

Directions for Koalabr8 Game

Directions from the Game Maker's Apprentice book by Jacob Habgood and Mark Overmars

Game Description

A Colony of koala bears have been captured by the evil Dr. Bruce for use in his abominable experiments. The koalas manage to escape from their cages only to find that the doctor has implanted some kind of mind control device in their brains. The only way they can overpower the controlling effect is to combine their thoughts and all perform the same actions at once. The koalas must work together to find their way past the many dangers in the doctor's laboratory and escape to freedom.

The arrow keys will simultaneously move all of the bears on a level, except bears whose paths are blocked by a wall or another bear. Each level will be a hazardous maze that is completed by getting all of the koalas to an exit. However, if a koala touches a dangerous hazard on the way, then he dies and the level must be replayed. The game will contain a number of fatal and nonfatal hazards shown in the following list:

- Fatal hazards
 - Explosive TNT
 - Moving circular saws
- Nonfatal hazards
 - Red exits—Allow any number of koalas to exit the level
 - Blue exits—Allow a single koala to exit the level
 - Locks—Block the path of koalas (red, blue, green)
 - Switches—Open locked passageways (normal, timed, and pressure)
 - Boulders—Can be pushed by koalas and destroy other hazards

We will use the same type of game framework used in the previous game. This consists of a front-end with buttons to start a new game, load a saved game, show help, and quit the game as well as a completion screen that congratulates the player. However there is no score in this game so there will be no high score table.

Creating the front-end:

1. Start a new game.
2. Create sprites using the following file from the Resources: Title_png, Button_start.png, Button_load.png, Button_help.png and Button_Quit.png. Name them appropriately.
3. Create a **background** using the file Background.png.
4. Create **sounds** using the Music.mp3 and Click.wav files.
5. Create a **title object** using the start sprite. Add an **Other, Game Start** event and include a **Play Sound** action. Select the **background** music and set **Loop** to **true**.
6. Create a **start button** object using the start sprite. Add a **Mouse, Left Pressed** mouse event and include an action to **play** the click **sound** followed by an action to **move** to the **next room**.
7. Create a **load button** object using the load sprite. Add a **Mouse, Left Pressed** mouse event and include an action to **play** the click **sound** followed by an action to **load** the **game**.
8. Create a **help button** object using the help sprite. Add a **Mouse, Left Pressed** mouse event and include an action to **play** the click **sound** followed by an action to **show** the **game information**.
9. Create a **quit button** object using the quit sprite. Add a **Mouse, Left Pressed** mouse event and include an action to **play** the click **sound** followed by an action to **end** the **game**.
10. Create a **room** using the **background**, and place the title and four object buttons in it.

Creating the completion screen:

1. Create a sprite using the Congratulations.png file.
2. Create a new object using this sprite. Add a **Create** event and include a **Set Alarm** action to set **Alarm0** using **120** steps.
3. Add an **Alarm, Alarm0** event and include an action to **move to the front-end room**.
4. Create a **completion room** using the background and place an instance of the congratulations object in it.

Changing the game settings:

1. Double click the **Game Information** near the bottom of the resource list and create a short help text based on the games description.
2. Double click the **Global Game Settings** at the bottom of the resource list.
3. Switch to the **Other** tab and **disable the two options Let <Esc> end the game and let <F5> save the game and <F6> load a game**.

SAVE your work as koala1.gmk in your folder.

Creating the wall object:

1. Create a sprite called sprite_wall using the Wall.png file.
2. Create a new object called object_wall using this sprite and enable **Solid**.

Creating the koala sprite:

1. Create a sprite called sprite_koala_left using the koala_left_strip8.png file.
2. Create sprites using the files koala_right_strip8.png, koala_up_stirp8.pgn, koala_down_strip8.png and koala_stand.png.

Since the grid and koalas are 40x40 we will use the Check Grid action. This conditional action lets us test for a koala being exactly lined up with the corridor of the maze. This means we ignore a player's key presses (which would make the koala bump into the walls) and wait until Game Maker knows the koala is where it belongs. We will also check that there is nothing blocking the koala's path.

