


## Galactic Mail: Stages 4, 5, and 6

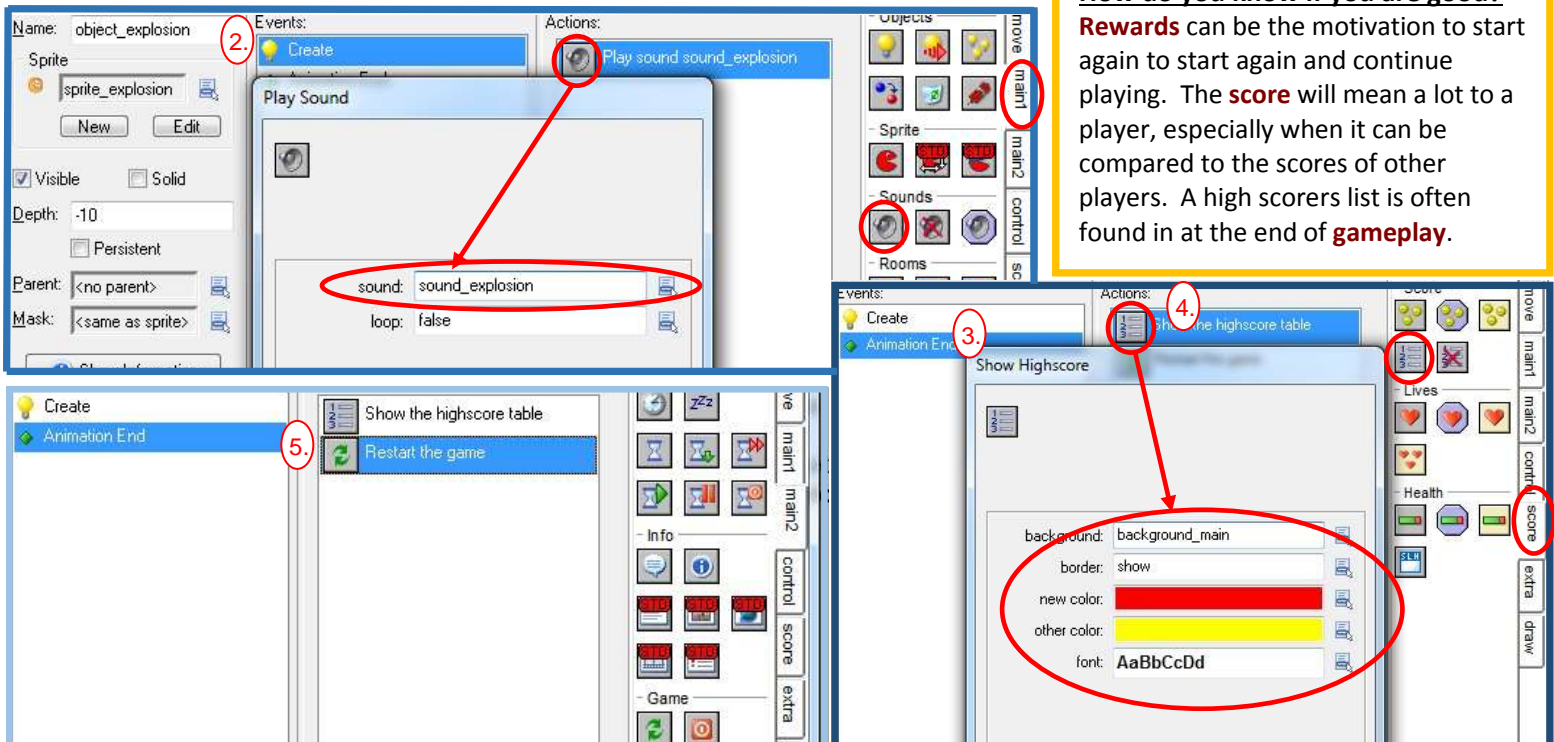
Continue and complete the *Galactic Mail* game by finishing Stages 4, 5, and 6. There will be more **HYPOTHESIS** statements as well as **EVALUATIONS** to complete on your tutorial guide. Be sure to write complete sentences for each before moving on through additional stages. In this tutorial, you will learn how to introduce your game with a basic **front end** designed in a room. You will deepen your understanding of programming with properties for **rewards** and **penalties**. You will be building properties to add points to **score** for good performance, or take away points if a game rule is broken. Another way to reward a player is to allow them to advance into a higher, more challenging **level**. The properties you build using **blocks** will use a set number of points to cause a change in the **gameplay** and **playability** of the game, making it more challenging, and of course, increasing the rewards and penalties. The player should be congratulated at the end of the game. What about rewards for the game designers, programmers, writers, artists, and others who invest time and talent to game builds? They deserve recognition in the **credits**, listed in a room at the back of a game in a **game information** screen. This is important to remind people that the *profits*, or *monetary rewards*, for a game belong to those responsible for its creation.

- **BE SURE TO COMPLETE HYPOTHESIS STATEMENTS FOR EACH STAGE.**
-  **DO NOT USE THE TEST BUTTON IN THIS ACTIVITY UNTIL THE END!**
- **SET UP GAMEMKER IN ADVANCED MODE**
- **OPEN YOUR FILE YOU SAVED AS *initials\_galactic***

### STAGE 4:

#### Adding an explosion object to the game

- 1- Create a new object called **object\_explosion** using **sprite\_explosion**. Make the **Depth -10**.
- 2- Add a **Create** event, drag and drop a **Play Sound** from the **main1 tab** for an action.
- 3- Add an **Other, Animation end** event. Adding this to the object allows action to happen after the last sub-image has shown in the animation.
- 4- Drag a **Show Highscore** from the **score tab**. To make it look more interesting, set **Background** to **background\_main**, set **Other Color** to Yellow then select a font from the menu.
- 5- Add a **Restart Game** from the **main 2** tab. Click **OK** to close.



The screenshot shows the game engine interface with several windows and panels. Red circles and arrows highlight specific actions:

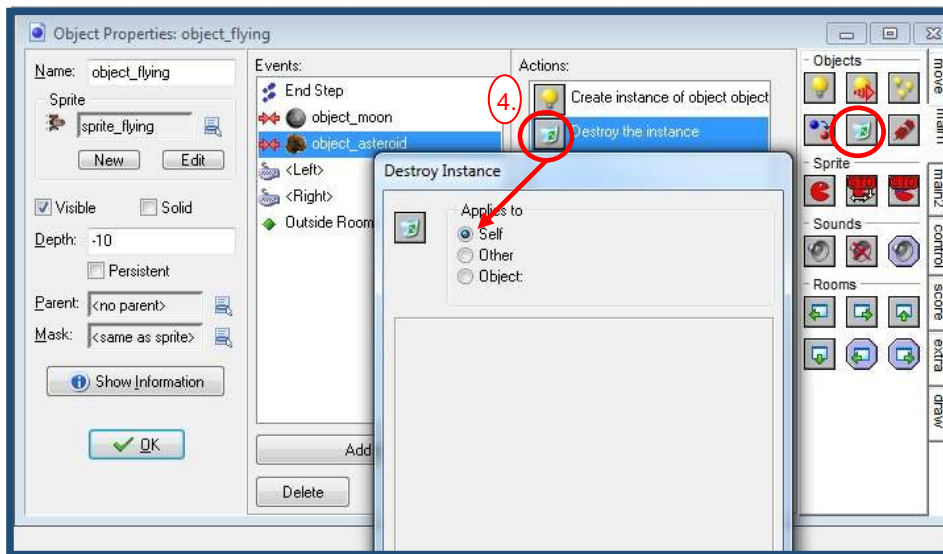
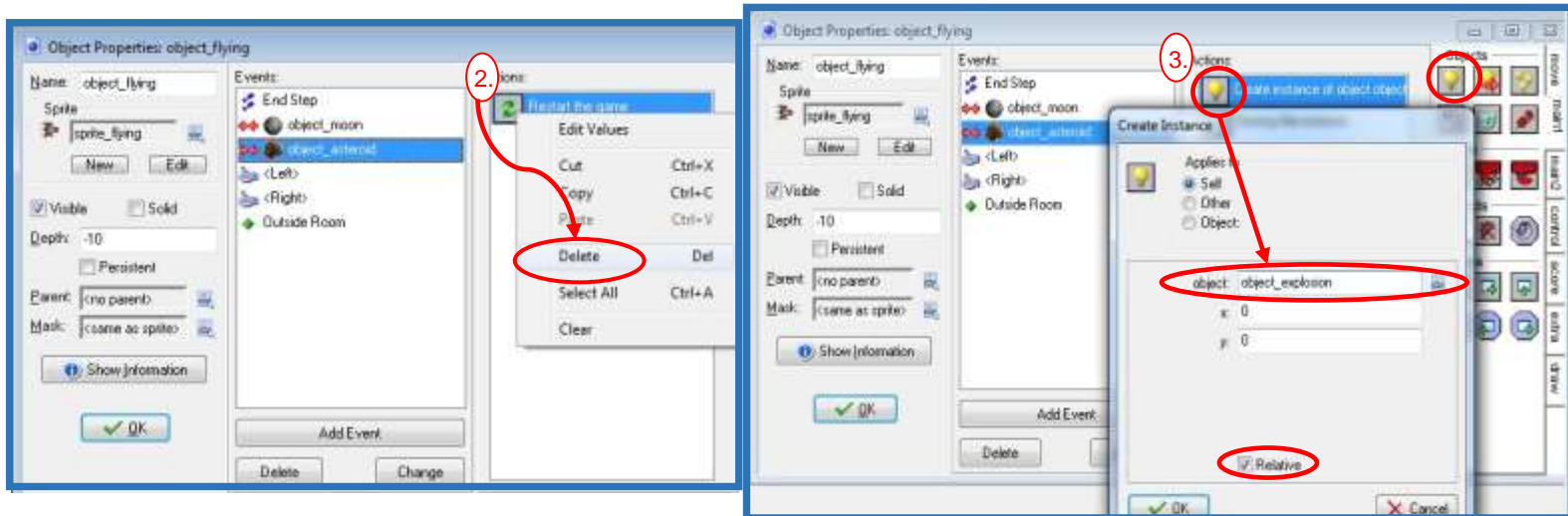
- 2.** A red circle highlights the 'Create' event in the Events list, and another red circle highlights the 'Play sound sound\_explosion' action in the Actions list.
- 3.** A red circle highlights the 'Animation End' event in the Events list.
- 4.** A red circle highlights the 'Show Highscore' action in the Actions list, and another red circle highlights the 'background\_main' setting in the Show Highscore dialog box.
- 5.** A red circle highlights the 'Restart the game' action in the Actions list.

A yellow box on the right contains the text: **How do you know if you are good? Rewards** can be the motivation to start again to start again and continue playing. The **score** will mean a lot to a player, especially when it can be compared to the scores of other players. A high scorers list is often found in at the end of **gameplay**.

**SAVE**

## Editing the Flying rocket object

- 1- Reopen the *object\_flying*.
- 2- Select the **Collision object\_asteroid** event. Select **Restart Game** (just so it is highlighted in blue) then right click and select **Delete** to remove it. (or you can just highlight **Restart Game** press <Delete> on your keyboard).
- 3- Add a **Create Instance** action from **main1** and set so it creates an instance of *object\_explosion*. Enable **Relative** is by checking the box.
- 4- Add a **Destroy Instance** from **Main 1** and leave it checked on **Self** so rocket is destroyed. Click **OK**.



### Motivation

Many games end with the **penalty** of an abrupt stop in the gameplay. Players are then motivated to start over. When an *animation* like an explosion is used, the impact of the penalty is more dramatic in the game and for the player. The player will continue to learn from mistakes and improve. Although **penalties** are the opposite of **rewards**, they can be influence a player to start over again and again until they get it right. They want the rewards of higher **scores** and the chance to advance **levels**.

