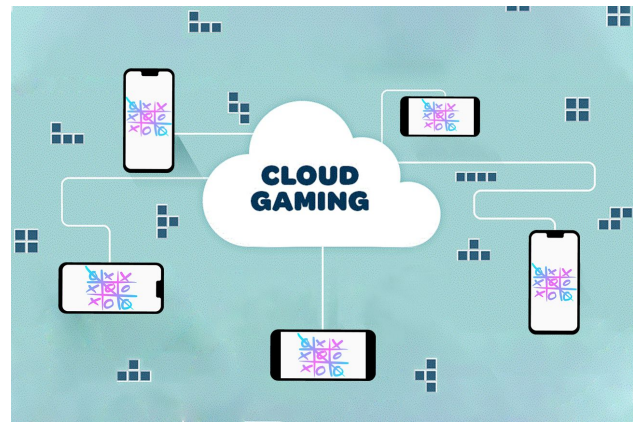


CLOUD GAMING



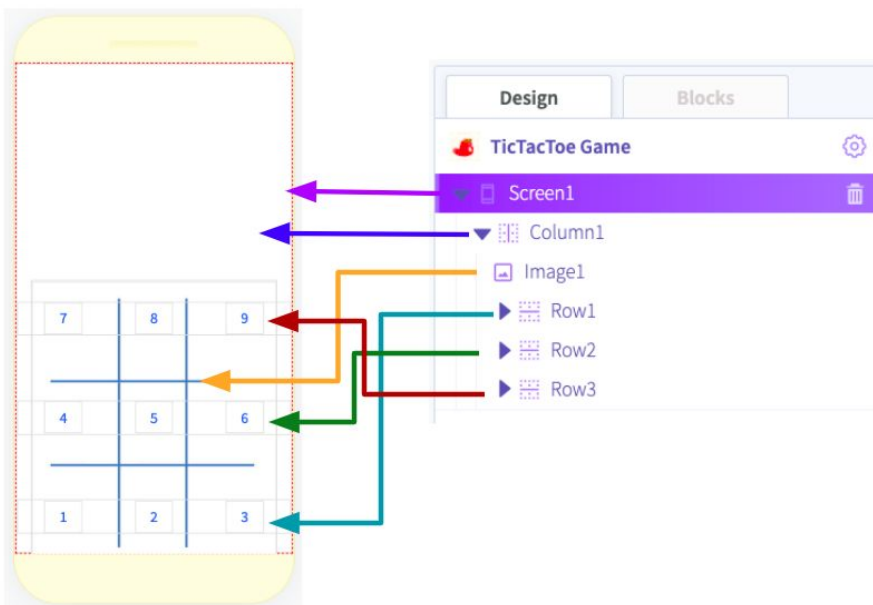
What we did:

- We learned the way cloud gaming works, to create a simple arcade game.
- We learned to apply cloud features making it a global multiplayer cloud-based game.
- We built Tic-Tac-Toe Cloud Game

How we did it:

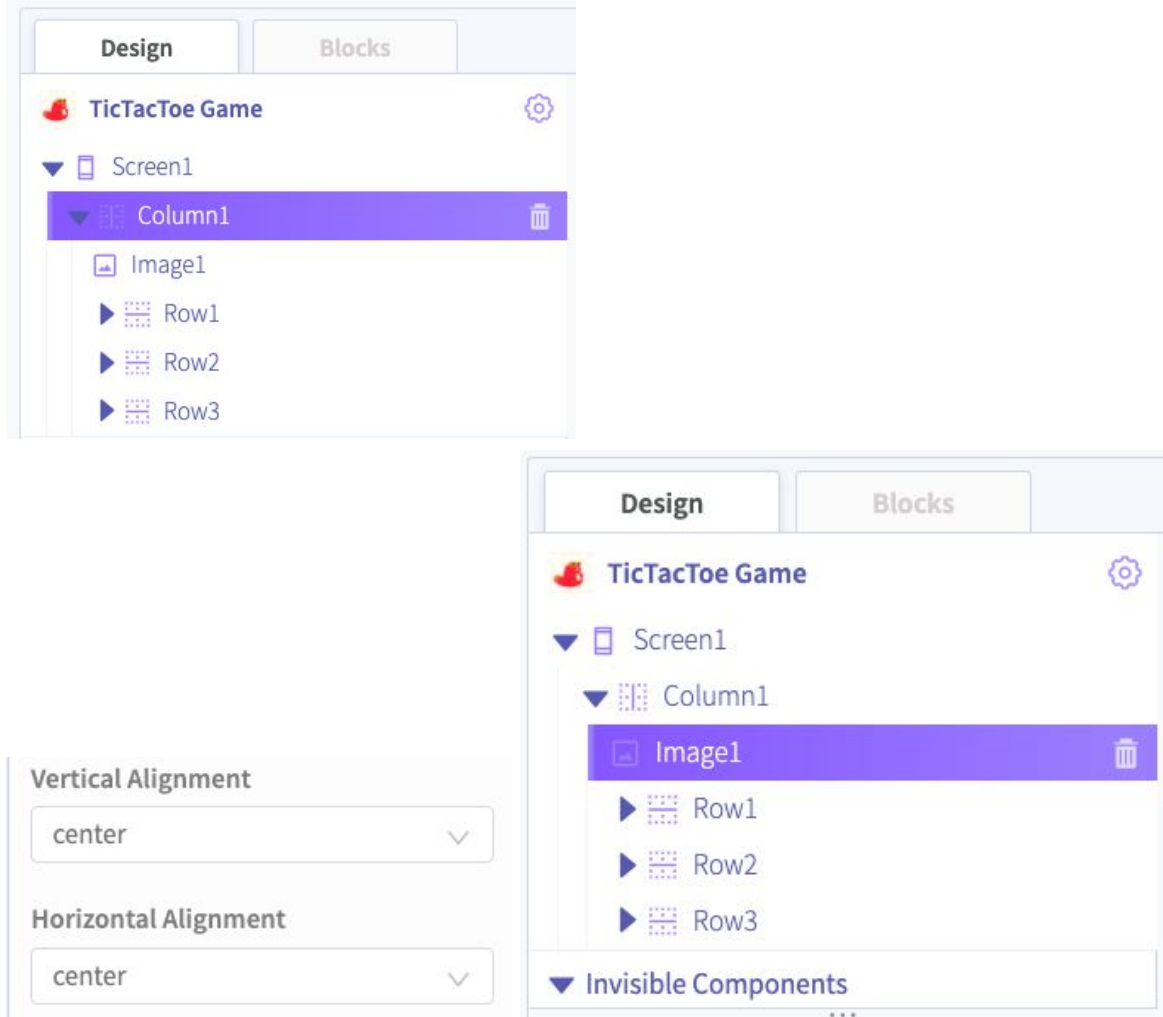
We developed a Tic-Tac-Toe game which can be played by anyone across the globe. It followed the same concept of sending and receiving data from the cloud.

Design:






We need 9 buttons with 3 buttons in each row. Initially, all the buttons have the same text

as their order Number. The properties of each component:




Note: Download the game template from the Internet.

