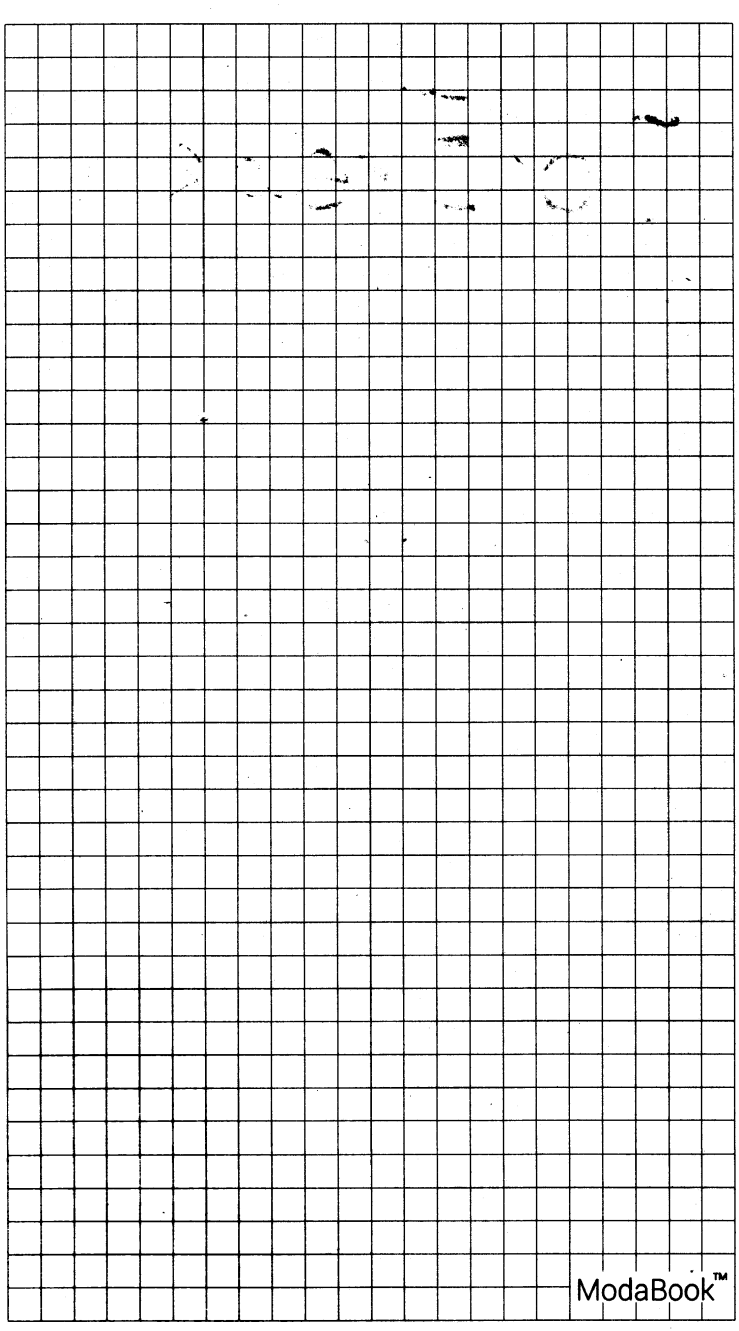
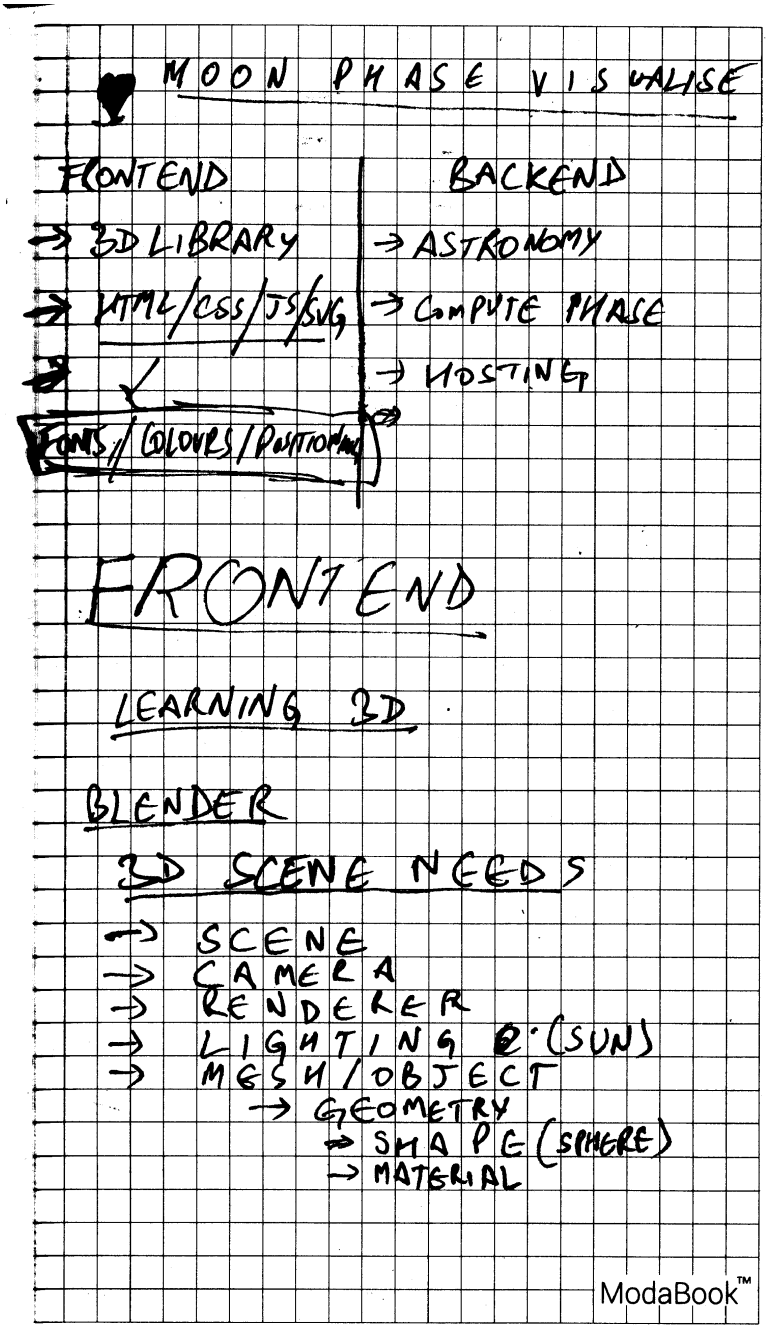
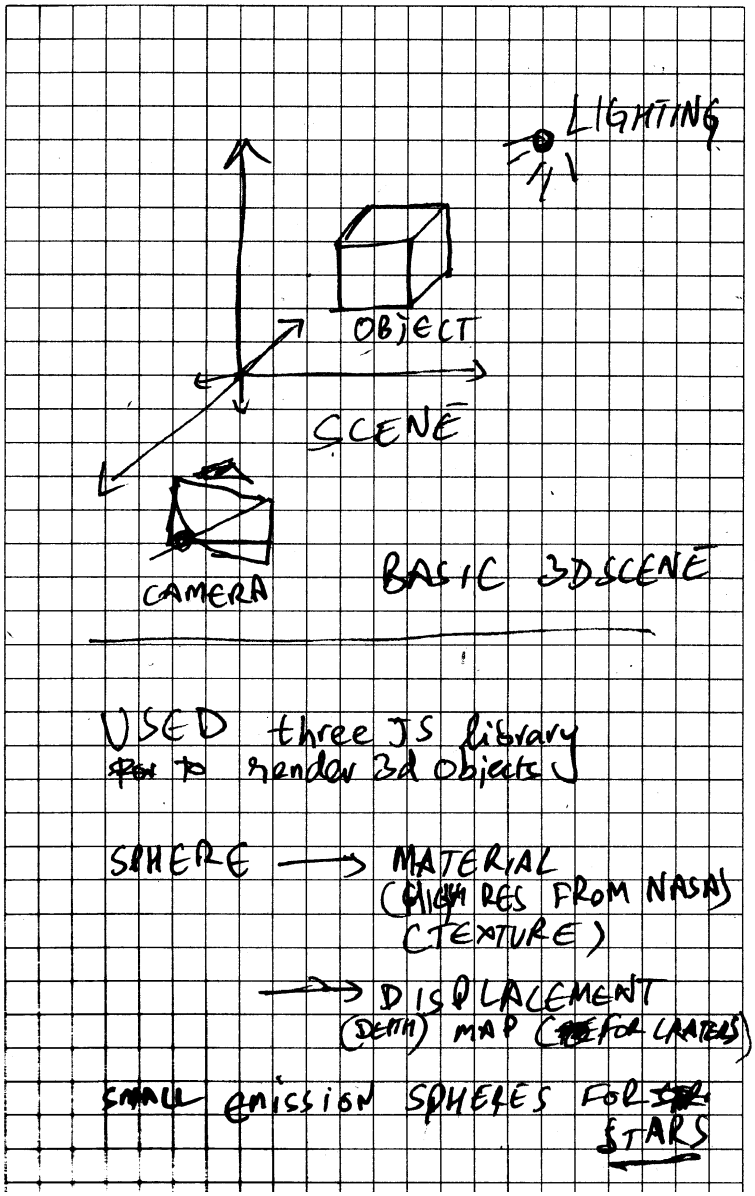
Moon Phase Documentation