**HAVE YOU SAVED LATELY? PLEASE DO SO!**

### DO THIS ON YOUR TUTORIAL GUIDE

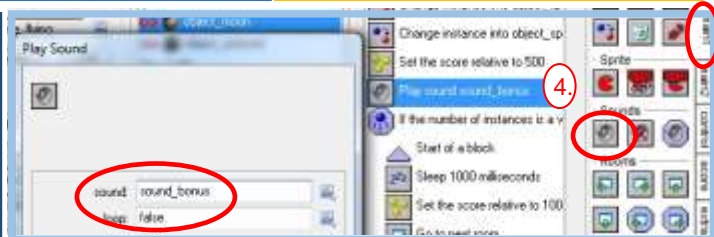
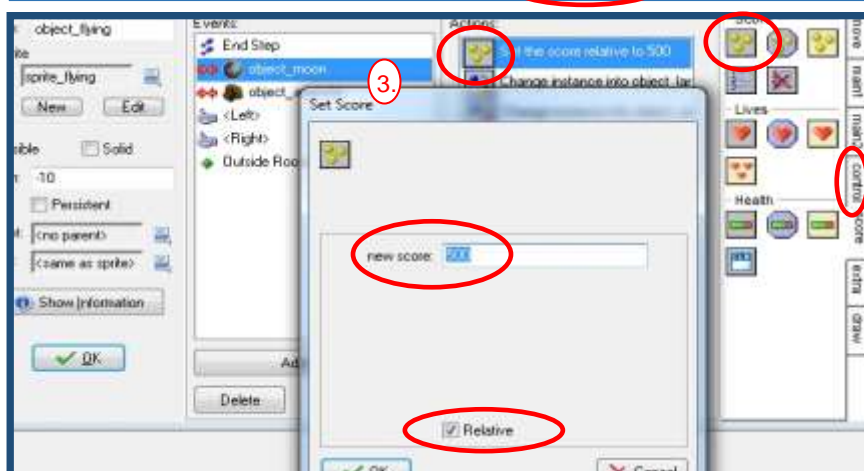
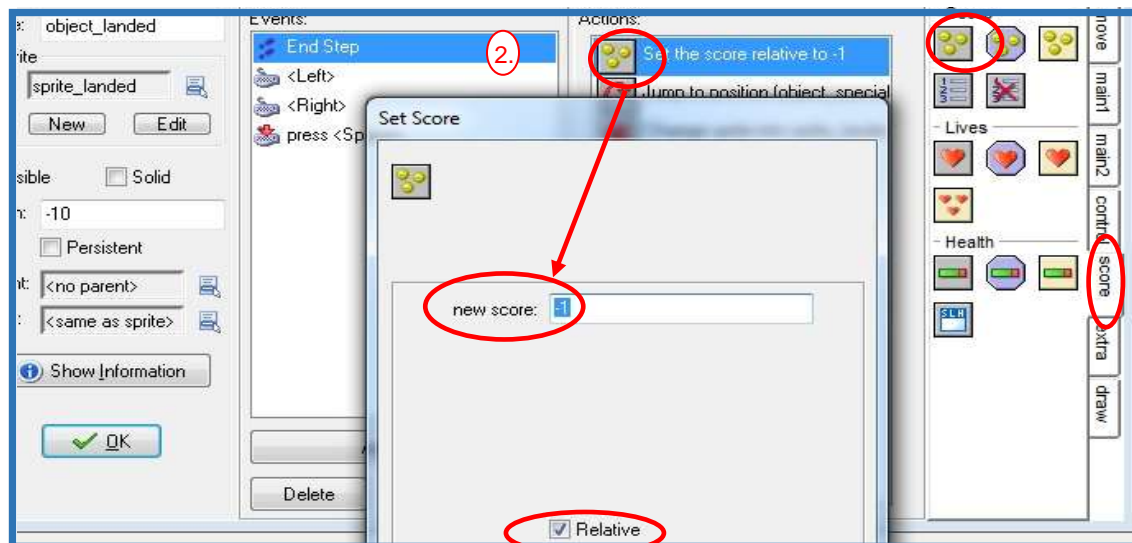
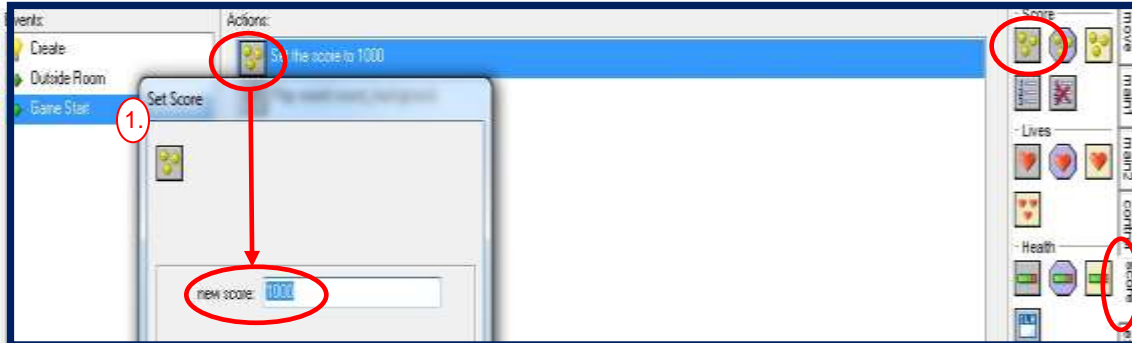
**STAGE 4 HYPOTHESIS STATEMENTS:** Now it's time to predict the behaviors of the rocket and the asteroid.

- Explain why we deleted "Restart Game" from the asteroid and how it changes the end of the game.

## STAGE 5:

### Adding scoring properties to objects

- 1- Reopen the **object\_specialmoon** properties and select the **Game Start** event. Add a **Set Score** with a **New Score** of **1000**. Click **OK** to close the properties.
  - 2- Reopen the **object\_landed** properties. Select the **End Step** event. Add a **Set Score** and for **New Score** enter **-1** with **Relative** enabled (checked). Click **OK**.
- NOTE:** Assuming that negative points are a penalty, what is the player being penalized for in the **game play**? Be sure to write about this in your Stage 5 Hypothesis.
- 3- Reopen the **object\_flying** properties and select the **Collision** event with **object\_moon**. Add a **Set Score** and **New Score** of **500** with **Relative** enabled.
  - 4- Add a **Play Sound** action and select **sound\_bonus**. Loop should be **false**.



### Playability

A game should be designed with **playability** in mind. When a player engages in play, they first need to know the story, as well as the goals, and know they are obtainable. As they develop the skills to succeed in the **gameplay**, they expect an increase in challenges along with the **rewards** and **penalties**. Game designers need to consider the **level** of challenge in the game, and the **ability** of the player to meet the challenge, as well as the **balance of play**, as introduced in *Evil Clutches*.

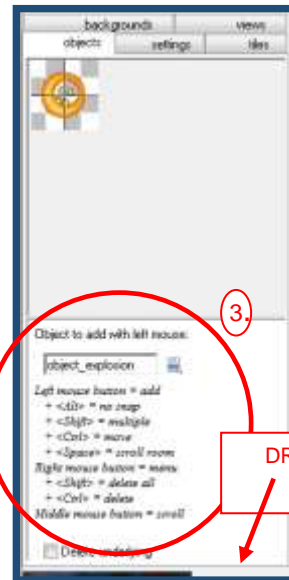
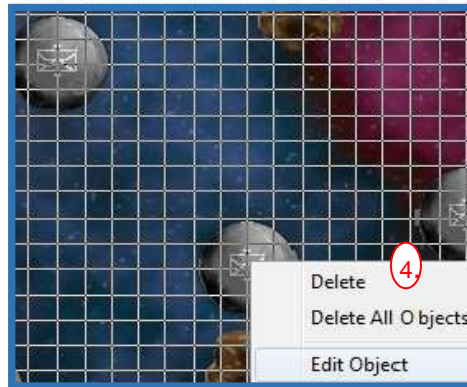
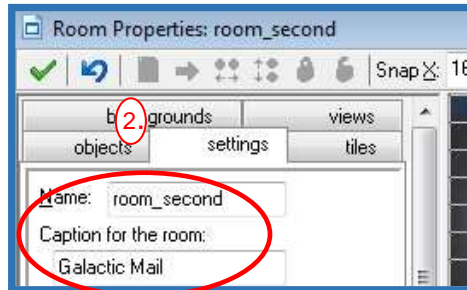
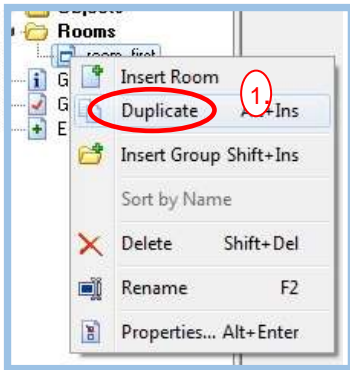
**Gameplay:** Similar in meaning to playability, **gameplay** is what the player focuses on. It is simply the experience of the player. When you play a game, you don't think about what the developers had to do to create it. You are just concerned with what you need to do to succeed, or **gameplay**.

SAVE

## Creating more levels for the game

- 1- Right click **room\_first** in the resource list and then select **Duplicate**.
- 2- In the Room Properties window, click on the **settings** tab. Name this room **room\_second**.
- 3- Go back to the top of Room Properties and select the **objects**. Here you can add or delete instances. Expand the menu "Objects to add with left mouse". If you forgot how to add or delete instances, you can refer to the italicized list at the bottom of the objects tab window. Add few moons and asteroids.
- 4- **Make sure each level contains only 1 special moon and one landed rocket.** Do this by moving the mouse pointer around the room. As you hover over each object, you should see the object name appear at the bottom of the room to the right of the x and y coordinates. If you find more than one special moon, delete it by hovering over it and right clicking.
- 5- Repeat the three steps above to make a third room (for a level) named **room\_third**.

