

Lab 3

Apr. 6/7

Lab 3 — Go

```
vector<Point> points;  
for(int i =1; i < 80; ++i){  
    for(int j=1; j < 40; ++j){  
        points.push_back({i,j});  
    }  
}
```

No dcurses yet
add in step 13

//randomly shuffle — algorithm library

Steps 1-2
check

Lab 3 — Go

```
for(Point p: points){  
    cout << p.x << ", " << p.y << endl;  
}
```

Steps 3-5
check

```
for(Point p: points){  
    double c = noice.getColor(p.x,p.y,0.1,1);  
    int color = (23*c)+232;  
    cout << p.x << ", " << p.y;  
    cout << "->" << color<< endl;  
}
```

Lab 3 — Go

getTimeMilli
a function to write!
look up clock_gettime

```
int main(){  
    double startTime = getTimeMilli();
```

Magic Stuff — steps 1-5

```
    double delta = getTimeMilli()-startTime;  
    cout << delta << endl;  
    return 0;  
}
```

Steps 6-7
check kind of

Lab 3 — Go

```
int main(int argc, char** argv){  
    noice.seed = atoi(argv[1]);  
    double startTime = getTimeMilli();
```

Magic Stuff — steps 1-5

```
    double delta = getTimeMilli()-startTime;  
    cout << delta << endl;  
    return 0;  
}
```

Step 8
check

Lab 3 — Go

```
int main(int argc, char** argv){  
    noice.seed = atoi(argv[1]);  
    int threads = atoi(argv[2]);  
    double startTime = getTimeMilli();
```

Magic Stuff — steps 1-5

```
    double delta = getTimeMilli()-startTime;  
    cout << delta << endl;  
    return 0;  
}
```

Step 11
check

Lab 3 — Thread

```
void run(){
  while(points.size() > 0){
    Point p = points[points.size()-1];
    points.pop_back();
    double c = noice.getColor(p.x,p.y,0.1,1);
    int color = (23*c)+232;
    cout << p.x << ", " << p.y;
    cout << "->" << color<< endl;
  }
}
```

Mutex?

what is shared across threads
what needs to be protected
by a mutex?

Step 9-10

check — need to create threads and join

Cheers