 **Image1**  


Simple

Advanced

Picture


Tic-Tac-Toe-Game-Template.jpg 


Height

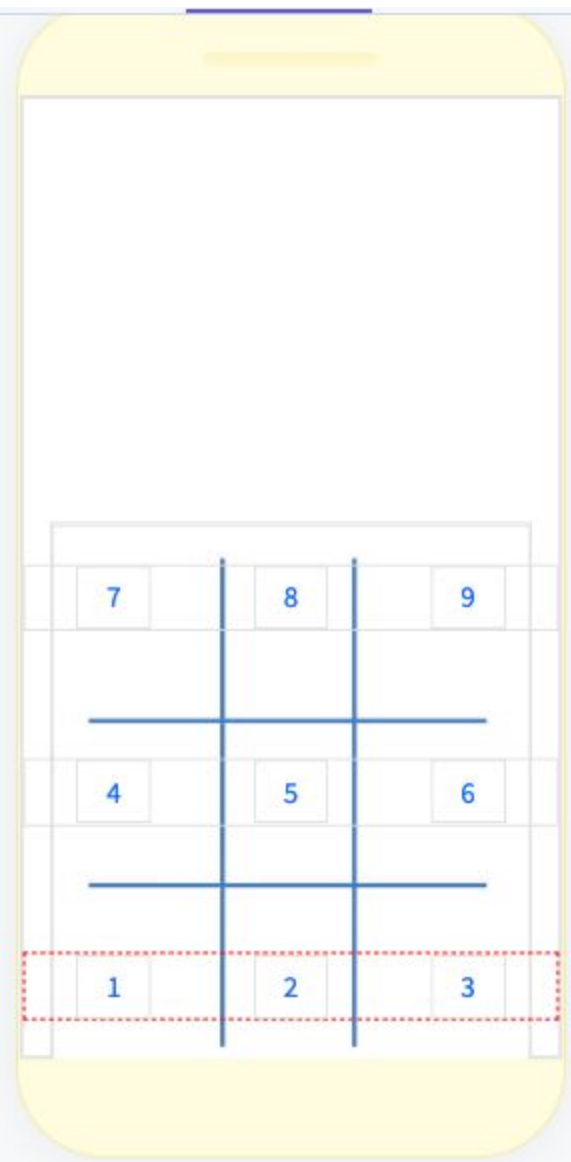
Absolute Size 

300

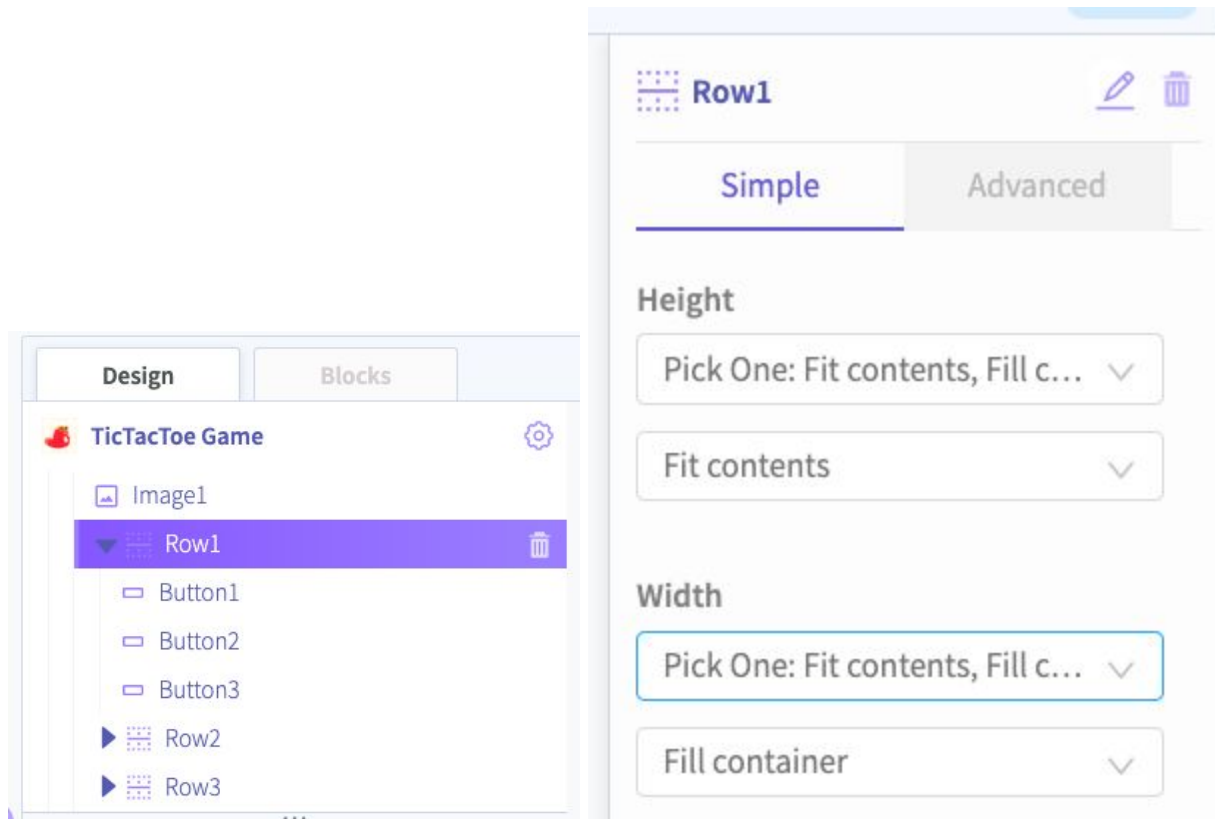
Width

Relative Size (e.g. "50%") 

90% 



A mobile app interface showing a 3x3 Tic-Tac-Toe grid. The grid is centered on a yellow background. The grid cells are numbered 1 through 9. The bottom row (cells 1, 2, 3) is highlighted with a red dashed border. The grid is composed of blue lines forming a 3x3 grid of squares.