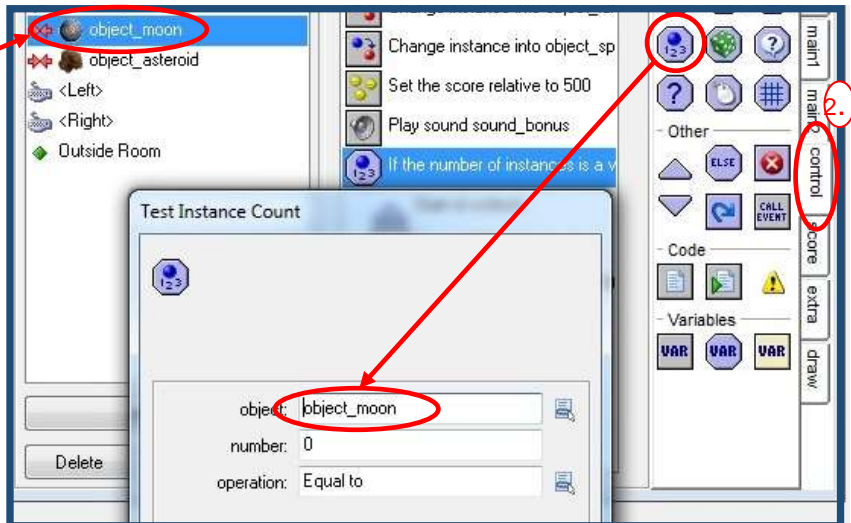
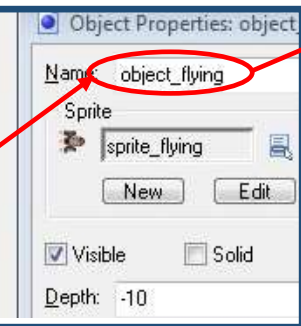
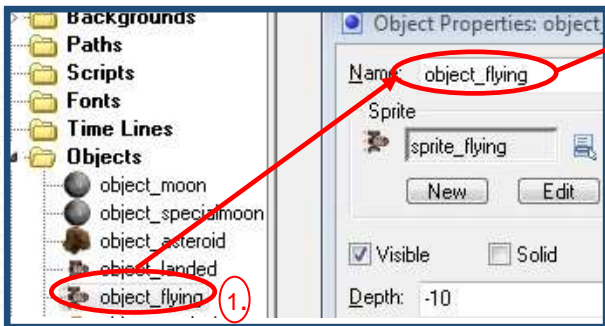
**NOTE:** The order of levels played are based on the order they are named. You can change that order by dragging the one you want to be next above the one to be occurring last, or the one occurring first to the top.



**Next Level!**  
A **level** is a change in the game properties, appearance, and **playability** that increases the challenge as the player progresses through the game.

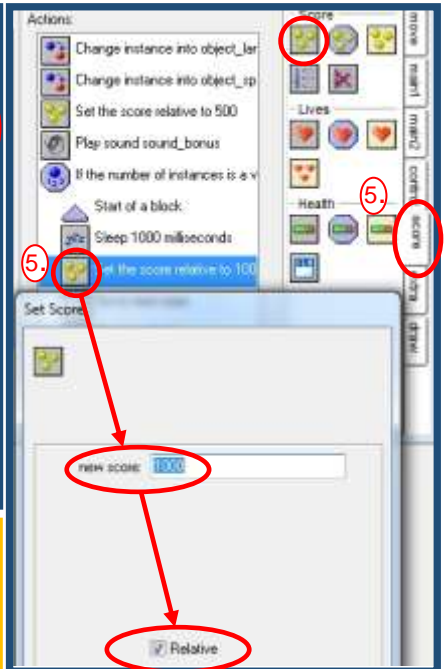
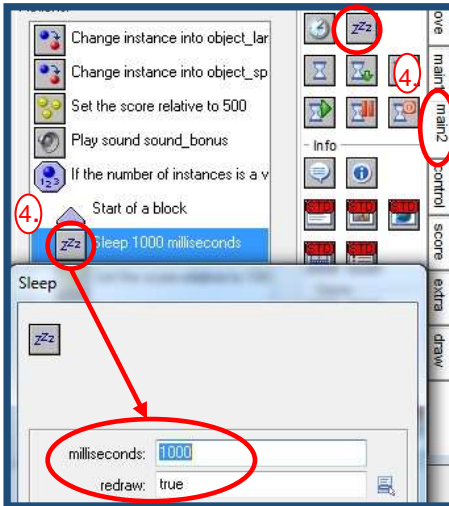
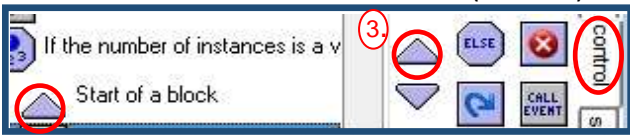
## Test for the number of moon instances

- 1- Reopen the **object\_flying** and select **Collision** with the **object\_moon**.
- 2- Include a **Test Instance Count** (control tab). Set the **Object** to **object\_moon** and leave the default settings.



**CONTINUE THIS PROCEDURE ON NEXT PAGE**

- 3- Add a **Start Block** action from the **control** tab below the condition to test the instances (count the moons).
- 4- Include a **Sleep action** from **main2** and set **Milliseconds** to **1000**. Leave redraw as **"true"**.  
**NOTE:** The prefix **"milli"** means 1/1000 (or .001). Here we are counting 1000 "milli"seconds, mathematically written as 1000 x 1/1000 (or 1000 x .001). How many seconds is this? Why is time important in the game? You can write about this in your STAGE 5 Hypothesis.
- 5- Include a **Set Score** from **score** tab with a **New Score** of **1000** and **Relative** is enabled (checked).  
**NOTE:** You want to consider what this step is about in your STEP 5 Hypothesis.
- 6- Add a **Next Room** action from **Main 1**. The transition should remain at **"no effect"**.
- 7- Add the **End Block** action (**control**) to end the conditional actions.

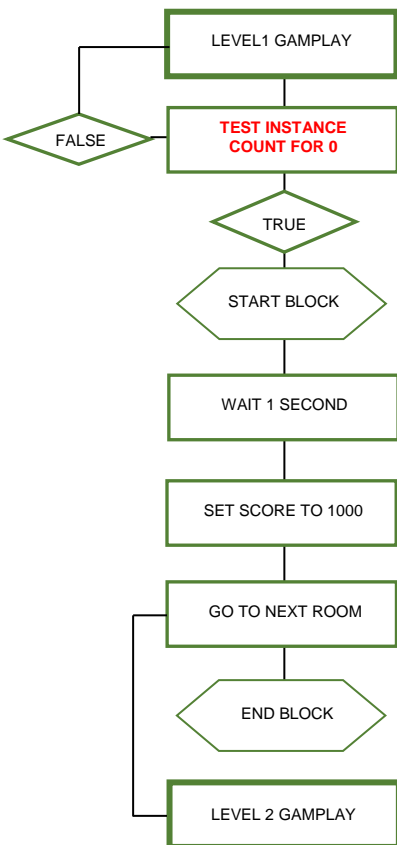


### Actions in blocks

A **block** is a number of actions grouped together as part of a **conditional action**. Like other conditional actions, an event results in an action or other event (*cause and effect*). Instead of one action as an effect, a **block** will perform all actions or tasks contained within it. Sometimes the word *sub procedure* is used. *Sub procedures* and *blocks* will be studied in future tutorials, especially with *scripting*. You can read more in Concepts Explained.