Moon Phase Documentation

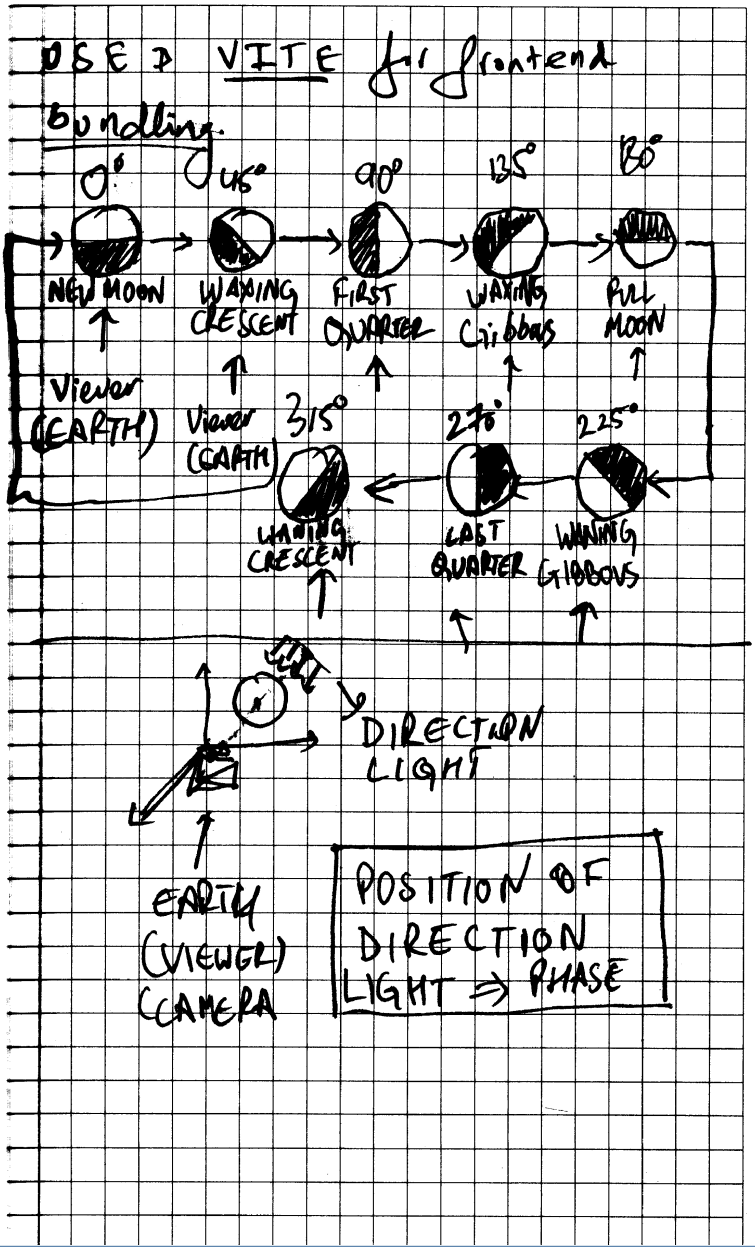




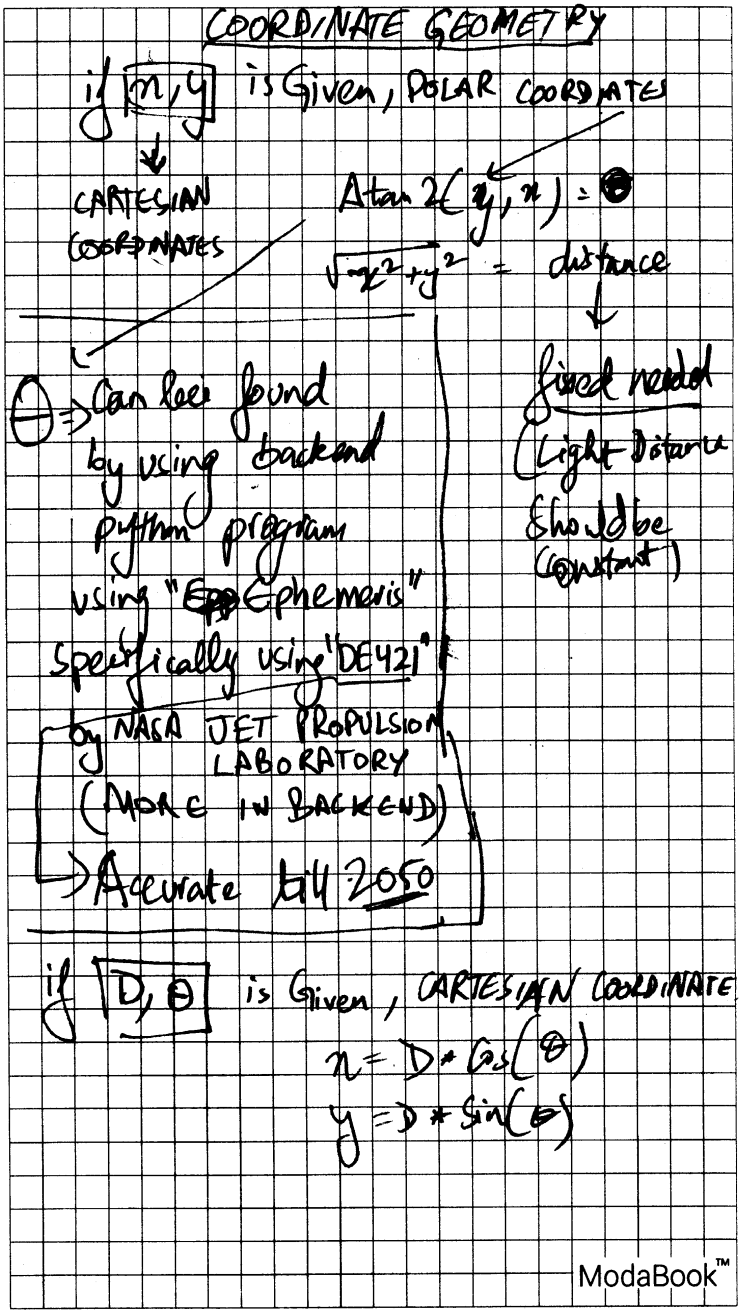
ModaBook™

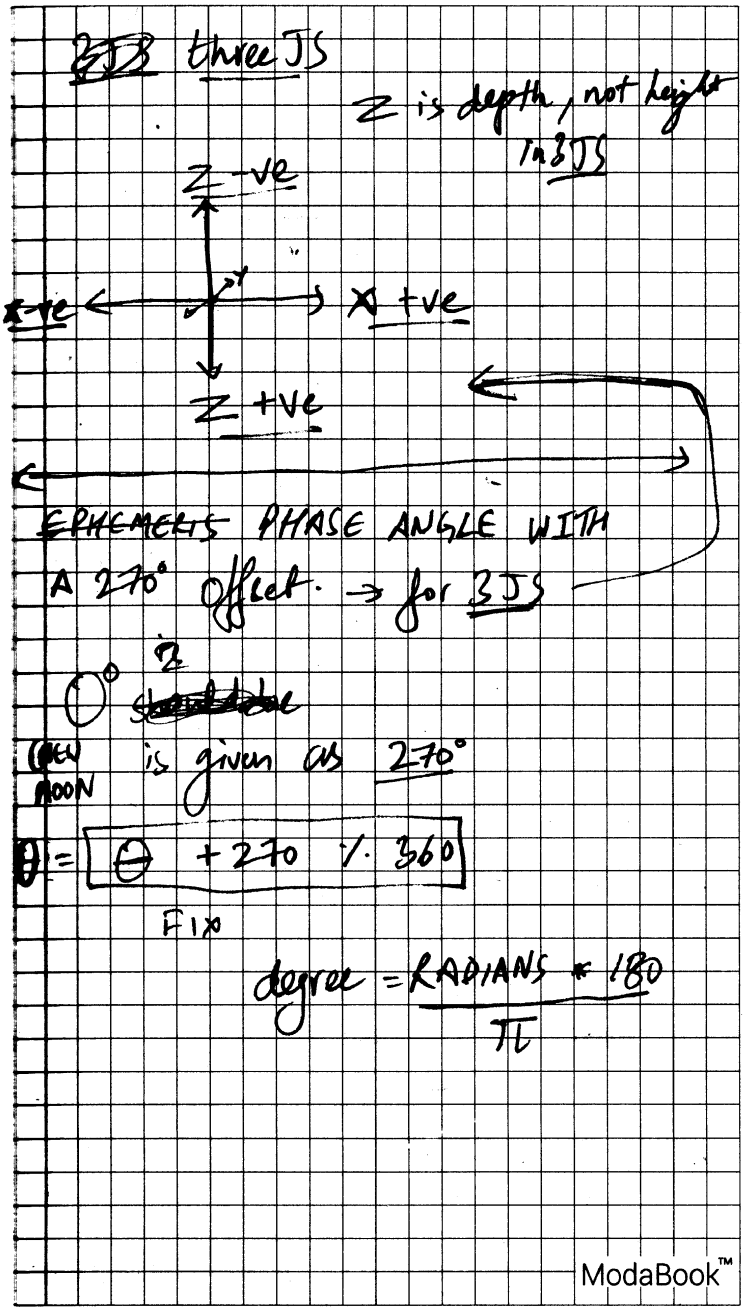



```
//Geometry and Material for Stars
const stargeo = new THREE.SphereGeometry(0.25, 24, 24);
const starmat = new THREE.MeshStandardMaterial({
  color: 0xffffff,
  emissive: 0xf0ff99,
});
function addStar() {
  const star = new THREE.Mesh(stargeo, starmat);
  const [x, y, z] = Array(3)
    .fill()
    .map(() => THREE.MathUtils.randFloatSpread(200));
  star.position.set(x, y, z);
  scene.add(star);
}
Array(200).fill().forEach(addStar);
```