Horizontal Alignment

space-around

Vertical Alignment

center

Scrollable

☐ false

Margin

top

-70px

bottom

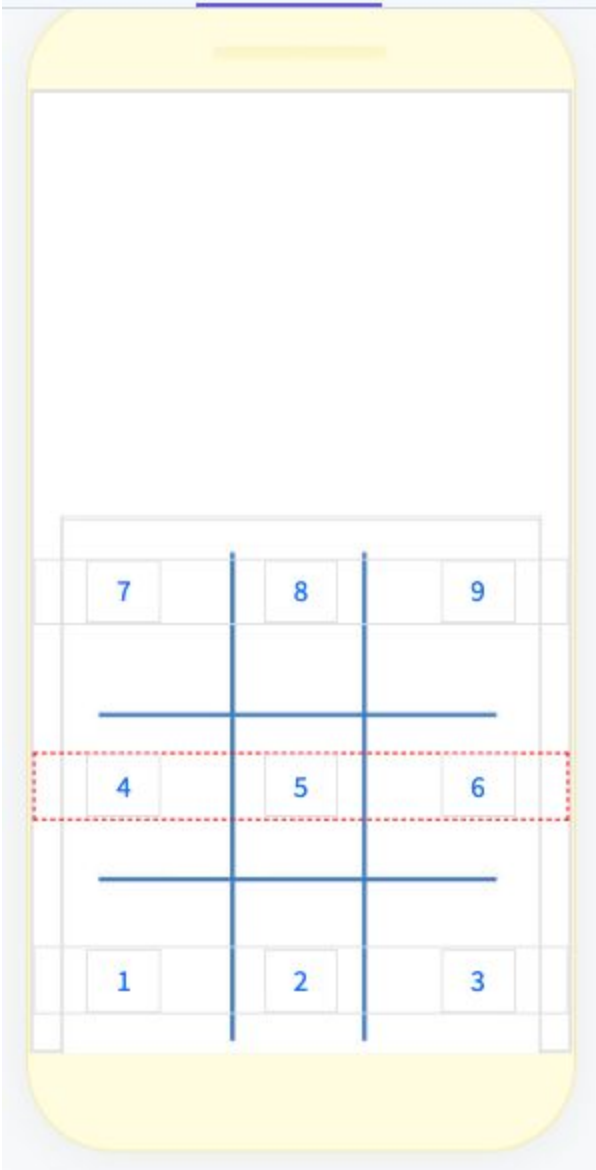
