

## Jukskei

Build your own jukskei game with Scratch



### Step 1 What you will make

---

In Jukskei, players try to knock over a wooden peg by throwing a skey at it. It takes skill, aim, and timing. That's what your Scratch game will test!



### Step 2 Animate the player sprite

---

Open the starter project (<https://scratch.mit.edu/projects/1202448566/editor/>).



The starter project contains starter code and all the sprites you need.

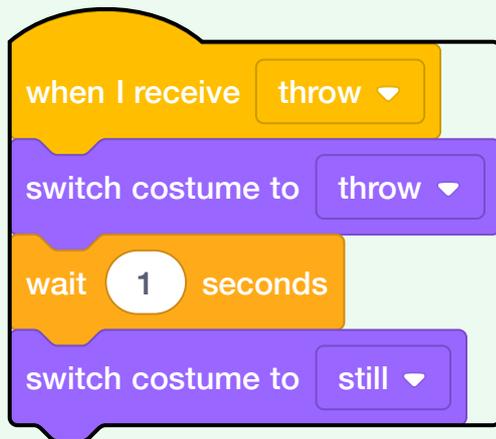
Select the **Player** sprite.



Throw!

Animate the Player with a throwing motion.

In the **when I receive** block, switch the costume.



**Test:**



- Press N to start a new game, then T to start a new throw. Check the power bar cycles from 0 to 100. The skey will not move yet.
- Press SPACE to stop the power bar. Check the Player sprite changes costume to the throw costume and then returns back to the still costume.

### Step 3 The skey sprite

---

Select the **Skey** sprite.



Move to the peg

Set the initial position of the skey.



A Scratch script for a key character, enclosed in a light green rounded rectangle. The script consists of five blocks: a yellow 'when I receive' block with a 'throw' dropdown menu; a blue 'go to x: -127 y: 29' block; a blue 'set rotation style' block with an 'all around' dropdown menu; a blue 'point towards' block with a 'Peg' dropdown menu; and a purple 'start sound' block with a 'Siren Whistle' dropdown menu. A checkmark icon is located in the top right corner of the green area.

Arc the key through the air.



```
when I receive throw
go to x: -127 y: 29
set rotation style all around
point towards Peg
start sound Siren Whistle
repeat until distance to Peg < Landing x
  turn 15 degrees
  move 10 steps
  point towards Peg
start sound Whistle Thump
wait 0.5 seconds
```

< means 'less than'



**Test:** Press `T`. Check the skye moves through the air in an arc and that throwing and landing sounds play.

## Reset

Reset the position of the skye after it lands.



```
when I receive throw
go to x: -127 y: 29
set rotation style all around
point towards Peg
start sound Siren Whistle
repeat until distance to Peg < Landing x
  turn 15 degrees
  move 10 steps
  point towards Peg
start sound Whistle Thump
wait 0.5 seconds
go to x: -136 y: -11
point in direction 120
```

### Trigger scoring

Add a broadcast message to trigger scoring.



```
when I receive throw
go to x: -127 y: 29
set rotation style all around
point towards Peg
start sound Siren Whistle
repeat until distance to Peg < Landing x
  turn 15 degrees
  move 10 steps
  point towards Peg
start sound Whistle Thump
wait 0.5 seconds
go to x: -136 y: -11
point in direction 120
broadcast score
```