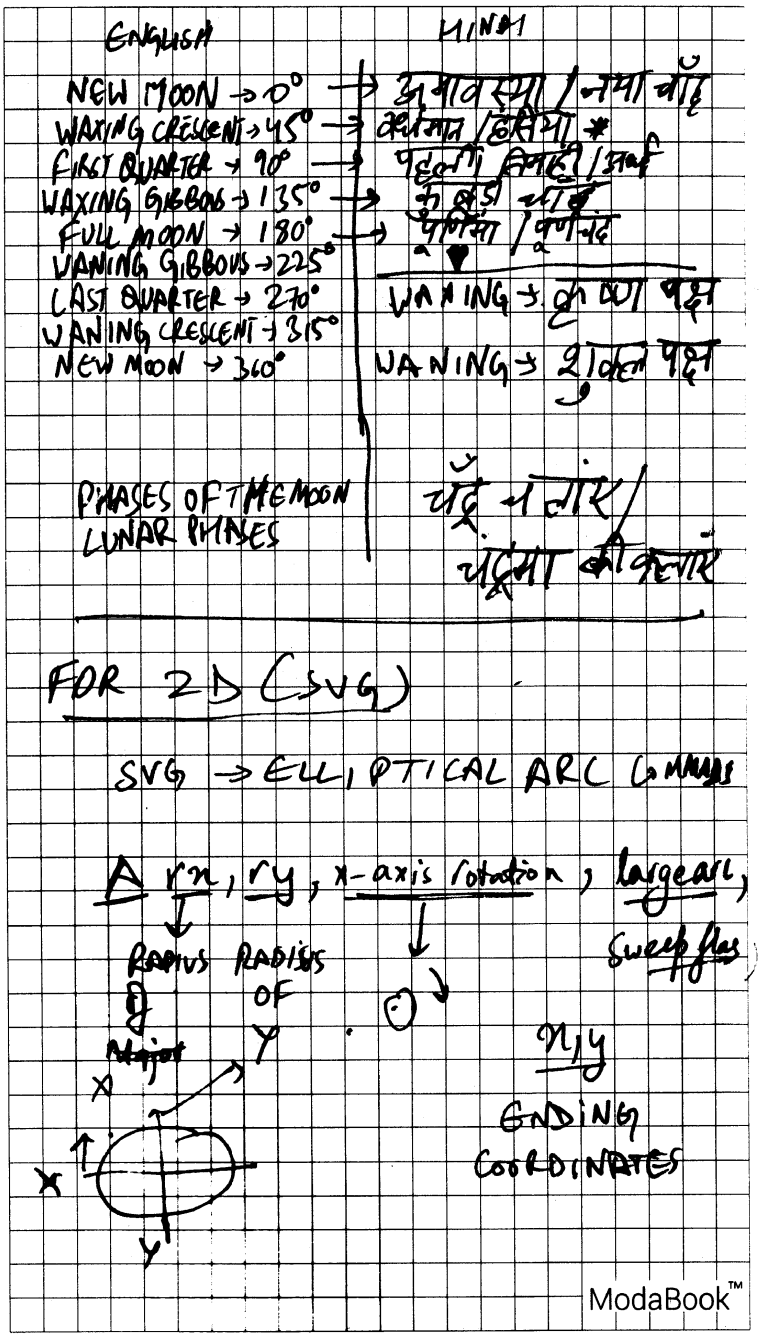


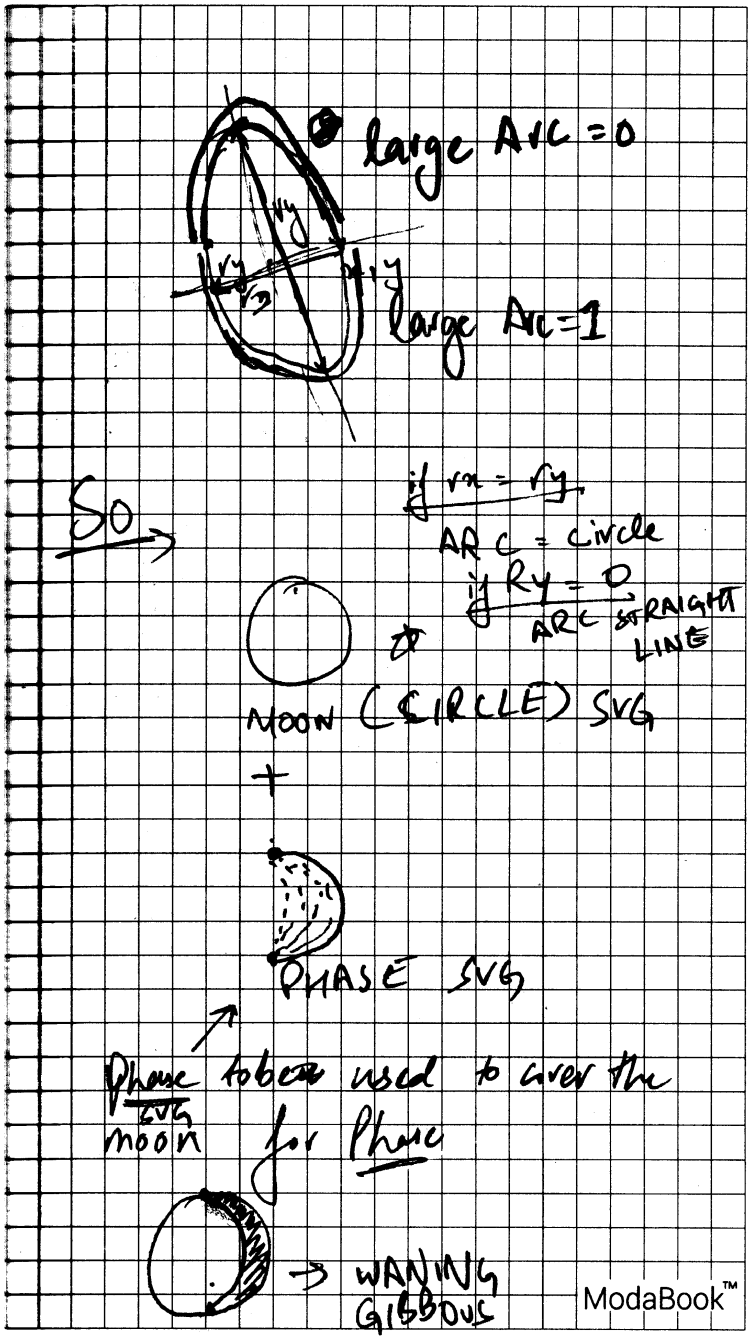

```
// Moon
const moonTexture = new THREE.TextureLoader().load(
  "./assets/lroc_color_poles_2k.png"
);
const displacementTexture = new THREE.TextureLoader().load(
  "./assets/lroc_color_poles_2k.png"
);
const normalTexture = new THREE.TextureLoader().load("assets/NormalMap.png");

const moon = new THREE.Mesh(
  new THREE.SphereGeometry(20, 64, 64),
  new THREE.MeshStandardMaterial({
    map: moonTexture,
    displacementMap: displacementTexture,
    normalMap: normalTexture,
  })
);
```



