

Draw and Transform:

First, creating some functions to draw different shapes in order to create all different components of the penguin.

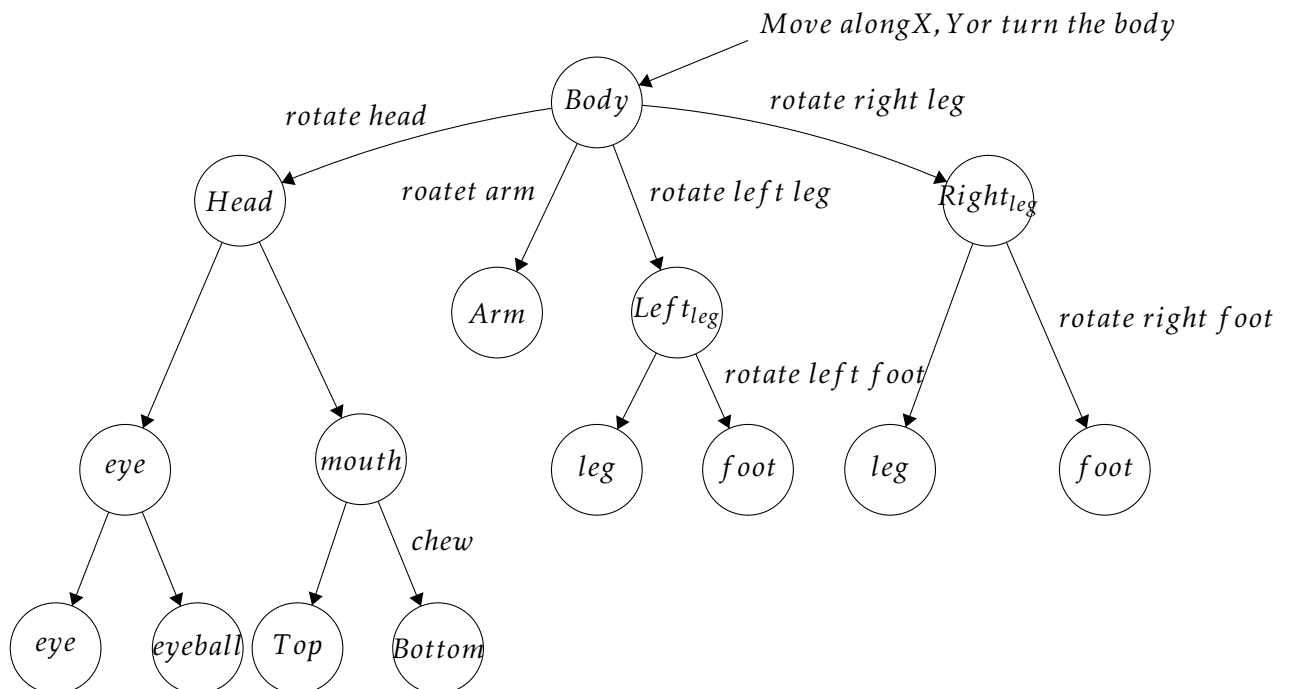
`drawCircle()`:

function takes a parameter radius r and draw a circle of radius r .

`drawPolygon()`:

function draw a polygon having vertices sets (X, Y) where X stands for x-coordinate and Y stands for y-coordinate. X is array contains all x-coordinates, Y is array contains all y-coordinates.

After done creating the shapes of all different components, let's start building the penguin. First draw the body in the middle of the window, then scale it. Then using translation takes us to local view relative to penguin's body then draw a component (i.e arm) give it some colour and scale it to a given size, then go back to world view(penguin's body) again and translation to another local view and draw another component. So every time we draw a component we always go to a local view relative penguin's body and after we done we go back to world view(penguin's body). Hence, when we moving penguin's body all other components are moving correspondingly. Especially, we need to notice that foot is also relatively to leg in the local view of leg, they are one component for world (penguin's body) but two separate parts if we go to the local view. This means when we moving leg, foot is moving correspondingly. And foot can also moving separately. Same for head, head also has subtrees, means when we moving head eye and mouth are also moving correspondingly. The structure tree below shows how does every part related to other parts and how does it move.



Every component is blocked by `glPushMatrix()` and `glPopMatrix()`.

Animate

The basic ideal of animate the penguin is using key frames. Following two functions will generally demonstrate how animation works.

Moving Arm:

```
const double arm_rot_speed = 0.1;
double arm_rot_t = (sin(animation_frame*arm_rot_speed) + 1.0) / 2.0;
arm_rot = arm_rot_t * ARM_MIN + (1 - arm_rot_t) * ARM_MAX;
```

Giving the rotation speed, this moving part using *sin* function(since it's periodic) to rotate the arm back and forth as the frame changes, within the given range specified by ARM_MIN and ARM_MAX.

Turning:

```
if(move_x < MOVE_X_MIN + 1.0 && check)
{
    check = false;
    turn = -1;
}
if(move_x > MOVE_X_MAX-1.0 && !check)
{
    check = true;
    turn = 1;
}
```

This piece of code checks if the penguin need to turn it's head. So basically if the penguin moves to very left mean $move_x < MOVE_X_MIN + 1.0$ or very right $move_x > MOVE_X_MAX - 1.0$. And *check* is a boolean variable which makes, the penguin only turn it's head once.

Then after combining all components together and moving them based on frame(i.e update x,y coordinate and rotation angle) the penguin is therefore moving. Moreover, turning it's head when it reached very left or right.