PSAM 5600 Section E CRN 4598
Spring Semester 2017
Wednesdays 7:00PM – 9:40PM

Currents: Coding with Cinder

Week 3: Object Oriented Programming / Particle System

Instructors
Luobin Wang (luobin@newschool.edu)
Weili Shi (weili@newschool.edu)

Bear with us!

Let’s work together to beat this Cinder thing!

```
class Buddha {
...
};

Buddha shakyamuni;
```

Object Oriented Programming

In **OOP**, computer programs are designed by making them out of objects that interact with one another.

http://www.wikiwand.com/en/Object-oriented_programming

```
Buddha shakyamuni;
shakyamuni.meditate(); // I don’t know how Budda meditates but I just want him to.
```

Data Encapsulation

Bundling the data, and the functions that use them

Abstraction

Exposing only the interfaces, and hiding the implementation details from the user.

https://www.tutorialspoint.com/cplusplus/cpp_data_encapsulation.htm

```
Buddha shakyamuni(42); // Constructing a buddha with the ultimate answer...
```

Constructors

A constructor is a special member function of a class that is executed whenever we create new objects of that class.

https://www.tutorialspoint.com/cplusplus/cpp_constructor_destructor.htm

```
class Buddha : public Monk {
...
};
```

Inheritance

Defining a class based on another class.

An opportunity to reuse the code functionality and fast implementation time.

https://www.tutorialspoint.com/cplusplus/cpp_inheritance.htm

Access	public	protected	private
Same class	yes	yes	yes
Derived classes	yes	yes	no
Outside classes	yes	no	no

Acces Control and Inheritance

https://www.tutorialspoint.com/cplusplus/cpp_inheritance.htm

Polymorphism

A call to a member function will cause a different function to be executed depending on the type of object that invokes the function.

https://www.tutorialspoint.com/cplusplus/cpp_inheritance.htm

ci::app::setFrameRate()

set your application frame rate, potentially making it more stable.

gl::drawString()

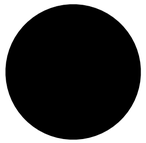
Draw a string, easiest way to put text on. But you can’t set font.

Particles!

Not one particle.

But many of them!

Particle



If you build a particle, what does it need

Velocity velocity = velocity + acceleration

Acceleration acceleration += force

Force force is usually a vec2 (it needs direction)