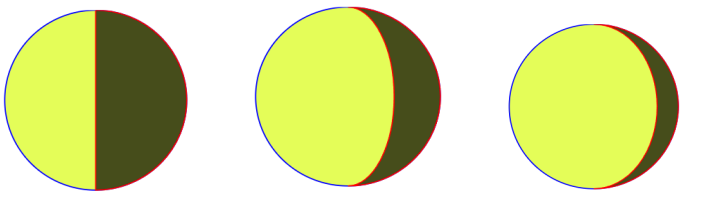



```
function pol2rect(r, θ) {
  return [r * Math.cos(θ), r * Math.sin(θ)];
}
const sunangle = (angle + 270) % 360;
const radians = (sunangle * Math.PI) / 180;
const xz = pol2rect(80, radians);
directionalLight.target = moon;
directionalLight.position.set(xz[0], 0, xz[1]);
scene.add(moon);
```






```svg
<?xml version="1.0" standalone="no"?>
<svg width="12cm" height="5.25cm" viewBox="0 0 1200 400" xmlns="http://www.w3.org/2000/svg" version="1.1">
  <title>Example arcs01 - arc commands in path data</title>

  <circle cx="100" cy="75" r="75" fill="#e4fd58" stroke="blue"
  stroke-width="1" />
  <path d="m100,0 a0,20 0 1,1 0,150 a20,20 0 0,0 0,-150"
  fill="rgba(0,0,0,0.695)" stroke="red" stroke-width="1" />

</svg>
```



```javascript
const moon_phases_en = [
  "New Moon",
  "Waxing Crescent",
  "First Quarter",
  "Waxing Gibbous",
  "Full Moon",
  "Waning Gibbous",
  "Last Quarter",
  "Waning Crescent",
];
const moon_phases_hi = [
  "नया चाँद/अमावस्या",
  "शुक्ल पक्ष हंसिया चांद/वर्धमान चांद",
  "पहली तिमाही/अर्द्ध चंद्र",
  "शुक्ल पक्ष कुबड़ा चांद",
  "पूर्णचंद/पूर्णिमा 💗",
  "कृष्ण पक्ष कुबड़ा चांद",
  "अंतिम तिमाही/अर्द्ध चंद्र",
  "कृष्ण पक्ष हंसिया चांद/वर्धमान चांद",
];
```

BACKEND

SKYFIELD → PYTHON MODULE

FLASK → WEB FRAMEWORK

DE421 → PLANETARY AND LUNAR EPHEMERIS
VALID ~~from~~ 1900 - 2050
↳ A TABLE ~~OF~~ CONTAINING PLANETARY POSITIONS

NORMALISE PHASE ANGLE
TO ↳ MOON_PHASE_NAME
NAME

MOON PHASES =
["NEW MOON" "WAXING CRESCENT",
"FIRST QUARTER" "WAXING GIBBOUS",
"FULL MOON" "WANING GIBBOUS",
"LAST QUARTER", "WANINGCRESCENT"]

DEGREES =
[0, 45, 90, 135, 180, 225, 270, 315]

Index = round(Phase ANGLE / 45) % 8

HOSTING

→ VPS → Virtual Private server
(FROM LINODE) AWS LIGHTSAIL, DIGITAL OCEAN etc)
(5 $/Month)

→ ~~NAME~~ DOMAIN → ~~10~~ 1$/Month
(Google domains, Go daddy) etc

VPS setup

NGINX → for reverse proxy

GUNICORN → SERVER GATEWAY

FLASK → FRAMEWORK
↳ JINJA2 → TEMPLATE ENGINE TO SEND PHASE ANGLE FRONT END

FLASK APP (PYTHON CODE (SKYFIELD)) → GUNICORN SERVER → NGINX SERVER (REVERSE PROXY) → HTTP REQUESTS


```python
from skyfield.api import load
from skyfield.framelib import ecliptic_frame
def phase():
    ts = load.timescale()
    t = ts.now()
    # t = ts.utc(2023, 7, 3, 11, 39,47)
    # year, month, day,hour,minute,second

    eph = load('de421.bsp')
    sun, moon, earth = eph['sun'], eph['moon'], eph['earth']

    e = earth.at(t)
    s = e.observe(sun).apparent()
    m = e.observe(moon).apparent()

    _, slon, _ = s.frame_latlon(ecliptic_frame)
    _, mlon, _ = m.frame_latlon(ecliptic_frame)
    phase = (mlon.degrees - slon.degrees) % 360.0

    percent = 100.0 * m.fraction_illuminated(sun)

    # print('Phase (0°–360°): {0:.1f}'.format(phase))
    return phase
# print('Percent illuminated: {0:.1f}%'.format(percent))
```

```python
def moon_phase(degree):
    moon_phases = [
        "New Moon",
        "Waxing Crescent",
        "First Quarter",
        "Waxing Gibbous",
        "Full Moon",
        "Waning Gibbous",
        "Last Quarter",
        "Waning Crescent",
    ]
    degrees = [0, 45, 90, 135, 180, 225, 270, 315]
    print(len(moon_phases))
    i = round(degree / 45) % len(degrees)
    moon_phase = moon_phases[i]

    return moon_phase
```



```python
from flask import Flask, render_template
from moon import phase

app = Flask(__name__)


@app.route("/moon")
def hello_world():
    angle = str(phase())

    return render_template("moon.html", angle=angle)


# main driver function
if __name__ == "__main__":
    app.run(host="0.0.0.0")
```