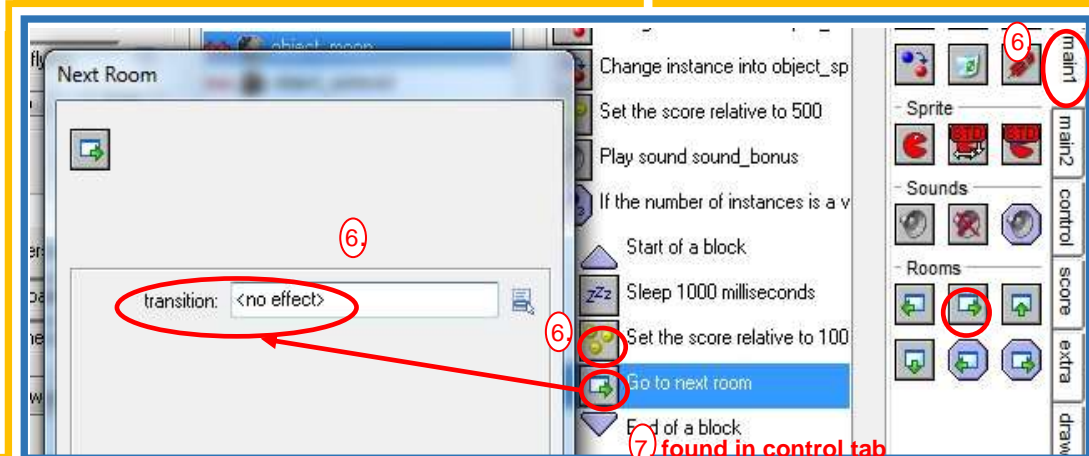
### Planning conditional action

In this stage you are building your logic system for scoring using a **block**, which will create the effect of numerous actions occurring due to one conditional event. This *flow chart* is what a game programmer might use to map out the events, actions, and gameplay for steps that program the game to advance through **levels**. This sample **block** (*sub procedure*) functions when the number of moon instances is "true", meaning **instance count = 0**. Flow charts are great planning tools that will be used in later tutorials to plan out processes for custom game builds.



### True or False Conditions

When the condition that causes an effect is present, it is referred to as "true" and "false" when it is not present. A "true" conditional statement for a **block** might be: **IF "true" THEN run all actions in the block.** The "false" condition might be: **IF "false" THEN skip block and go to next step.**



**SAVE**

## DO THIS ON YOUR TUTORIAL GUIDE

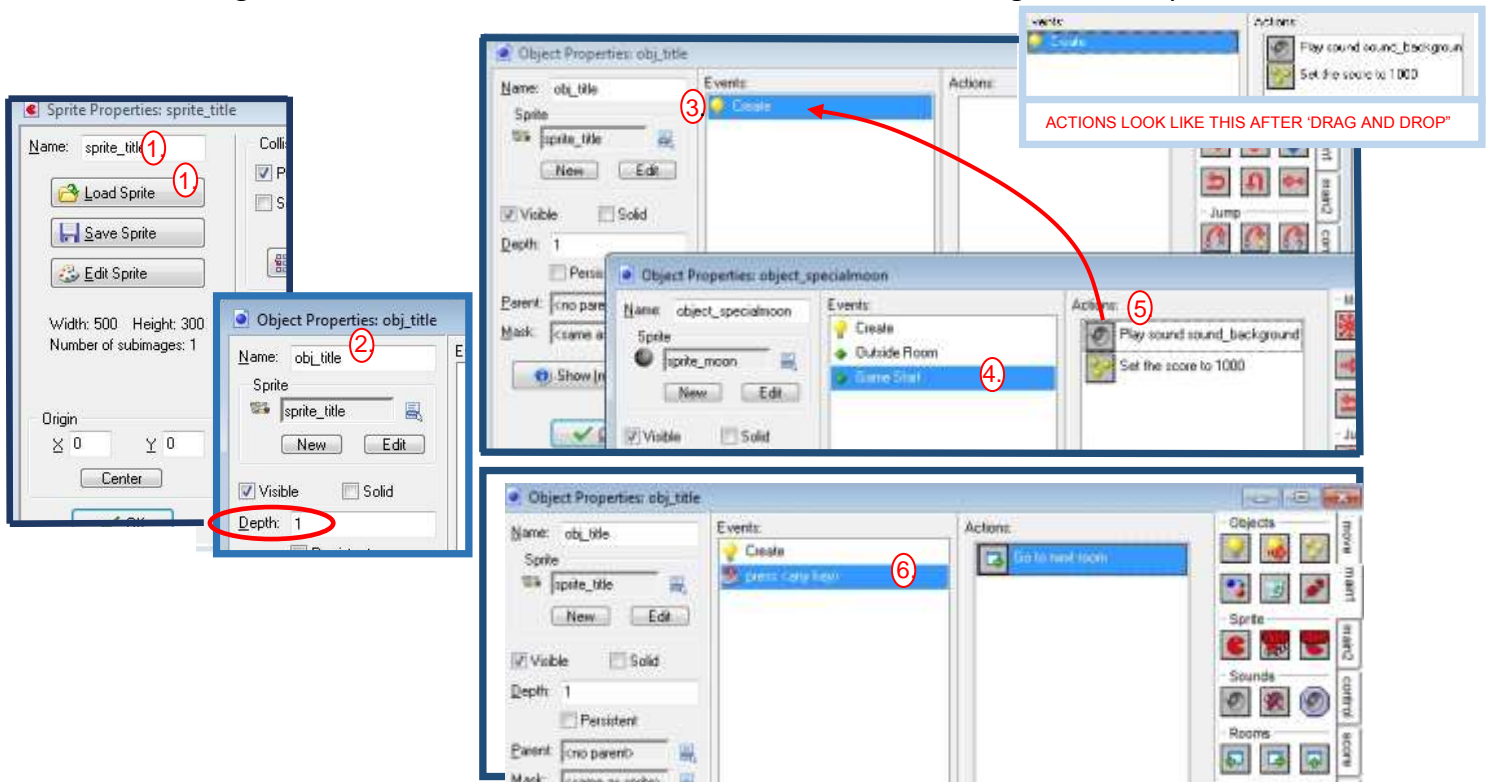
**STAGE 5 HYPOTHESIS STATEMENTS:** Now it's time to predict how objects will score points and advance through levels.

- Explain how properties will prevent a player from cheating and how a player will be penalized (time plays a role in this).
- What will you have to do to be rewarded and how do you earn points?
- Explain the effect that property changes in new levels will have the challenge of the game.

