5, 10, duck and cover pin logic

```
int int ball = 1
int int frame = 1
int int score = 0
int int strikescore1 = 0
int int strikescore2 = 0
```

1, 2, 3

```
int frame score = 0
int score = 0
int strike position = 0
int frame count = 0
int frames = 10
int ball limit = 3 (2 if 10 pin)
int random number = 1;
```

```
if int ball > int ball limit;
    int frame += 1
    int ball = 1
    int frame score = 0
    int ball score1 = 10;
    int frame += 1 (is if spin)
    int strike position = 1
```

respawn pins

// strike logic

```
if int strike position = 1;
    int score +=
    int strike position = 0
```

```
int strike score1 = int score
int frame count = int frame
if int frame > int frame count + frame
int strike score2 = int score
(int strike score1 * 2) + (int strike score2 * 2)
```

frame

```
if int ball score2 = 10 - int ball score1;
    int frame += 1 (is if spin)
    int strike position = 2
```

// spare logic

```
if int strike position = 2;
    int strike score1 = int frame score
    int score +=
    int strike position = 0
```

int strike score * 2

if ball collides with pins in 10 and duck pin
if random number = 100
remove pins and count score for each pin removed
int score += 1
pin gets Deleted
if ball collides with gutter
int ball += 1
if frame = 10 and ball score 1 = 10:
int frames += 2
int ball limit = 1
if frame = 10 and ball score 2 = 10 - ball score 1:
int frames += 1
int ball limit = 1
if frames > int frames:
back to menu
reset variables
if ball collides with 5, 3 or 2 pin in spin
score += 5, 3 or 2 depending on what pin is hit
delete pin

logic for german pin

```
int int ball = 1
int int frame = 1
int int score = 0
int frames = 24
int ball limit = 1
int random number = I

if int ball > ball limit
    int frame t = 1
    int ball = 1
    reset Pins

if ball collider with a pin
    int score t = 1

if ball collides with gutter
    int ball t = 1

if frames > int frames
   back to menu
   reset variables
```

if int random number = 100
remove all pins and score those removed pins

logic for texas q pin

int int ball = 1
int score = 0
int int frame = 1

int white pins 1-8 = 0
int Red Pin = 0
int frames = 6
int ball limit = 2
int random number = 1

if int ball > ball limit
int frame += 1
int ball = 1
reset pins and pin variables

if ball collided with gutter
int ball += 1

if white pins 1-8 = 1 and Red Pin = 0
score += 12
reset white pins and white pin variables

if white pins 1-8 = 1 and red pin = 1
score += 9
reset pins and pin variables

if ball collides with pin
if random number = 100
remove all pins except red
else if random number = 95-99
remove all pins

int
if frame = 6 and int ball = 2
collide with pins = 1 point to score
remove each pin

if ball collides with white pins 1-8 and red pin
each integer for each corresponding pin becomes a 1
removes each pin

if int frame > frames
back to menu
reset variables

Controls

```
power meter = 1; 2nd powermeter = 1;
int gutter = 0;
if mode = 0
  while a button is not pressed
    powermeter += 1;
    if power meter > 10;
      power meter = 1;
  when a button is pressed
    launch ball at powermeter level
  if int gutter = 1
    go back to loop
    int gutter = 0;
if mode = 1
  Mouse-y = power meter += 1;
  Mouse+y = Powermeter -= 1;        seconds can be fine tuned
  wait 2 seconds
  2nd power meter = power meter
  if power meter = 0
    launch ball at 2nd powermeter level
  if power meter > 10
    powermeter = 10
```

movement is left and right for both rotate and around the angle