Friction velocity *= friction

force = m * a

if m == 1 then force = a

Considering our particles are all of equal mass

speed = speed + force

Apply force to speed every frame

force *= 0

Force cannot be add up. so don’t forget to reset it every frame

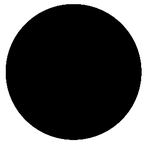
speed *= friction

We need friction, or particle will never stop

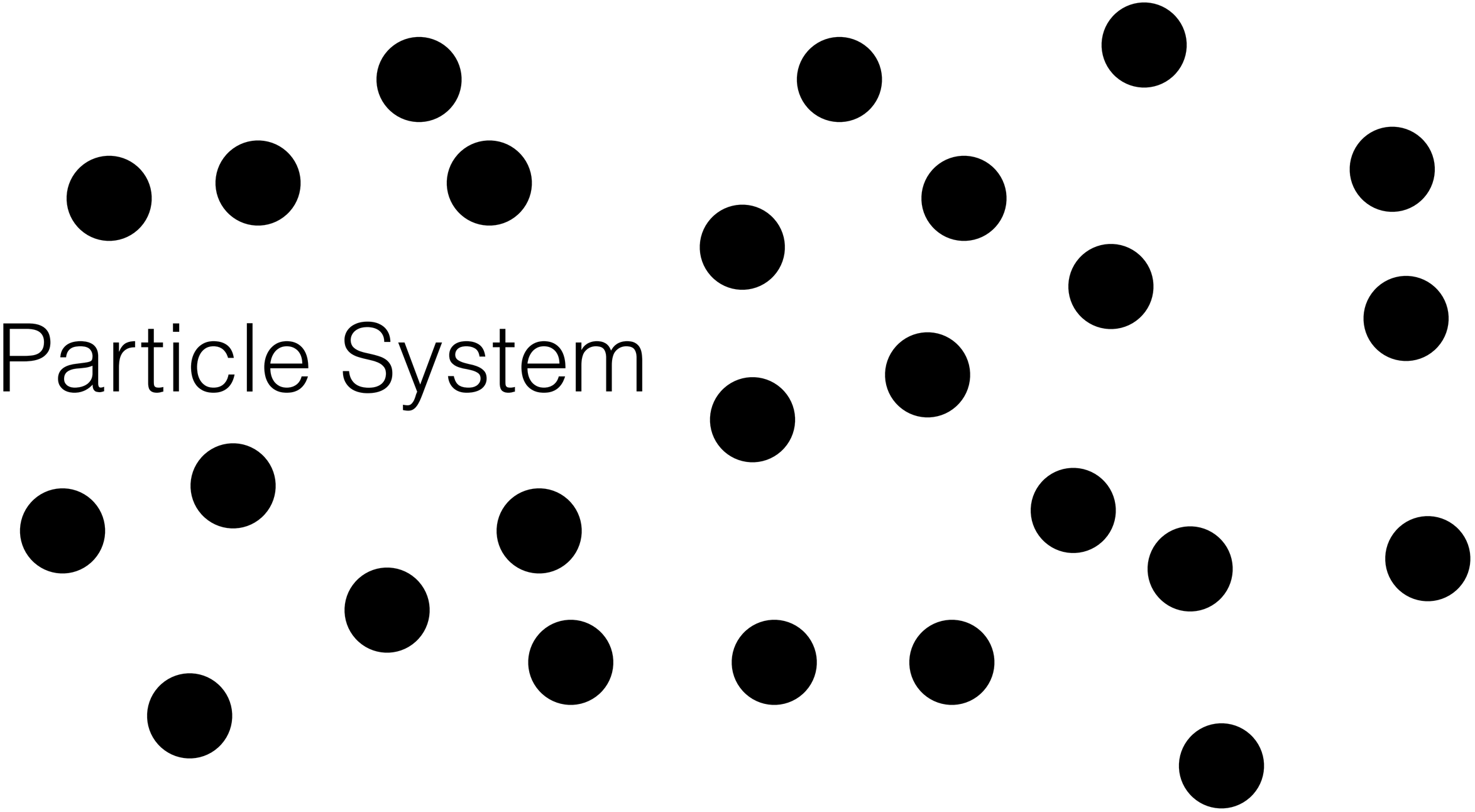
ci::length(ci::vec2)