0px

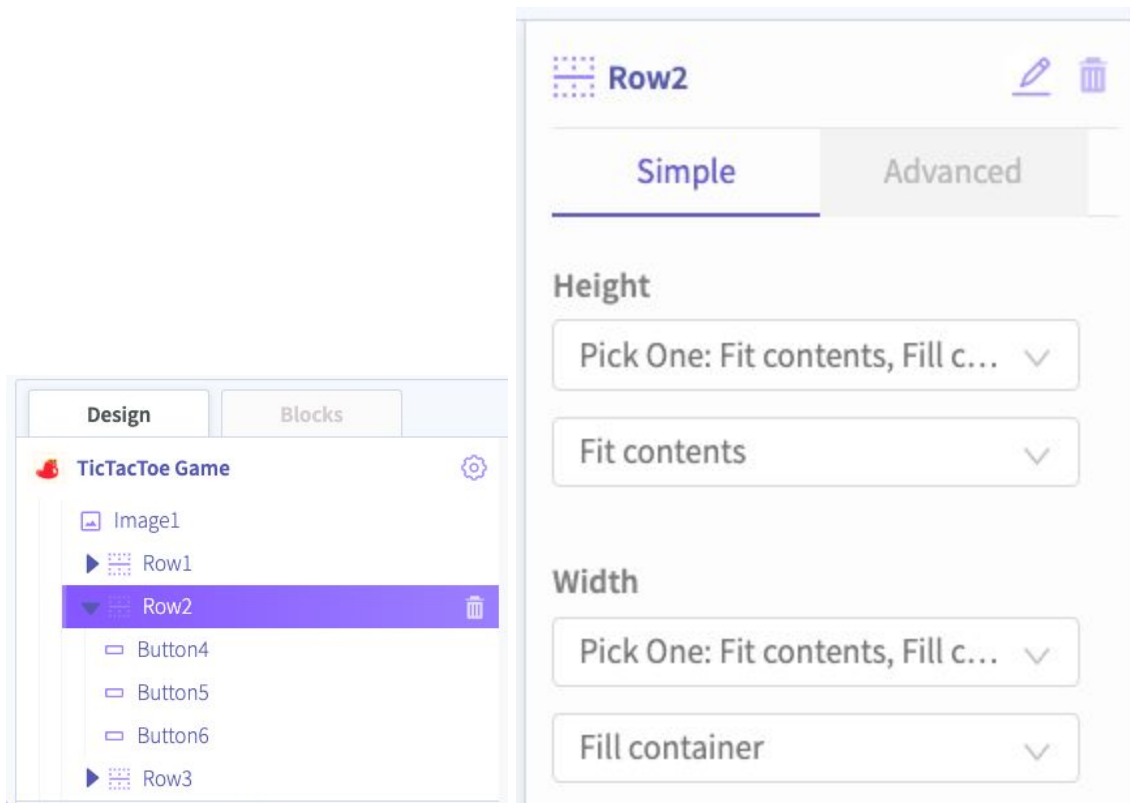
left

0px

right

0px





Horizontal Alignment

space-around ▼

Vertical Alignment

center ▼