## STAGE 6:

### Creating a Title Screen for the Front End

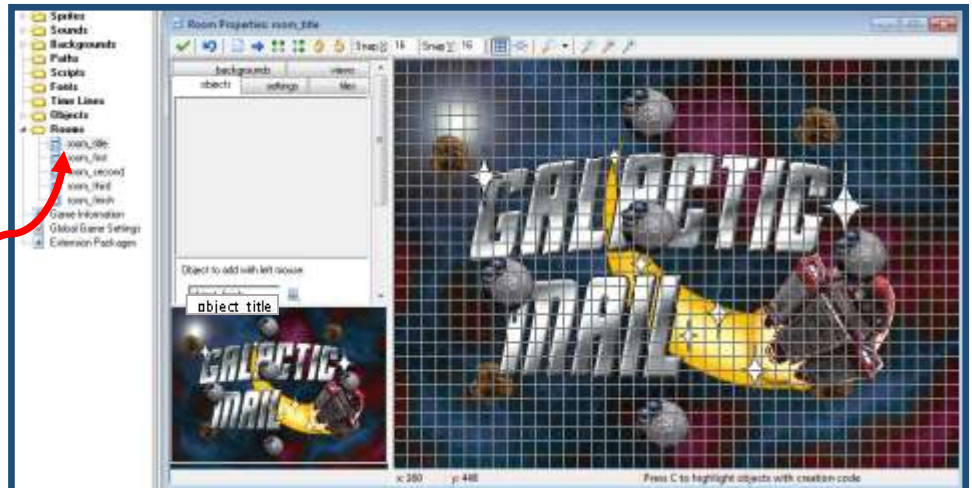
- 1- Create a new sprite called *sprite\_title* using the *Title.png* file from the assets folder.
- 2- Create a new object called *obj\_title* using *sprite\_title* . Set **Depth** at 1.  
NOTE: Based on this setting, where will the moons and asteroids appear in relation to the title? Write about this in your STAGE 7 Hypothesis.
- 3- Add a **Create** event.
- 4- Open the special moon properties and select the **Game start** event.
- 5- **Drag and drop** the **play sound** and **set score** actions from the special moon's Game Start into the **Create** event of the *obj\_title*. Click **OK** in the special moon properties, and the Game Start event will delete itself.
- 6- Add a **Key Press <any key>** event to *obj\_title*. Drag and drop a **Next Room** action from *main1*. Make no changes to the "Go to next room" action, since the default settings are what you need.



**SAVE**

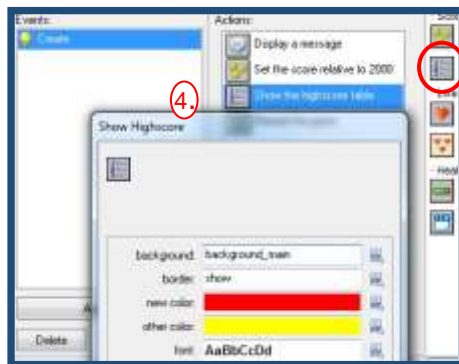
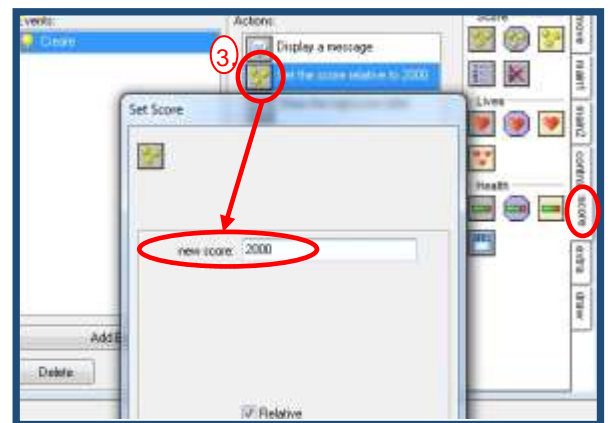
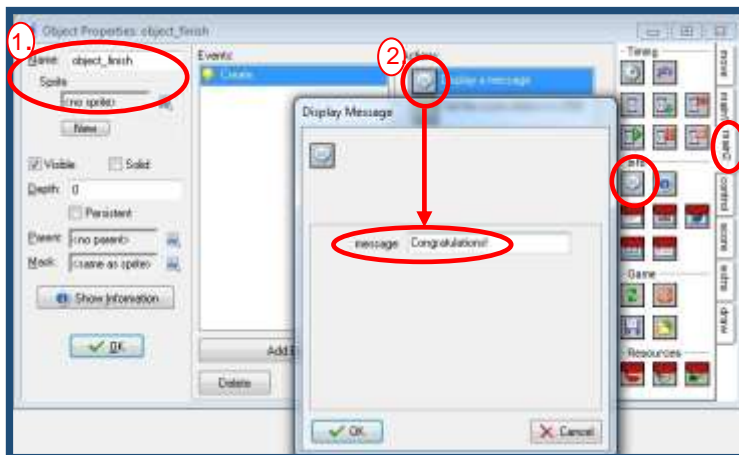
## Title Screen: Creating a new title room resource for the game

- 1- Create a **new room** named **room\_title** and give it an appropriate caption. Set the room's background like before.
- 2- Add a few moon and asteroid instances in the room.
- 3- Place an instance of the title in the center of the room.
- 4- Close the room properties.
- 5- To make sure the title loads first, **drag the new room to the top of the rooms listing**.



## Creating a new finish object resource for the game

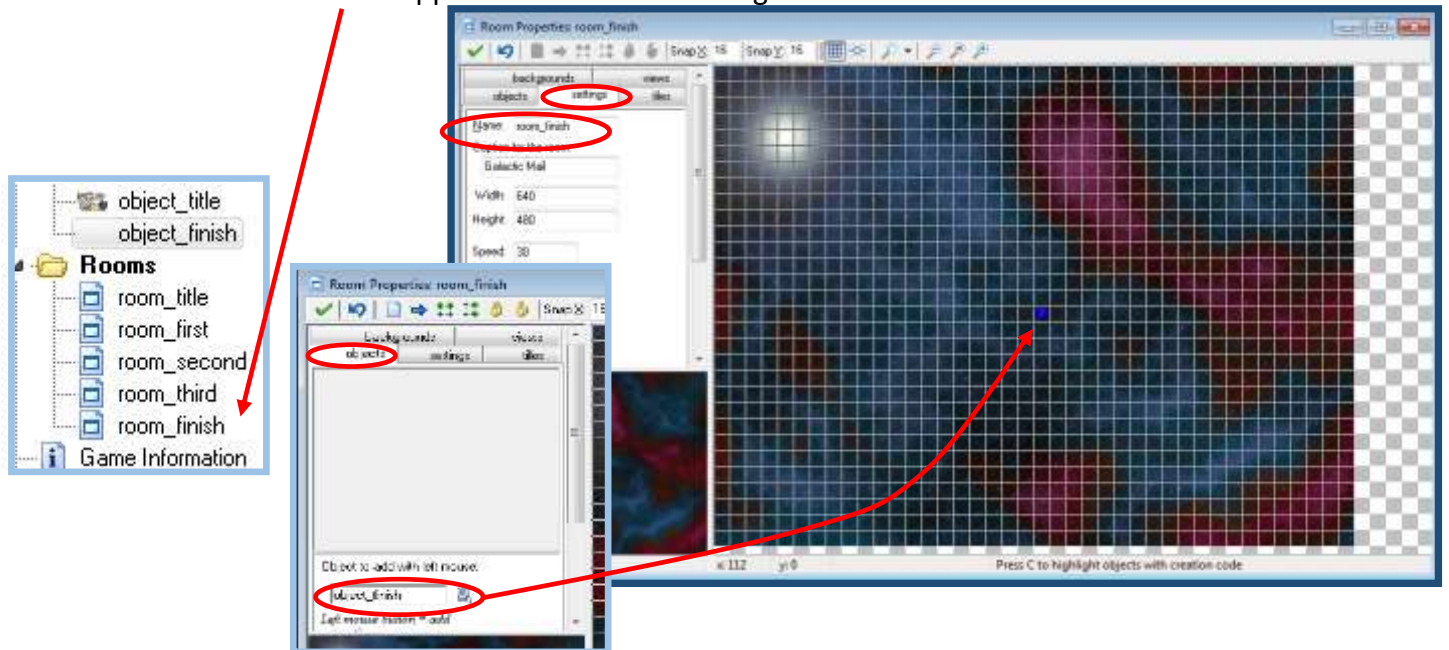
- 1- Create a new object named **object\_finish**. No sprite is needed.
- 2- Add a **Create** event and include a **Display Message** action from the **main 2** tab. Set the message to "Congratulations! You are an awesome mail carrier!"
- 3- Add a **Set Score** action with a **New Score** of **2000** and **Relative** enabled.
- 4- Add a **Show Highscore**. Use the same settings for Background, Other Color and font as last Set Score.
- 5- Add a **Restart Game** action.