QR Code

Generated for CHAITANYAMALHOTRA.COM/moon

QR code beautified using →

→ STABLE DIFFUSION
→ CONTROLNET
→ PHOTOSHOP

WEB UI FOR STABLE DIFFUSION

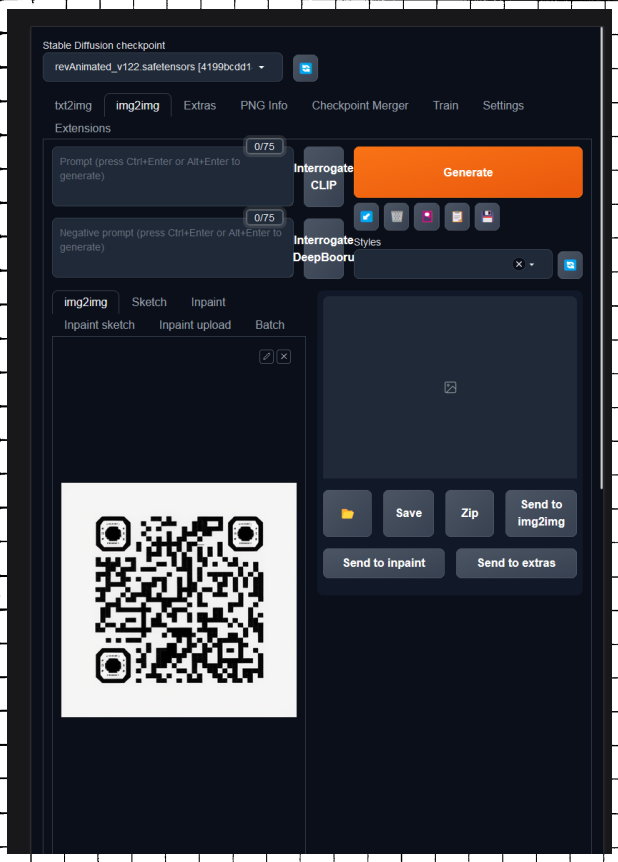
↳ AUTOMATIC 1111
↳ STABILITY MATRIX

CONTROLNET SETTINGS

↳ CONTROL NET MODEL ⇒ SD 15 Tile
→ PIXEL PERFECT
→ PREPROCESSOR ⇒ TILE_RESAMPLE

STABLE DIFFUSION XL for OUTPAINT

SPACE ART



Three.js

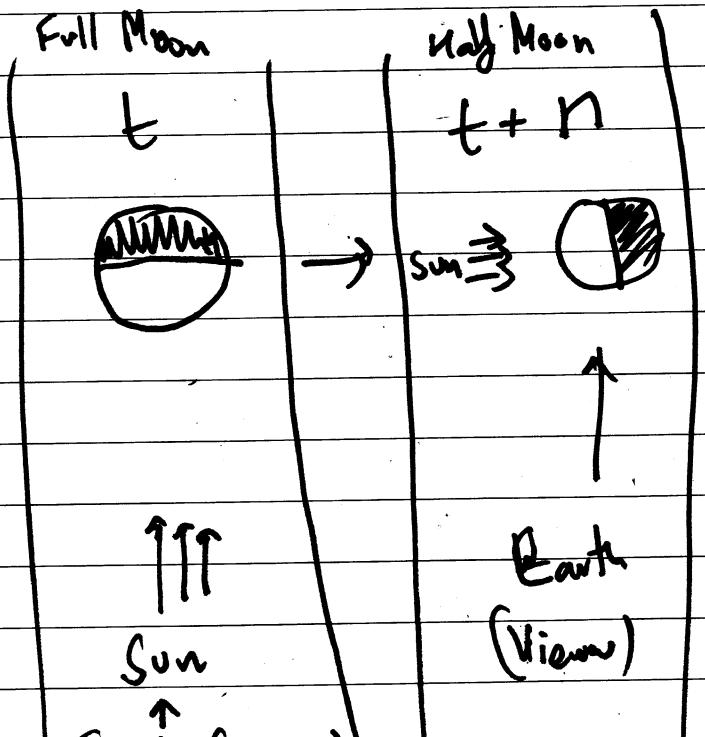
3 Things needed for every canvas

→ Scene
→ Camera
→ Renderer

Mesh → Geometry (Shape) eg. Sphere, cube etc
→ Material → color, affected by lighting

Lighting → Ambient Light
→

Moon



Given (x, y) Find → Polar Coordinates ~ (θ & distance)