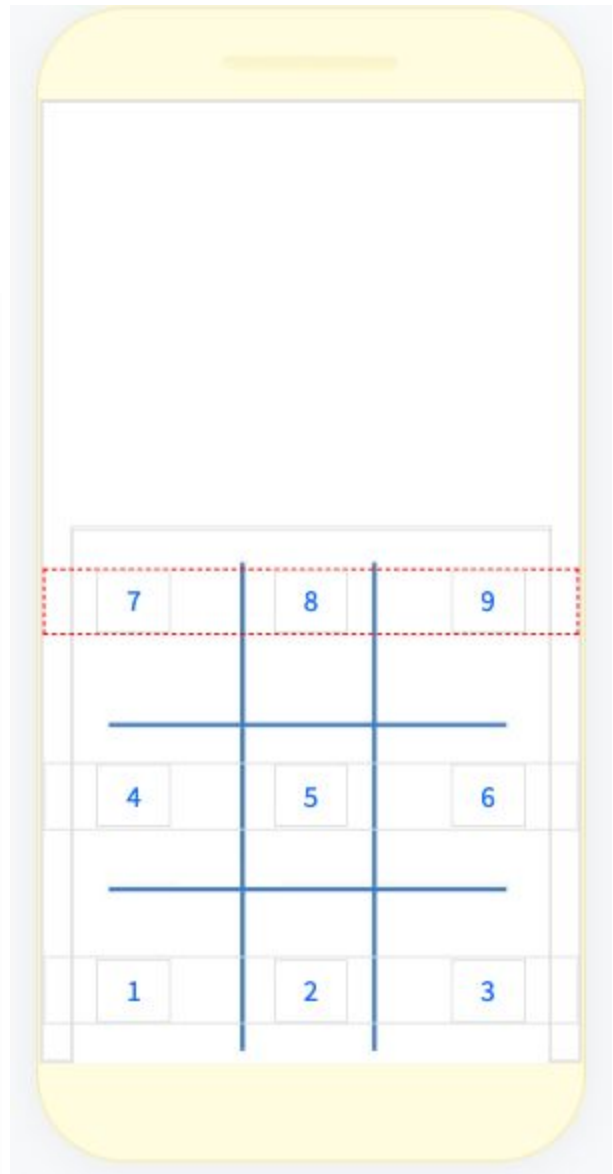
Scrollable

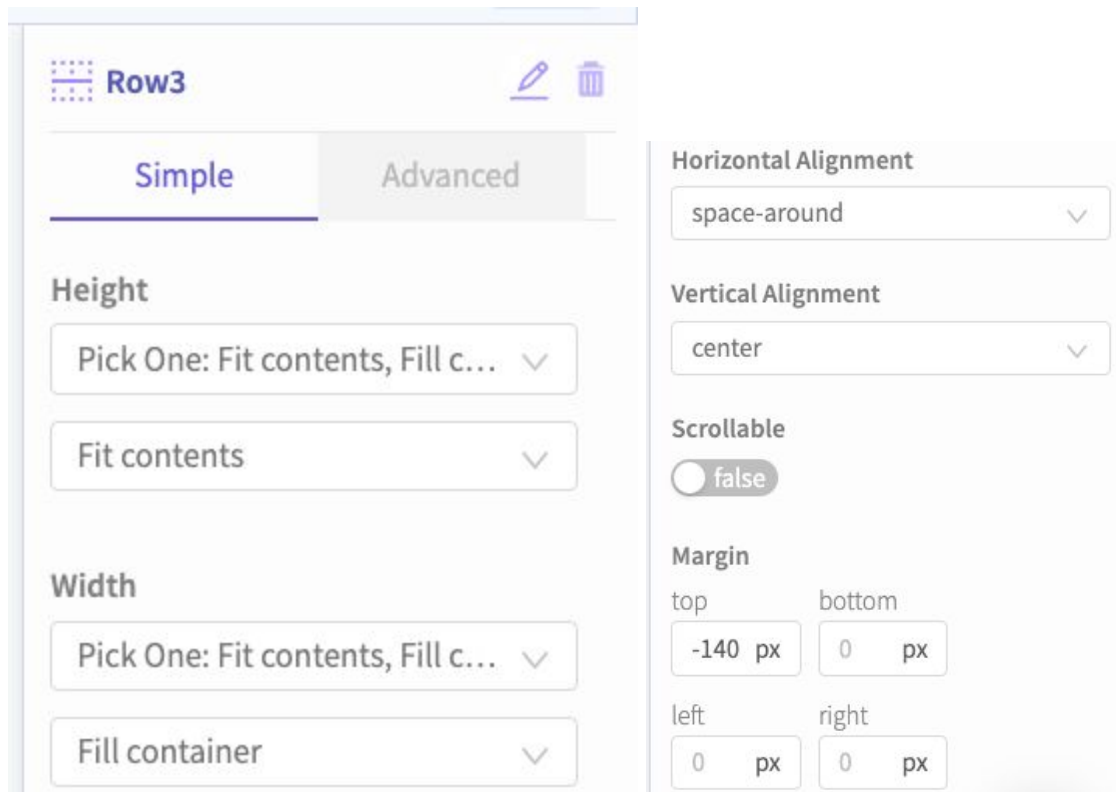
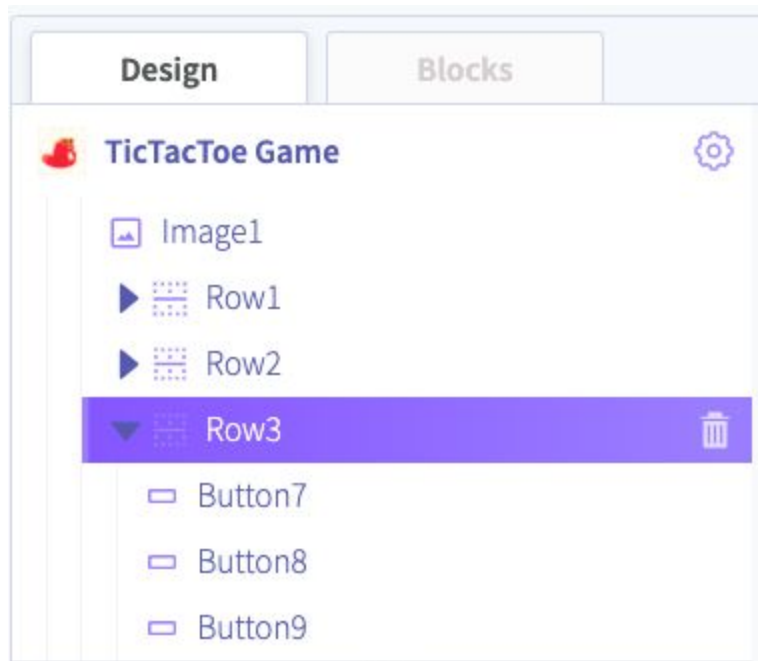
☐ false

