# Game Description

The objective is to direct a flying bird, which moves continuously to the left or right, between sets of pipes. If the bird touches the pipes, the game is over. The bird can be moved upward/downward using the up/down arrow keys. The player is scored based on the number of pipes that the bird successfully passes through.

## Specification:

A flying bird “appears” to fly continuously to the left although it actually only moves up and down using the arrow keys. A pair of vertical bars (or pipes) with a small gap between them appears at the left edge of the screen and glides to the right edge. More such pairs follow continuously at random intervals. Each pair of pipes is exactly aligned vertically. The gap in each pair appears at a random height. The objective of the game is to make the bird pass through the gap between each pair of pipes. If the bird touches any of the pipes, the bird dies and the game is over. The bird can be moved upward/downward using the up/down arrow keys. The player is scored based on the number of pipes that the bird successfully passes through.

## Levels of difficulty:

For each increasing level:

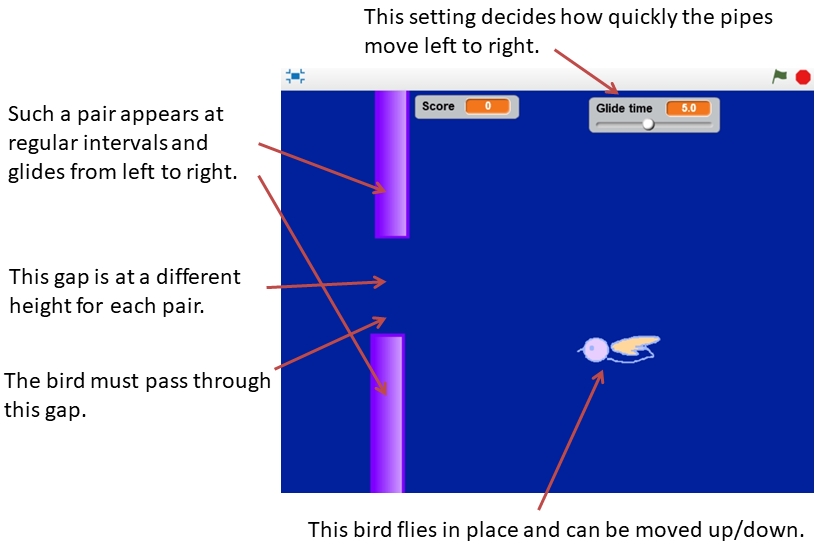
* the frequency of pipes increases
* the gap decreases
* their speed of motion increases

# Scratch and CS Concepts Used

* Costumes
* Random
* Conditionals
* Variables
* Broadcasting
* Operators
* Clones

# High Level Design

See the screen shot below:



### Checkpoint 1:

1. Get a flying bird facing west:
   * flapping wings using costumes
   * arrow keys to move up and down
2. Get just one vertical pipe (call it "upper"):
   * make it as tall as the screen
   * make it travel from the left edge to the right edge repeatedly, and use "score" variable to count repetitions
   * use “random” command to change its Y position for every travel

### Checkpoint 2:

1. Add the second pipe (make a duplicate of "upper"):
   * align it vertically with “upper” pipe
   * it should move exactly like “upper” (use broadcast to synchronize)
   * it should be positioned below “upper” with a small gap (make its Y less by 360+gap than Y of "upper")
2. Bird should sense touching both pipes.
   * if it touches any of the pipes, it should fall down
   * use "game over" broadcast to hide pipes

### Checkpoint 3:

1. Add a "help" screen and start game with "space bar pressed" event
2. Add "Glide time" slider variable to control pipe speed.

### Checkpoint 4: Use clones

1. “Upper" pipe:
   * parent will create clone every few seconds (at random)
   * use "topY" variable to save its Y position
   * signal “lower” pipe
   * start moving
2. “Lower" pipe:
   * parent will create a single clone when signaled by “upper”
   * clone will use "topY" to calculate its Y
   * start moving

## User Interface:

1. Flying bird: costumes for flying.
2. Two vertical rectangular pipes: vertically aligned, with some gap between.
3. Level slider variable: values 1 to 3.
4. Stage has no role.

## Behavior:

### Flying bird:

1. The bird will fly in place: using costume animation.
2. It will sense collision with pipes: will fall down and stop game.
3. It will allow up/down movement using up/down arrow keys.

### Pipes as sprites:

Pipes will repeatedly glide together from the left edge of the screen to the right:

- the top one will position itself at a random point near left edge, wait some, glide to right edge, and repeat.

- The bottom pipe will align itself with the top one and glide.

### Pipes as clones:

Both top and bottom pipes will be created as clones. For them to work in tandem, the top pipe needs to transmit its “y position” to the bottom pipe.

### Algorithm for the two pipes:

Given:

Two sprites - one for each pipe

Each of them is full-length.

The costume center of each at its center.

Variables:

YTOP – to save the Y position of the top pipe

GAP – the gap between the two pipes

Top pipe:

When program starts:

- Forever:

-- Create a clone

-- send message to bottom pipe

-- Wait a few seconds

When the pipe clone starts:

-- pick X of leftmost edge (-240), pick Y randomly (> 0 and < 360 to be visible)

-- save Y coordinate in variable (YTOP)

-- go to X, Y

-- glide towards the other end (X=240, no change in Y)

-- delete clone

Bottom pipe:

When message received:

-- create a clone

When top pipe clone starts:

-- X = -240, Y = YTOP – (360 + GAP)

-- go to X, Y

-- glide towards the other end (no change in Y)

-- delete clone

# Version 1

## Feature idea 1:

Get a flying bird:

- flapping wings (optional)

- arrow keys to move up and down

## Feature idea 2:

2. Get one vertical pipe (call it "upper"):

- top to bottom

- make it move left to right forever

- use random to change Y position

- "score" variable to count pipes

## Feature idea 3:

3. If bird touches the pipe, it should fall down and game should stop.

# Version 2:

## Feature idea 4:

Add the second pipe (duplicate of "upper"):

- use broadcast to start its motion

- its Y should always be >360 less than Y of "upper"

## Feature idea 5:

Bird should sense touching both pipes.

- use "game over" broadcast to hide pipes

# Version 3:

## Feature idea 6:

Add a "help" screen and start game with "space" event

## Feature idea 7:

Add "Glide time" slider variable to control pipe speed.

# Version 4:

## Feature idea 8:

Use clones for "upper" pipe:

- parent will create clone every few seconds (at random)

- clone will not wait

- use "topY" variable to save Y

## Feature idea 9:

Use clones for "lower" pipe:

- parent will create a single clone upon receiving "ready"

- clone will use "topY" to calculate its Y

# Version 5:

## Feature idea 10:

Include prizes between each pair of pipes.