUIDE

CONDITIONAL STATEMENTS: Remember, events and actions that occur in games often cause other events and actions. This is **cause and effect** and can be written as *IF/THEN* conditional statements.

- **Write a conditional statement (*IF/THEN*) for STAGE 8 Step 3.**
- **Write a conditional statement for STAGE 7.**

Now it’s time to test your game, so go ahead and click on the green triangle (▶) on the menu bar to run the game normally.

DOES THE ACTION THAT YOU SEE AND THE CONTROL OF THE OBJECTS MEET YOUR HYPOTHESIS STATEMENTS?

The Demon Boss is now sending out those pesky demons and the fireballs from STAGE 6 now have real purpose. Mama dragon will do what it takes to protect her babies from those little minions by destroying them with fire. Hopefully, she doesn’t slip up in her eagerness to be their protector by hitting her own babies. You should see the point scores from the properties working in terms of awards and penalties for the player controlling the Mama! If a demon gets the Dragon mother or if a fireball hits a baby dragon, the game ends and you should see the high score table.

What about those babies? When you created the demons in STAGE 5 you put in a fixed move action that moved them to the left at a speed of 12. The dragon babies are set for a speed of 8, so unfortunately the Demon Boss has faster babies. However, the Dragon mother produces, or “spawns,” babies much faster than the Boss. Gamemaker programs will roll the die 30 times per second. The property that produces the dragon baby instances is set so the chances of producing a baby are 1 in 50 for each roll at 30 rolls per second. For the demons, the chances are 1 in 100 at 30 rolls per second. **Probability** tells us that chances are that we will see more dragon babies than demons. Since it is 1 in 50 versus 1 in 100, the chances of seeing a dragon baby are double. This is a good way to create *balance of play* for the babies to compensate for the faster demons.

Random number generators

The “chance to perform” property creates a *random number generator* in your game. This term is used for devices that can create a number from a given range of numbers (or number of chances). Several common *random number generators* are shown, along with the range of chances for each. This will be explored again in future tutorials and can be referenced in Concepts Explained.

1 in 100

1 in 20

1 in 2

1 in 6

1 in 6
Assign a color for each number

DO THIS ON YOUR TUTORIAL GUIDE

STAGE 7, 8, 9 TEST & EVALUATION: Compare your game to the hypothesis statements by testing their validity.

- **Does the action of your game play out the way you anticipated? Did you have to fix problems to make it work? If so, what were they and how did you fix them?**