Margin

top: -140 px bottom: 0 px

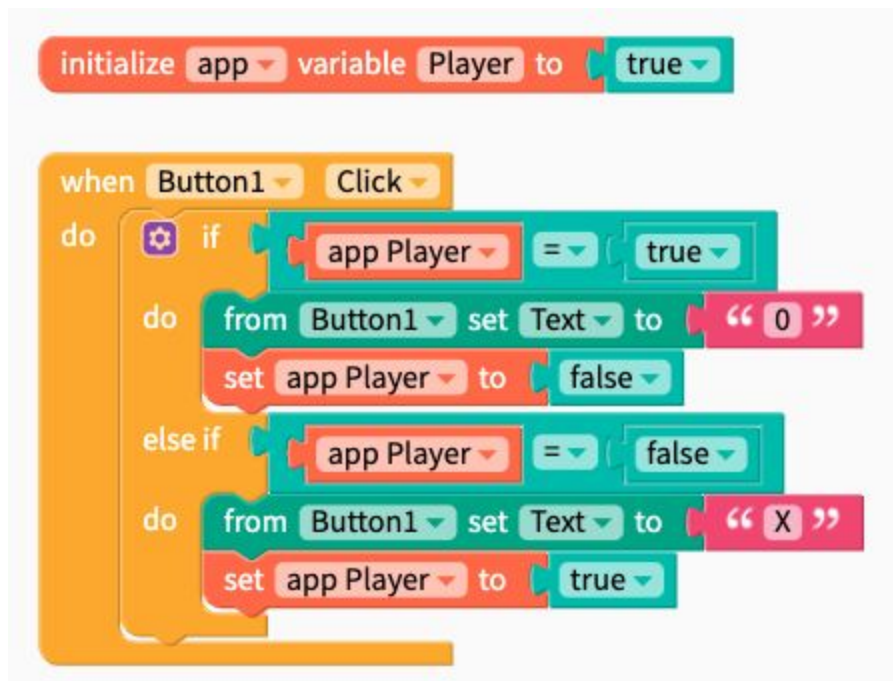
left: 0 px right: 0 px





The Rows are set with negative margins. This is because we need the buttons to be overlayed on the Tic-Tac-Toe Game Template Image.

CODE -



This game is played alternately by the two players.

We define a variable player and set it to true since this is a two player game.