Creating the koala object:

1. Create a new object called object_koala using the standing koala sprite. Set the objects Parent to be the wall object.
2. Add a **Keyboard, <Left> event** and include a **Check Grid** action (**control**). Set both **Snap Hor** and **Snap Vert** to **40** to make grid 40x40. Enable **NOT** option so event checks for koalas not being aligned to grid.
3. Include the **Exit Event** action. This stops other actions from being performed. Since it is put under the Check Grid, no other actions are performed when the koala is NOT aligned with the grid.
4. Include the **Check Object** action. Indicate the **wall object** and enable the **NOT** option. Set **X=-40** and **Y=0** and enable **Relative** option. This makes it look for no wall objects one grid to the left of the koala.
5. Include the **Start Block** action followed by a **Move Fixed**. Select the **left** arrow and **Speed=5**.
6. Include a **Change Sprite** and indication the left facing koala. Set **SubImage** to -1 so animation keeps playing.
7. Add an **End Block**.
8. Now to check if a wall is in the way we need other conditions to occur. Add an **Else** action and follow it by a **Start Block** action.
9. Include a **Move Fixed** action, choose the **middle** square and **Speed=0**.
10. Include the **Change Sprite** action using standing sprite with **SubImage =-1**.
11. **End Block**.

12. Repeat the steps with similar rules for **Right, Up** and **Down** arrow keys. Make sure you choose the correct direction for each **Move Fixed** action, the correct sprite for the **Change Sprite** action and the correct **X** and **Y** values for the Check Object action. **When moving up and down, the Y increases as you move down the screen.**

Adding a Begin Step event to the koala object:

1. Add a **Step, Begin Step** event to the koala object and include **Check Grid**. Set **Snap hor** to 40 and **Snap vert** to 40 but leave **NOT DISABLED**.
2. Include a **Start Block** action with a **Move Fixed** action. Set **Speed** to 0 and select the **center square**.
3. Include a **Change Sprite** action, chose the **standing koala** sprite with **Subimage** set to -1.
4. **End Block** and close koala object's properties.

Creating the controller object:

1. Create a new object called object_controller and leave it with no sprite assigned.
2. Add a **Key Press, Others, <Escape>** event. Include the **Different Room** action and chose **front-end room**.

Creating a test room:

1. Right click the completion room and select **Insert Room** from the pop up menu.
2. Switch to **Settings** tab and name the room.
3. Switch to the **backgrounds** tab and give the room the background.
4. Set **Snap X** and **Y** to 40.
5. Select the **objects** tab and create a maze using the wall object. Leave the top row free. We will display the game information later on. Put 3 or 4 koalas in the maze and 1 instance of the controller In the top left corner of the room *above the wall.

SAVE YOUR WORK as koala2.gmk

Creating the exit objects:

1. Create sprites called sprite_exit1 and sprite_exit2 using Exit1.png and Exit2.png.
2. Create an object called object_exit1 and use the first exit sprite, setting the **Depth** to 10. Add a **Collision event** with the Koala object and include a **Check Grid** conditional action. Select **other** from **Applies** to Koala and set **Snap hor** and **Snap Vert** to 40.
3. Include a **Start Block** action.
4. Include a **Destroy Instance** action and select **Other** from **Applies** to koala.
5. Include a **Set Lives** action with **New Lives** set to 1 and **Relative** enabled.
6. Include an **End Block** and close the form.
7. Create the **other exit object** using the sprite_exit2. The only difference is the need to include an additional **Destroy Instance** after step 4 with **Applies** to **Self** (the exit).