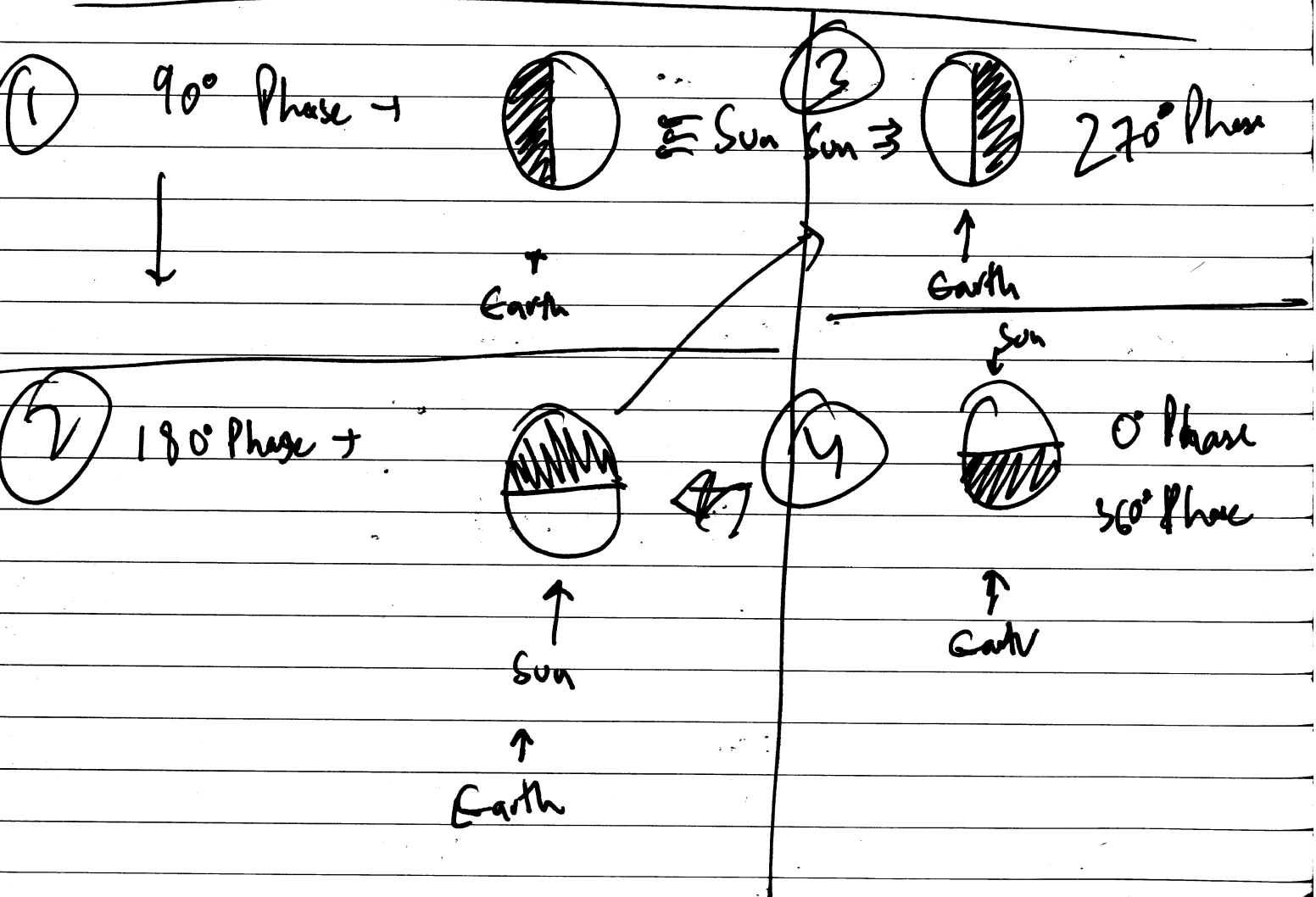
↳ Cartesian : $\theta = \text{Atan2}(y, x) = \text{Radians}$

$\text{distance} = \sqrt{x^2 + y^2}$

Given θ, distance Find → Cartesian Coordinates ~ (x, y)