**SAVE**

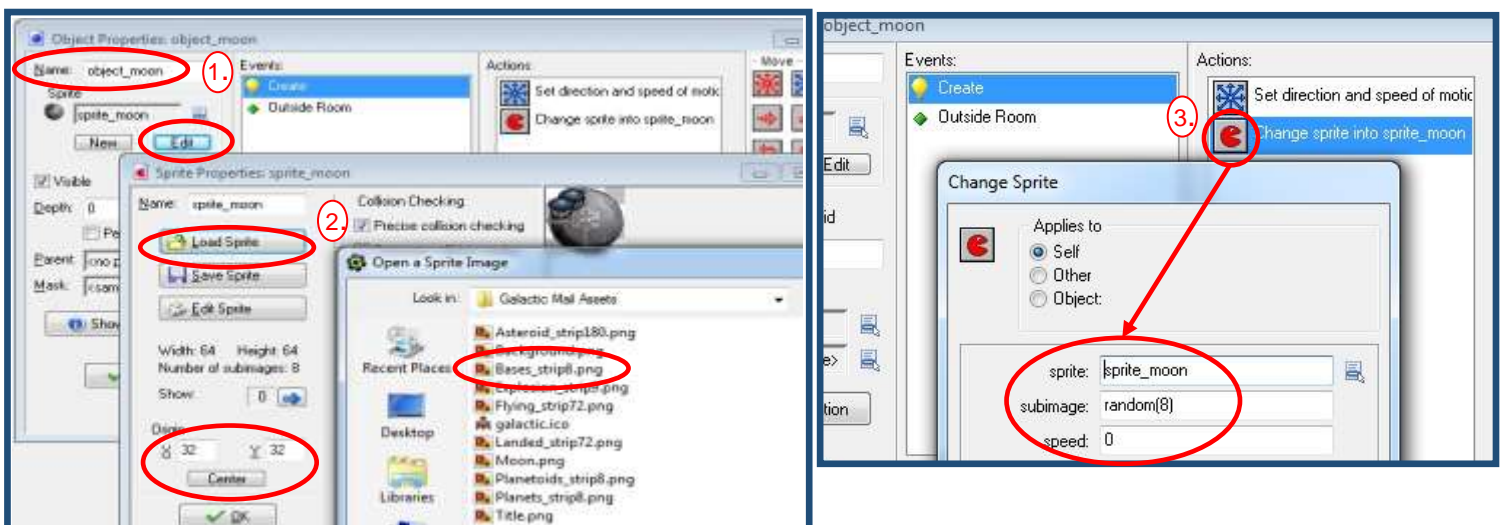
## Creating a new finish room resource for the game

- 1- Create a new room called **room\_finish** and place an instance of the finish object inside it. With no sprite, you will see a blue ball with a red question mark. This doesn't show on the game but tells you there is an invisible object there.
- 2- Make sure this room appears last in the list. Drag it there if it does not.



## Editing the moon and asteroid objects

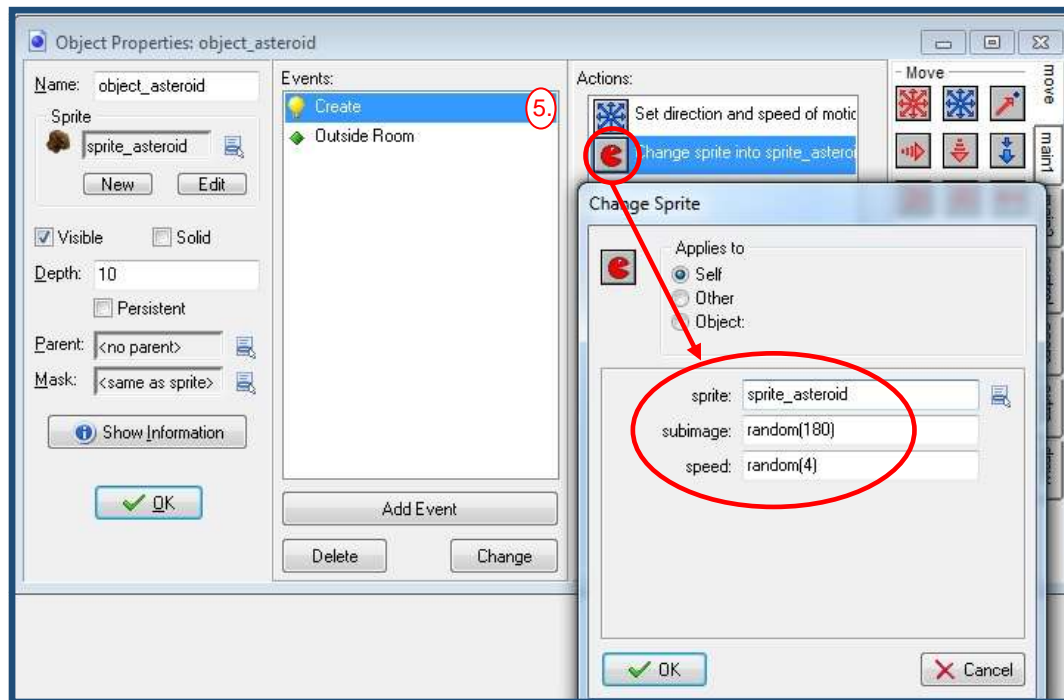
- 1- Open the properties for the **object\_moon** and click the **Edit** button below the sprite's name.
- 2- When the properties open, click **Load Sprite** and change the sprite to **Bases\_strip8.png**. Don't forget to **Center** it and click **OK** to close the form.
- 3- Back in the Moon Object properties form, select the **Create** event and include a new **Change Sprite** action. Select the moon sprite and type **random(8)** in the **Subimage**. Set **Speed** to **0**.  
NOTE: Try this! Go to your Assets Folder and click on the **Bases\_strip8.png** asset so it opens in a picture viewer. Study the *sub-images*. Can you predict what they will be doing on the moons? This will be useful in the STAGE 6 Hypothesis.
- 4- Close the action properties and Moon object properties forms.