Adding events to handle rescued koalas in the controller object:

1. Create a sprite called sprite_rescued using rescued.png.
2. Reopen the properties for the controller object.
3. Add a **Create** event and include the **Set Lives** action with **New Lives** set to 0.
4. Add a new **Step, Step** event include the **Test Instance Count**. Set **Object** to Koala and **Number** to 0. When no koalas left, player has completed the level.
5. Include the **Start Block** action followed by the **Sleep** action. Set **Milliseconds** to 2000 (2 seconds).
6. Include the **Next Room** action and set **Transition** to Create from **Center**.
7. Include an **End Block** action.
8. Add a new **Draw** event. Switch to the **Draw** tab and include the **Draw Sprite**. Set **Sprite** to rescued sprite, **X** to 10 and **Y** to 0. Leave **Relative** disabled.

9. Include the **Draw Life Images** with **Image** set to the standing koala sprite. Set **X** to 150 and **Y** to 0. This draws the sprite once for each of the player's lives (saved koalas).

Add some exits to the test room and try the game to see if it works properly. Add a **duplicate** room to see if it moves to the next room.

Creating the dead koala object:

1. Create a sprite called `sprite_koala_dead` using `Koala_dead.png`.
2. Create a new object called `object_koala_dead` using this sprite. Give it a **Depth** of -10 so it appears in front of all objects.
3. Add a **Create** event and include a **Move Free** action. Set **Direction** to 80 and **Speed** to 15.
4. Include a **Set Gravity** action with **Direction** to 270 (down) and **Gravity** to 2.
5. Add an **Other, Outside** room event including a **Sleep** action. Set **Milliseconds** to 1000 (1 second).
6. Include the **Restart Room** action with **Transition** left as **<no effect>**.

Creating the TNT object:

1. Create a new sprite called `sprite_TNT` using `TNT.png`.
2. Create a new object called `object_tnt` using this sprite.
3. Add a **Collision** event with the koala object. Include a **Change Instance** action, select **Other** from **Applies to** and set **Change Into** dead koala with **Perform events** set to **Yes**.

Editing the controller object to fix the dead koala bug:

1. Double click the controller object to open it and select the **Step** event.
2. At top of action list, include another **Test Instance Count** and set **Object** to dead koala. The block will now only be performed if both conditions are true.
3. Add a few **TNTs** on your level.

Creating the saw objects:

1. Create sprites called `sprite_saw_horizontal` and `sprite_saw_vertical` using `saw_horizontal_strip2.png` and `saw_vertical_strip2.png`.
2. Create an object called `object_saw_horizontal` and use horizontal sprite. Add a **Create** event and include **Move Fixed**. Select the **right** arrow and set **Speed** to 4 (slightly slower than the koala to give players a chance to escape).
3. Add a **Collision** event with the wall object and include the **Reverse Horizontal** action.
4. Add a **Collision** event with the koala object. Include the **Change Instance** action and select **Other** from **Applies to**. Set **Change Into** to the dead koala object and set **Perform** events to **Yes**.
5. Create an object called `object_saw_vertical` using the `sprite_saw_vertical`. Add a **Create** event and include the **Moved Fixed** action. Select the **down** arrow and set **Speed** to 4.
6. Add a **Collision** event with the wall and include the **Reverse Vertical** action.
7. Add a **Collision** event with the koala object. Include the **Change Instance** and select **other** from **Applies to**. Set the **Change Into** the dead koala object and set **Perform** events to **Yes**.
8. Create a new level with some moving saws. Add a cheat in the controller object so that pressing **N** will move you to the next room and pressing **P** will move you to the previous room. **Save this as Koala4.gmk.**

To make our maze look more interesting, we are going to use tiles. Tiles are a new background resource that consists of small 40x40 images. These are called tile sets.

Creating the tile set:

1. Create a new background called back_tiles using the wall_tiles.png.
2. Enable the **Use as Tile** set option.
3. Set **Tile width** and **height** to 40 and leave other values to 0.
4. Close the form.

Adding tiles to your rooms:

1. Double-click the first room in the resource list and right click to delete all wall instances.
2. Click the tiles tab. Select the background with the tiles.
3. Click on one of the tile images to select it and it will become outlined. Now you can place and remove tiles as needed. Hold the **SHIFT** key to add multiple copies.
4. Select and place the tiles to create a maze.
5. Select the object tab and place instances of the wall on top of the wall tiles.
6. Repeat this process for the other rooms.