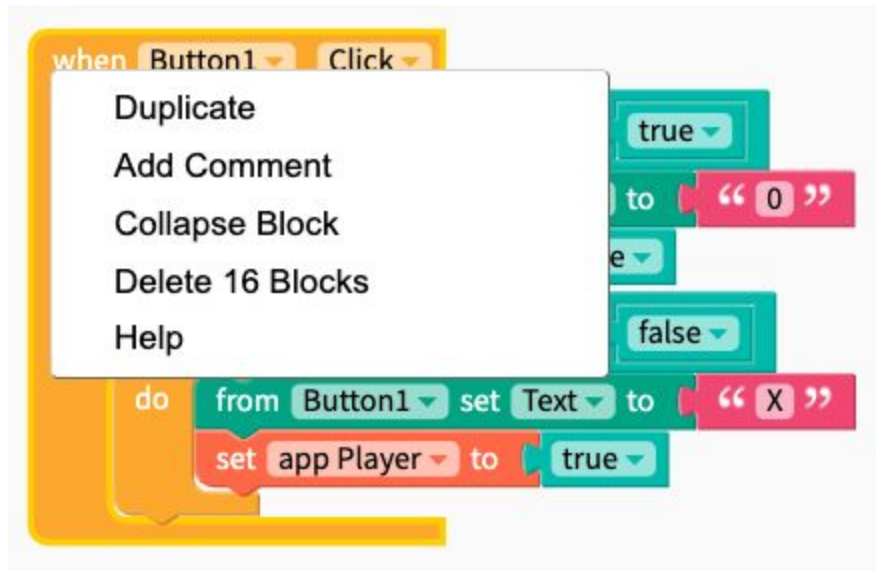
We need -

When a player clicks a button it should display 0 and

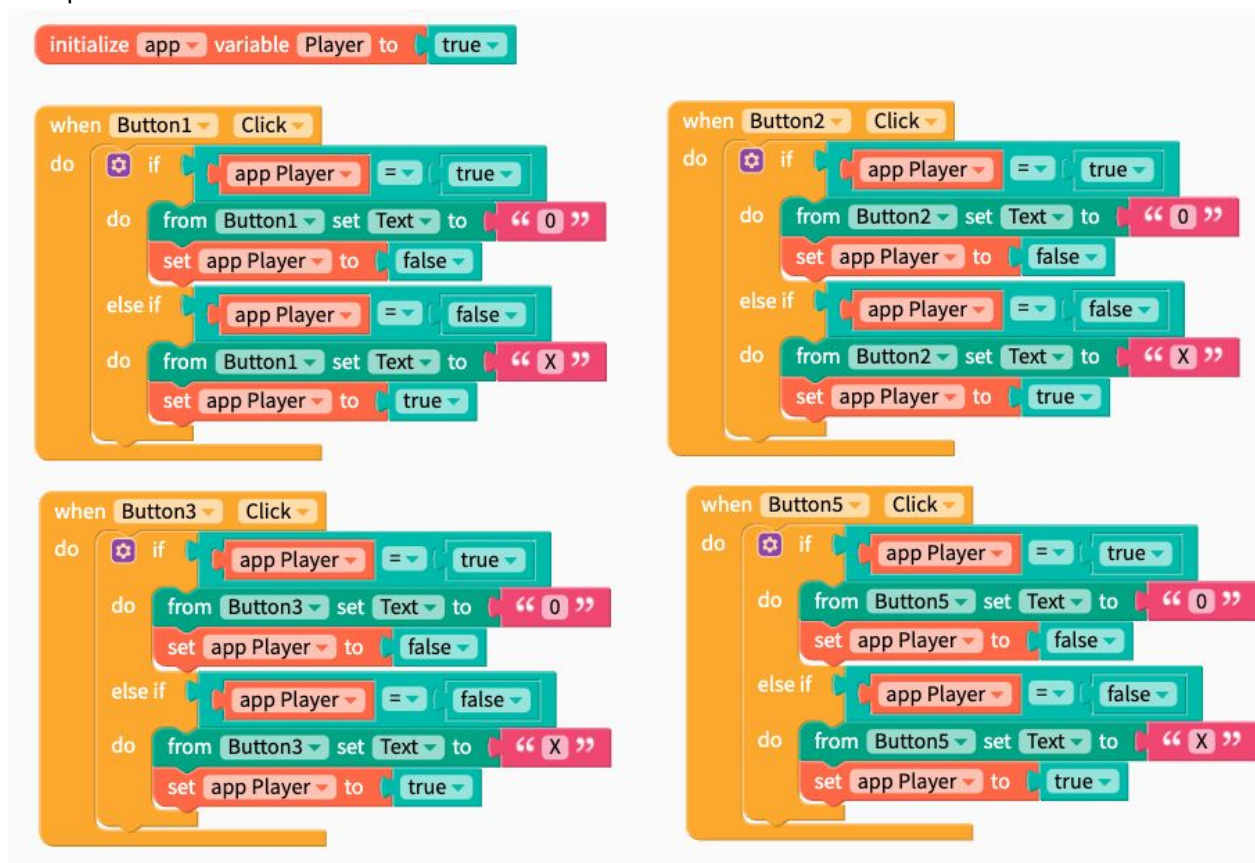
When the other player clicks another button it should display X.

For this every time a player is true, the button prints 0 and sets the player to false, so that when the next player clicks another button the app player is set to false by the first player so this time the button prints x and again the player variable is switched to true so that at next button press test displayed is 0.

All the 9 buttons should have same functionality so we can use the Duplicate Block feature by right clicking on button1 code block and changing the button number.



Complete code:



```

when Button4 Click
do
  if app Player = true
  do
    from Button4 set Text to "0"
    set app Player to false
  else if app Player = false
  do
    from Button4 set Text to "X"
    set app Player to true
  
```

```

when Button6 Click
do
  if app Player = true
  do
    from Button6 set Text to "0"
    set app Player to false
  else if app Player = false
  do
    from Button6 set Text to "X"
    set app Player to true
  
```

```

when Button7 Click
do
  if app Player = true
  do
    from Button7 set Text to "0"
    set app Player to false
  else if app Player = false
  do
    from Button7 set Text to "X"
    set app Player to true
  
```

```

when Button8 Click
do
  if app Player = true
  do
    from Button8 set Text to "0"
    set app Player to false
  else if app Player = false
  do
    from Button8 set Text to "X"
    set app Player to true
  
```

```

when Button9 Click
do
  if app Player = true
  do
    from Button9 set Text to "0"
    set app Player to false
  else if app Player = false
  do
    from Button9 set Text to "X"
    set app Player to true
  
```

Thus, you built a native app.

What's next? :

In the next class, you will be learning to code the basic conditions and rules of this game.