**SAVE**



- 5- Open the object properties for **object\_specialmoon**. Include an identical **Change Sprite** action to the **Create** action. It will look like the change sprite (above/right) except it will be the properties window for *object\_special moon* instead of *object\_moon*).
- 6- Open the properties for the **object\_asteroid** and include a **Change Sprite** action in the **Create** event as well. Choose the **sprite\_asteroid** and type **random(180)** in the **Subimage**. Also type **random(4)** in the **Speed** property. Close the form.



**SAVE**

### DO THIS ON YOUR TUTORIAL GUIDE

**STAGE 6 HYPOTHESIS STATEMENTS:** Now it's time to predict how all of the room changes effect the game start and the climactic ending!

- Explain how your depth settings for objects will effect the appearance of the front end (first room).
- What are the "bases" and what is their relation to the moons. What will they do?

Now it's time to test your game, so go ahead and click on the green triangle ( ▶ ).

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

### DO THIS ON YOUR TUTORIAL GUIDE

**STAGE 4, 5, & 6 TEST & EVALUATION:** Now test the validity of your hypothesis statements.

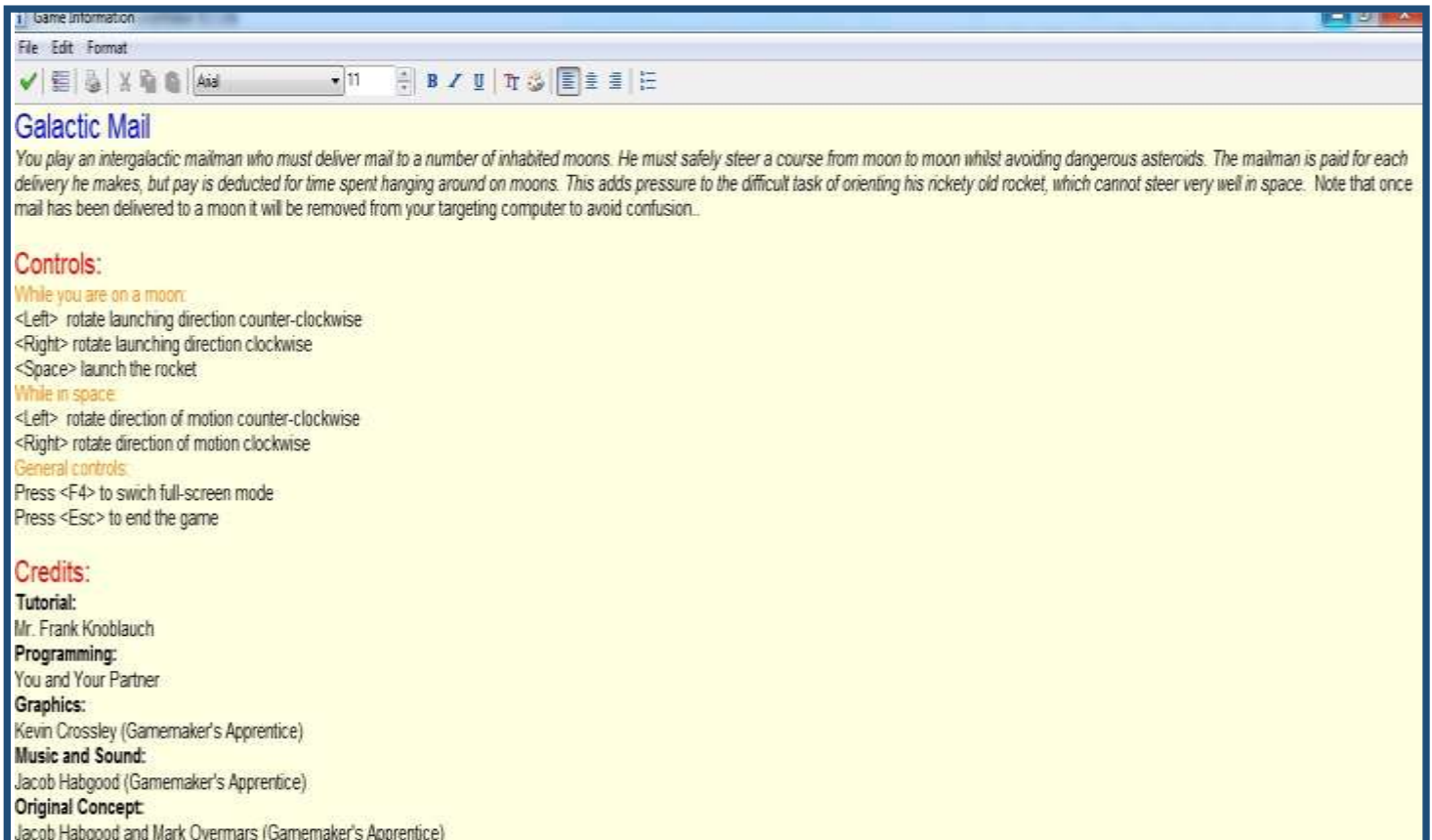
- Does the action of your objects meet what you anticipated in your hypothesis statements?
- If not, what was the difference in behaviors and how did you fix the problem?

**YOUR GAME SHOULD MEET THE DESCRIPTION IN THE SAME "GAME INFORMATION" WINDOW BELOW. COMPLETE THE HELP INFORMATION STEPS ON THE NEXT PAGE.**

**NOTE:** You need to provide some help to explain the controls and basic idea of the game. We do this in the **game information** area. It is also an opportunity to award **credit** to the people for their creative and technical work that made this game possible.

### Adding Game information to the game

- 1- Double-click the **Game Information** at the bottom of the resource list.
- 2- A text editor will open. You can set the fonts and colors.
- 3- Typically you end the name of the game, the name of the game creator(s), a short description of the goals and a list of the controls.
- 4- When you are done, click on the **green checkmark** in the top left to close the editor.



### Credit when credit is due!

Just like the rewards of gameplay for the game players, the *professionals* who devote their time and talent to the game project deserve fair *compensation*, or pay, for their work. It is also important to them that they are given **credit** for their creative contributions, technical work, and specific tasks that are present in the finished game product. Game development companies have a responsibility to every person who performed tasks in a game project. This includes listing their names, along with their specific role, in **credits** presented within the gaming *software*. Credit is important for two reasons. First, it is a record that the workers can refer to in order to get jobs in the future. Then, it can be a source of motivation for them. They know their work has been valued and appreciated, so they are likely to continue working hard to contribute more to future game projects.

Professionals working in game development not only devote time and talent to game projects, they often spend years of in schools earning advanced degrees, or gaining work experience in their fields as preparation for their work. They include *writers, artists, programmers, and producers*, just to name a few. They perform tasks such as writing *concept documents, drawing storyboards, sound and graphic asset development, writing code and programming*, and much more. Having their names in **credits** is very important for their *careers*. There will be separate units in this course for *career preparation, careers and compensation*, and studies in specific tasks related to game development.

**IT WAS FUN TO BUILD. NOW HAVE FUN PLAYING GALACTIC MAIL. MAKE SURE YOU SAVE IT FIRST!**