Now we will make the wall object invisible so we only see the tiles.

Making the wall object invisible:

1. Double-click the wall object.
2. Disable the Visible object in the form. Close the form.

Test the game and make sure the tiles are displayed and that koalas cannot walk through the walls.

Save this version as Koala5.gmk.

We will now add some other hazards in the form of locks. In order to unlock the lock, you must click on the switch. There will be 3 colors: blue- which disappears forever once the switch is activated, yellow will reappear 5 seconds later after the switch is opened and red will only stay unlocked when the switch is continually press. So for the red one, a koala must keep the lock open for another koala to pass through. We will start with the blue lock and switch.

Creating the blue lock and switch objects:

1. Create sprites called sprite_lock_blue and sprite_switch_blue using lock_blue.png and switch_blue.png.
2. Create a new object called object_lock_blue using the blue lock sprite. Enable **Solid**. Set **Parent** to the wall object. This makes it block the koalas.
3. Create a new object called object_switch_blue and use the blue switch sprite. Set **Depth** to 10 so it appears behind other objects.
4. Add a **Create** event and include a **Change Sprite** action. Select the blue switch sprite and set **SubImage** and **Speed** to 0.
5. Add a **Collision** event with the koala object and include the **Destroy Instance** action. Select **Object** from **Applies To** and select object_lock_blue.
6. Include a **Change Sprite** action select sprite_switch_blue with a **SubImage** of 1 and **Speed** to 0.

Creating the yellow and red lock and switch objects:

1. Create sprites called sprite_lock_yellow and sprite_switch_yellow using lock_yellow.png and switch_yellow_strip2.png.
2. Create a new object called object_lock_yellow and use the yellow lock sprite. Enable the **Solid** option. Set **Parent** to the wall object.

3. Add an **Alarm, Alarm 0** event and include the **Check Empty** action. Set **X** to xstart, **Y** to ystart and **Objects** to all. Using xstart and ystart will check that there are no collisions at the start of the lock.
4. Switch to the **Move** tab and include the **Jump To Start**.
5. Include the **Else**, followed by the **Set Alarm** action. Set **Number of Steps** to 2 so the lock tries again quickly.
6. Create a new object called object_switch_yellow using the yellow switch sprite. Set **Depth** to 10 to make it appear behind other objects.
7. Add a **Create** event and include a **Change Sprite** action. Select sprite_switch_yellow and set **Subimage** and **Speed** to 0.
8. Add a **Collision** event with the koala object and include a **Jump to Position** action. Select **Object** from **Applies to** and select object_lock_yellow. Set **X** to 1000 and **Y** to 0.
9. Add a **Set Alarm** action with 150 steps (5 seconds). Select **object** from **Applies to** and select object_lock_yellow.
10. Add a **Change Sprite** action, selecting sprite_switch_yellow with a **SubImage** of 1 and **Speed** of 0.
11. Reopen the properties for the yellow lock and select the **Alarm 0** event. Include a **Change Sprite** action at the top of the list of actions. Select **Object** from **Applies to** and select object_switch_yellow from the menu. Select sprite_switch_yellow and set **SubImage** and **Speed** to 0.
12. Create the red lock and red switch the same way. Change the time set for the alarm to 2 Steps so it resets if a koala moves away from the switch.

Creating the detonator object:

1. Create a sprite called sprite_detonator using detonator_strip2.png.
2. Create a new object called object_detonator and use the detonator sprite. Set the Depth to 10.
3. Add a **Create** event and include the **Change Sprite** action. Select the detonator sprite and set **SubImage** and **Speed** to 0.
4. Add a **Collision** event with the koala object and include a **Destroy Instance** action. Select **Object** from **Applies to** and select object_TNT.
5. Include a **Change Sprite** action. Select sprite_detonator and then set **SubImage** to 1 and **Speed** to 0.