This function can get you the length of your vector

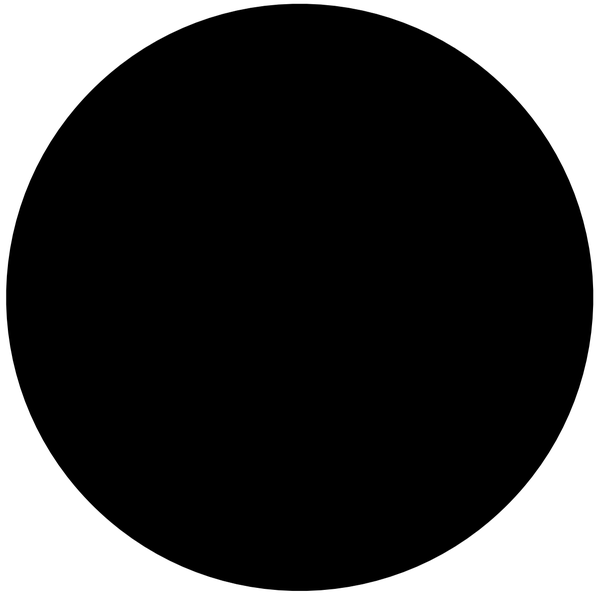
Particle System

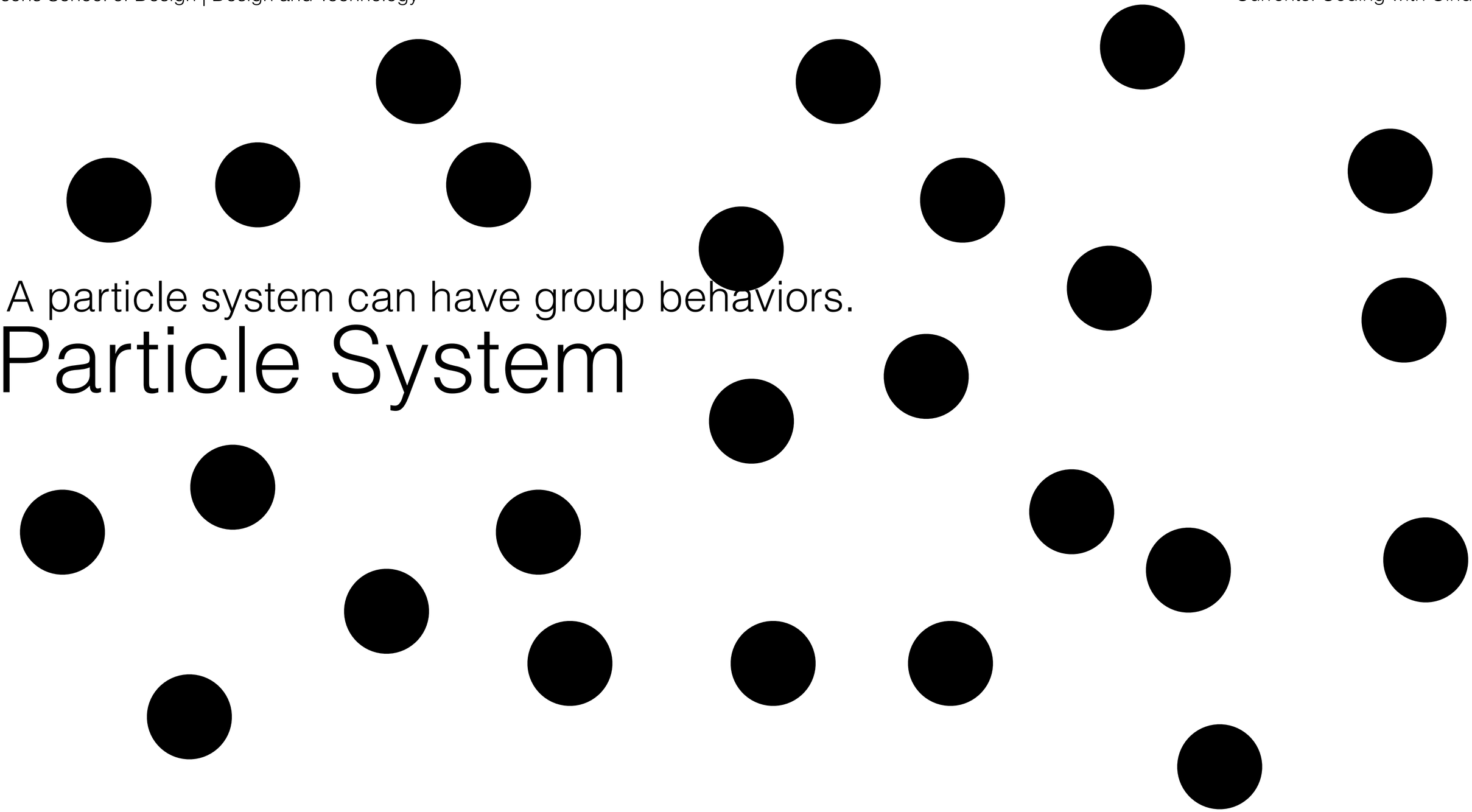


Particle System



Individual particle will have their own parameters - like position, acceleration, and etc.