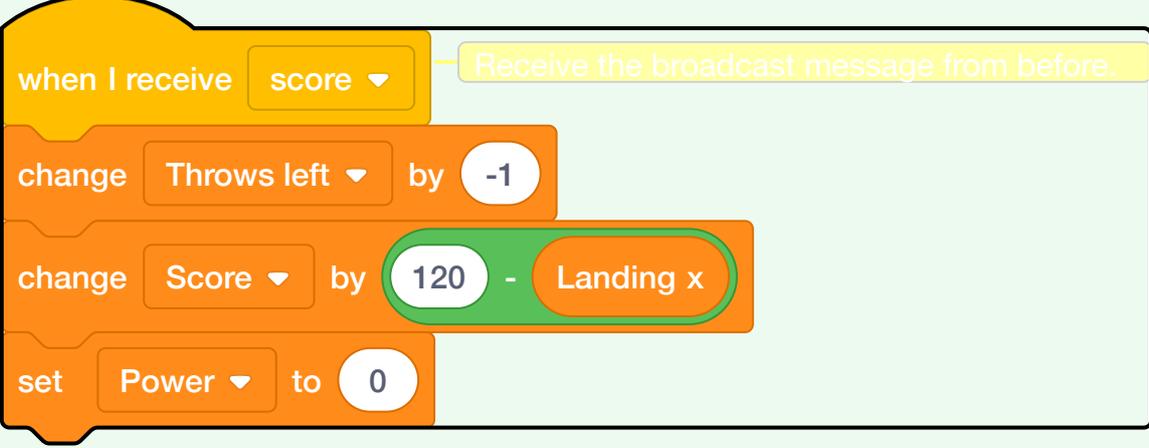
#### Step 4 Scoring on the peg

Select the **Peg** sprite.



## Change the score

The skye receives more points if it lands closer to the peg.



The code block is a yellow 'when I receive' block with 'score' selected in the dropdown. It is connected to a yellow broadcast message block labeled 'Receive the broadcast message from before.'. Below it are three orange blocks: 'change Throws left by -1', 'change Score by 120 - Landing x', and 'set Power to 0'. A checkmark icon is in the top right corner.

**Test:** Press `T`. Check the score increases and that the number of throws reduces by 1 and the power resets.



## Display the score

When there are no throws left, show the score, then reset the throws and score.

**Notice:** There is a space after the word 'Score: ' to separate the score from the word.



```
when I receive score
  change Throws left by -1
  change Score by 120 - Landing x
  set Power to 0
  if Throws left = 0 then
    say join Score: Score for 2 seconds
    set Throws left to 3
    set Score to 0
  else
```

Tell the player to throw again



```
when I receive score
  change Throws left by -1
  change Score by 120 - Landing x
  set Power to 0
  if Throws left = 0 then
    say join Score: Score for 2 seconds
    set Throws left to 3
    set Score to 0
  else
    say Press T for next throw for 1 seconds
  stop this script
```

**Test:** Press T again.



- If there are throws left, check a prompt appears to continue.
- If there are no throws left, check the score is shown and that then the throws and score are reset.

### Challenge!

## Challenge

Add a celebration!

Add a party hat and play a tune when a score is reached!

Select the **Hat** sprite.



When a score more than (>) 200 is reached, the Hat sprite will show and a sound will play.



```
when I receive score
  forever
    if Score > 200 then
      show
      play sound Dubstep until done
```

Try changing this score

You can change the sound

Add reset code.



```
when n key pressed
  hide
```

**Test:** Press N and play to a score above 200, then check the hat shows and you hear your chosen sound.



## Show the game controls

Select the Stage and open the Backdrops tab.



Duplicate the Hill backdrop and rename it 'Controls'.



Add text to the Controls backdrop to show how to control the game.



Add this single block of code to the 'when n key pressed' script.



```
when n key pressed
  stop all sounds
  set Score to 0
  set Power to 0
  set Throws left to 3
  switch backdrop to Hill
  show variable Power
  show variable Score
  show variable Throws left
```

The image shows a Scratch script for a 'when n key pressed' event. The script consists of the following blocks: a yellow 'when n key pressed' block, a purple 'stop all sounds' block, three orange 'set' blocks for 'Score' (0), 'Power' (0), and 'Throws left' (3), a purple 'switch backdrop to Hill' block, and three orange 'show variable' blocks for 'Power', 'Score', and 'Throws left'.

## Choose a different Player sprite

Choose the Player sprite and select a new sprite from the library, paint your own, upload an image, or select a random one.



Make sure your new sprite has two costumes: 'throw' and 'still' to keep the throwing animation.



Published by

<https://www.raspberrypi.org> under a  
<https://creativecommons.org/licenses/by-sa/4.0/>.  
<https://github.com/RaspberryPiLearning/jukskei>

Published by Raspberry Pi Foundation (<https://www.raspberrypi.org>) under a Creative Commons license (<https://creativecommons.org/licenses/by-sa/4.0/>).

View project & license on GitHub (<https://github.com/RaspberryPiLearning/jukskei>).

Accessibility (<https://www.raspberrypi.org/accessibility/>)

Cookies Policy (<https://www.raspberrypi.org/cookies/>)

Privacy Policy (<https://www.raspberrypi.org/privacy/>)

Translate for us (</en/projects/translating-for-raspberry-pi>)