↳ Polar coordinate : $x = D * \cos(\theta)$

$y = D * \sin(\theta)$

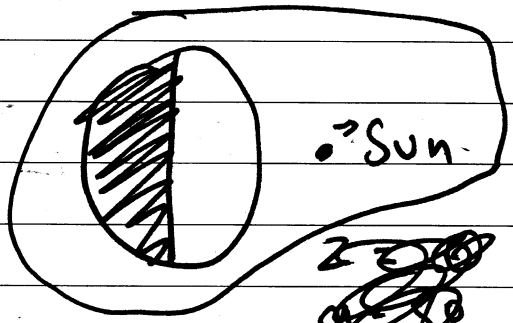


D=80

$\cos 0° = 1$
$\sin 0° = 0$

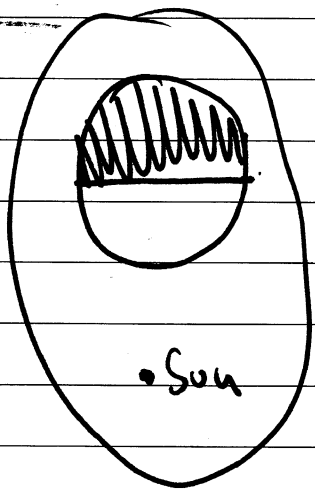
0° Phase
360° Phase

$x = 80$
$z = -80$



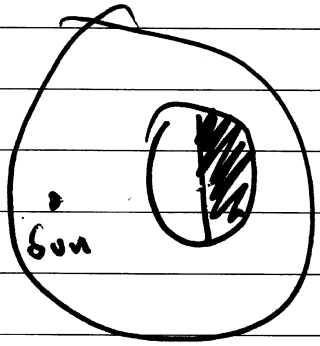
90° Phase

$x = 80$
$z = 0$



180° Phase

$x = 0$
$z = 80$



270° Phase

$x = -80$
$z = 0$

$y = 3d = 0$

$x = D * \cos(\theta)$
$y = D * \sin(\theta)$

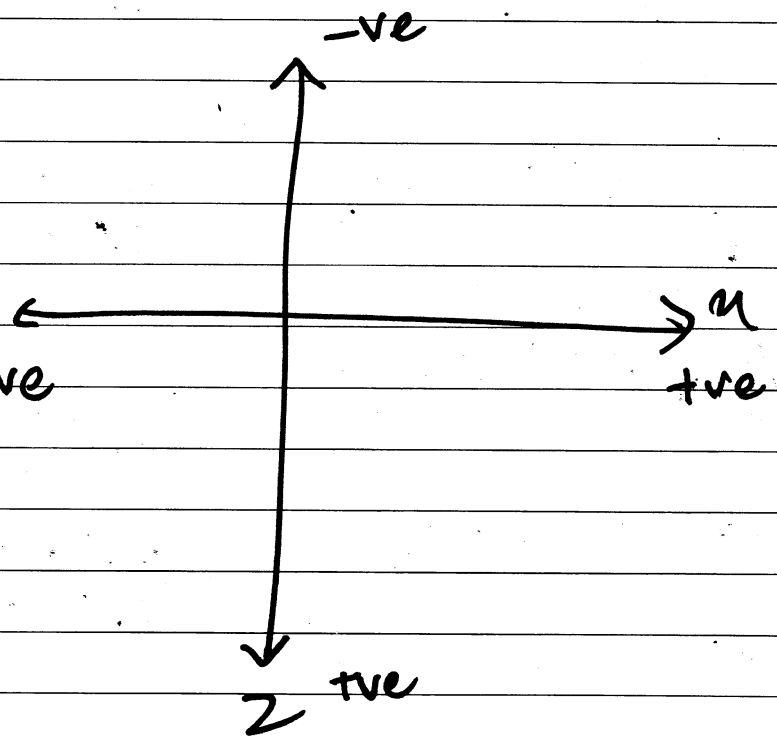
$\cos 0° = 1$
$\sin 0° = 0$

$\cos 90° = 0$
$\sin 90° = 1$

$\cos 180 = -1$
$\sin 180 = 0$

$\cos(270) = 0$
$\sin(270) = -1$

$\cos(360) = 1$
$\sin(360) = 0$

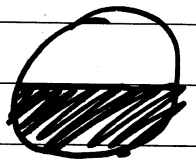


Conversion

(+ve) z

$+270^\circ$ (i) $\%360$

0°



$n = 0$

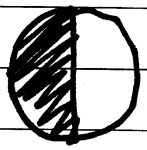
$z = -1$

$\cos 0 = 1$

$\sin 0 = 0$

$(0 + 270) \% 360 = 270^\circ$

90°



$n = 1$

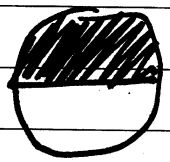
$z = 0$

$\cos 90 = 0$

$\sin 90 = 1$

$(90 + 270) \% 360 = 0^\circ$

180°



$n = 0$

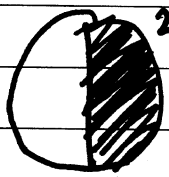
$z = 1$

$\cos 180 = -1$

$\sin 180 = 0$

$(180 + 270) \% 360 = 90^\circ$

270°



$n = -1$

$z = 0$

$\cos 270 = 0$

$\sin 270 = -1$

$(270 + 270) \% 360 = 180^\circ$

Post Conversion

$(\theta + 270^\circ) \% 360$

$\cos 270 = 0$

$\sin 270 = -1$

$0^\circ \rightarrow 270^\circ$

$90^\circ \rightarrow 0^\circ$

$\cos 0 = 1$

$\sin 0 = 0$

$180^\circ \rightarrow 90^\circ$

$\cos 90 = 0$

$\sin 90 = 1$

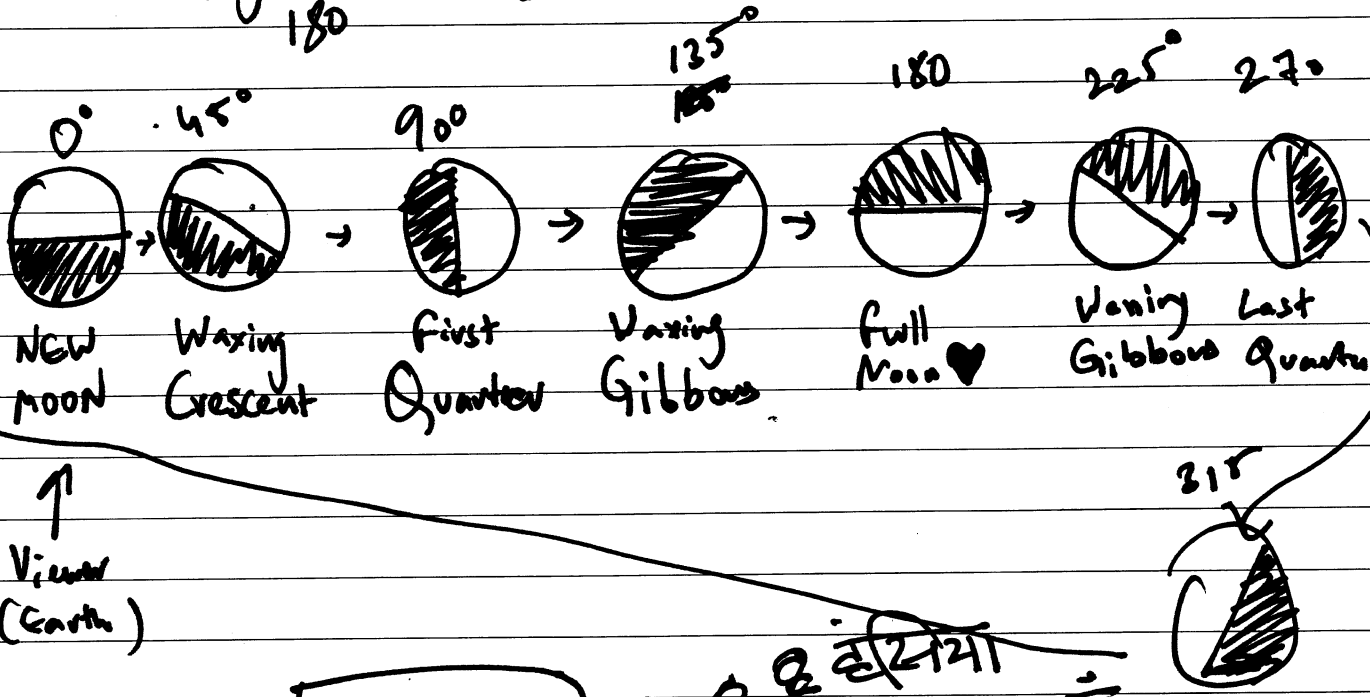
$270^\circ \rightarrow 180^\circ$

$\cos 180^\circ = -1$

$\sin 180^\circ = 0$

degree = radian × $\frac{180}{\pi}$

$\frac{\text{degree} \times \pi}{180}$ = radian



± 22.5

NEW MOON → 0

Waxing Crescent → 45°

First Quarter → 90°

Waxing Gibbons → 135°

Full Moon → 180°

Waning Gibbons → 225°

Last Quarter → 270°

Waning Crescent → 315° → 292.5 / 337.5] 45° Δ

NEW MOON → 360

अमावस्या → नया चाँद

वर्धमान चाँद

अर्ध चाँद → तिमाही

कुबड़ा चाँद

पूर्णिमा → पूर्णचंद

Waning Crescent

Waning → कृष्ण पक्ष

Waxing → शुक्ल पक्ष

Phases of the moon → ~~चंद्र~~ चंद्र कलाएं / चंद्रमा की कलाएं