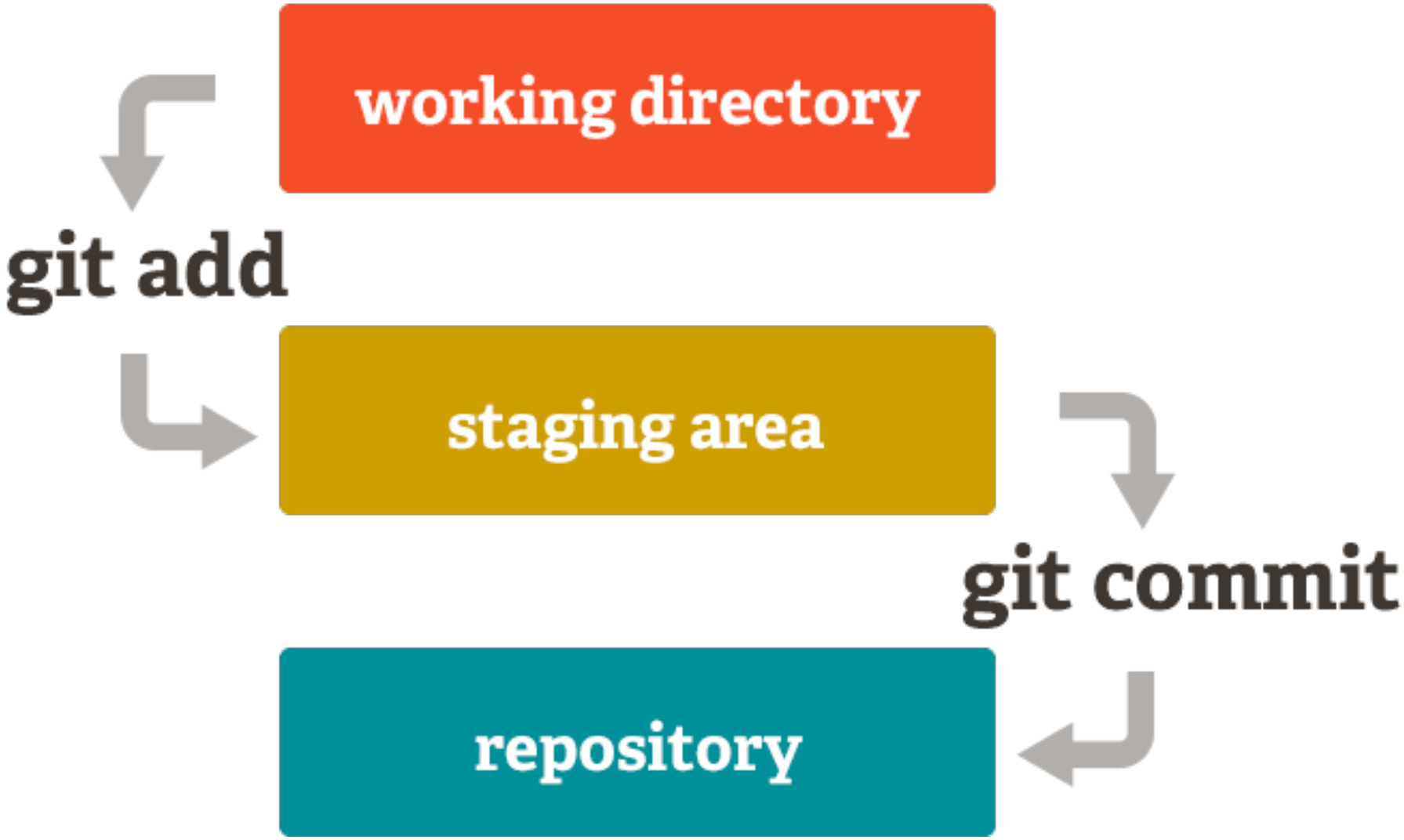
A particle system can have group behaviors.

Particle System

Navigating in console

List all files in directory:	ls
Go in to directory:	cd [directory]
Go up:	cd ..
Open this folder:	open .

Git is not GitHub


working directory
git add
staging area
git commit
repository

Basic git commands

Initialize this directory:	git init
add file to HEAD:	git add [file]
commit current changes:	git commit -m “your message”
check your history:	git log
go to a particular history branch:	git checkout [commit id]

go here for a cheat-sheet, it is easy

http://rogerdudler.github.io/git-guide/

Homework

1. Use a Particle System to simulate a phenomenon in nature (rain, snow)
2. (Bonus) Integrate images (ever wanna try png images?) into your particle system, so you can do something like snow flakes? or even smoke (a lot of games use image to simulate smoke).
3. (Bonus) Make multiple versions of your particle system which spawn different types of particle. Use inheritance to reduce duplicate code.

Due: Feb 14 (Tue).

Post a video demo to Slack channel. Push code of your first 3 weeks' homework to GitHub, and send the link to your GitHub repo to Slack channel.