Rocks are objects that can be pushed around by the koalas as long as there is free space to move them. Rocks will block moving saws and cause TNT to explode when pushed into it..destroying both the rock and TNT. If the rock is pushed onto the switch or detonator, it will also be destroyed, allowing the player to remove the switches without pressing them.

Creating the basic rock object:

1. Create a sprite called sprite_rock using Rock.png.
2. Create a new object called object_rock using the rock sprite. Enable the **Solid** option and set **Depth** to -5 so it appears in front of most objects but behind the dead koala object.

To push the rock in the right direction, we need to know which direction the koala was moving in when it collided with the rock. We will use the variables hspeed (horizontal speed) and vspeed (Vertical speed). The koala will be the other object involved in the collision so its speed is shown using other.hpspeed and other.vspeed. The koalas moves are a speed of 5 and the cell is 40 so we will multiply the horizontal speed by 8 to give the rock's x position and y position.

Adding actions to the rock object to allow koalas to push it around:

1. Add a Collision with the koala object.

2. Include the **Check Object** and indicate the wall object. Set **X** to $8 * \text{other.hs}$ and **Y** to $8 * \text{other.v}$. Enable both with **Relative** and **Not**.
3. Add a similar **Check Object** action for the rock object using the same settings.
4. Add 2 more **Check object** actions for the horizontal and vertical saws using the same settings.
5. Include a **Jump to Position**. Set **X** to $8 * \text{other.hs}$ and **Y** to $8 * \text{other.v}$ and enable **Relative**.
6. Add some rocks to your level and push them around to make sure they work. Now we need to make the rock destroy the TNT, switches and detonator.

Adding actions to the rock object to make it destroy things:

1. Add a **Collision** event with the TNT object and include a **Destroy Instance** with default settings. Include another **Destroy Instance** and select **Other** (to destroy the TNT) from the **Applies to**.
2. Add a **Collision** event with the detonator object and include the **Destroy Instance**. Select **Other** from **Applies to** (to destroy the detonator).
3. Add **Collision** events with the 3 switches and include **Destroy Instances** to apply to those switches.

Now we will make sure the saws do not move through the rocks.

Adding events to the saw objects to make them turn for rocks:

1. Double-click the horizontal saw object.
2. Add a **Collision** event with the rock object and include a **Reverse Horizontal** action.
3. Close the form.
4. Double click the vertical saw object.
5. Add a **Collision** event with the rock object and include a **Reverse Vertical** action.
6. Close the form.

Now make levels using combinations of locks, switches, detonators and rocks. Save as Koala6.gmk

With the levels we have created, there is the possibility that a level is unable to be completed. So we will now create a Restart Game option using the R key as a shortcut.

Creating a restart button object:

1. Create a sprite called `sprite_restart` using `Restart.png`.
2. Create a new object called `object_restart` using the restart sprite. Add a new **Mouse, Left Pressed** event. Include the **Play Sound** using the click sound and follow it with the **Restart Room** action.
3. Add a **Key Press, Letters, R** event and include the **Restart Room** action.
4. Add an instance of this new object in the top right corner of every level room.

Now we will make a way to save the game.

Save the game in the controller object:

1. Double-click the controller.
2. **Add an Other, Room Start** event and include a **Save Game** action.

Add sound effects to objects:

1. `Explosion.wav` needs to play with Collision events between TNT and koala and Destroy event of TNT.
2. `Saw.wav` needs to play in Collision events between both saws and koalas.
3. `Saved.wav` needs to play in the Collision event between two exit objects and the koala. Put inside the block so it is only played when koala is removed.

4. Rock.wav needs to play in Collision event between rock objects and the koala. You only want it to play when the rock actually moves so add a Start Block and End Block around the Jump to Position and include the Play Sound within it.
5. You may want to add a sound for pressing switches and opening locks. This may be a bit more difficult than it sounds because Collision events happen instantly. To have it play, it will be continuous and annoying. There is a sound in the resource folder if you want it.
6. Update the game information to include details about the different hazards.. unless you want players to find them out for themselves.

SAVE YOUR WORK as KoalaCompleted.gmk for me to know that this is your final product to be graded.