Gràfics i Visualització de Dades

Extensions Visualization Mapping

Anna Puig

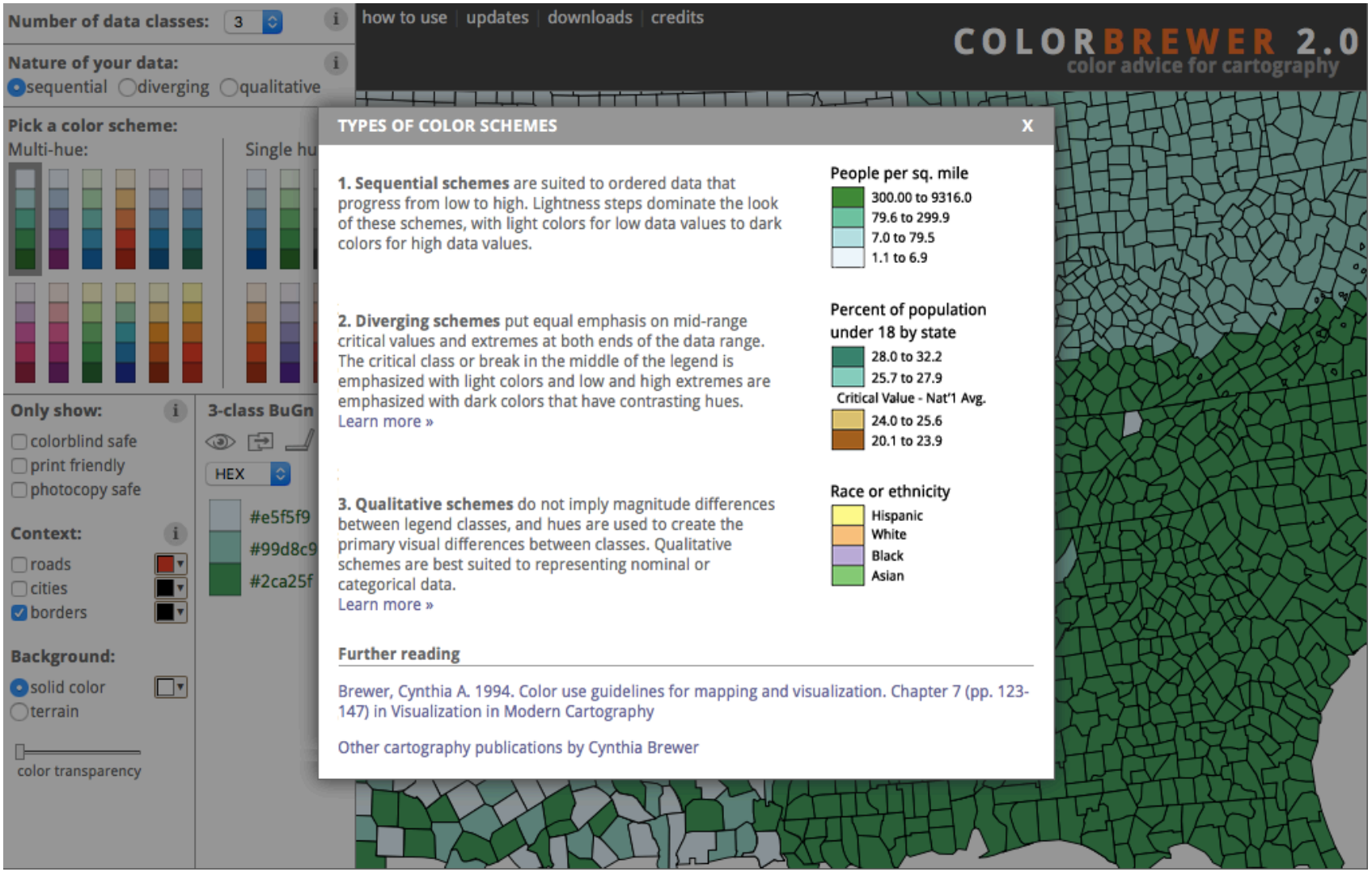
Visualization mapping (extensions de la pràctica)

- Visualization mapping
  - ús de paletes diferents
  - definició de n propietats (localització i gizmo)
  - animacions de les dades
  - mapeig de les dades en una esfera

Visualization mapping (extensions de la pràctica)

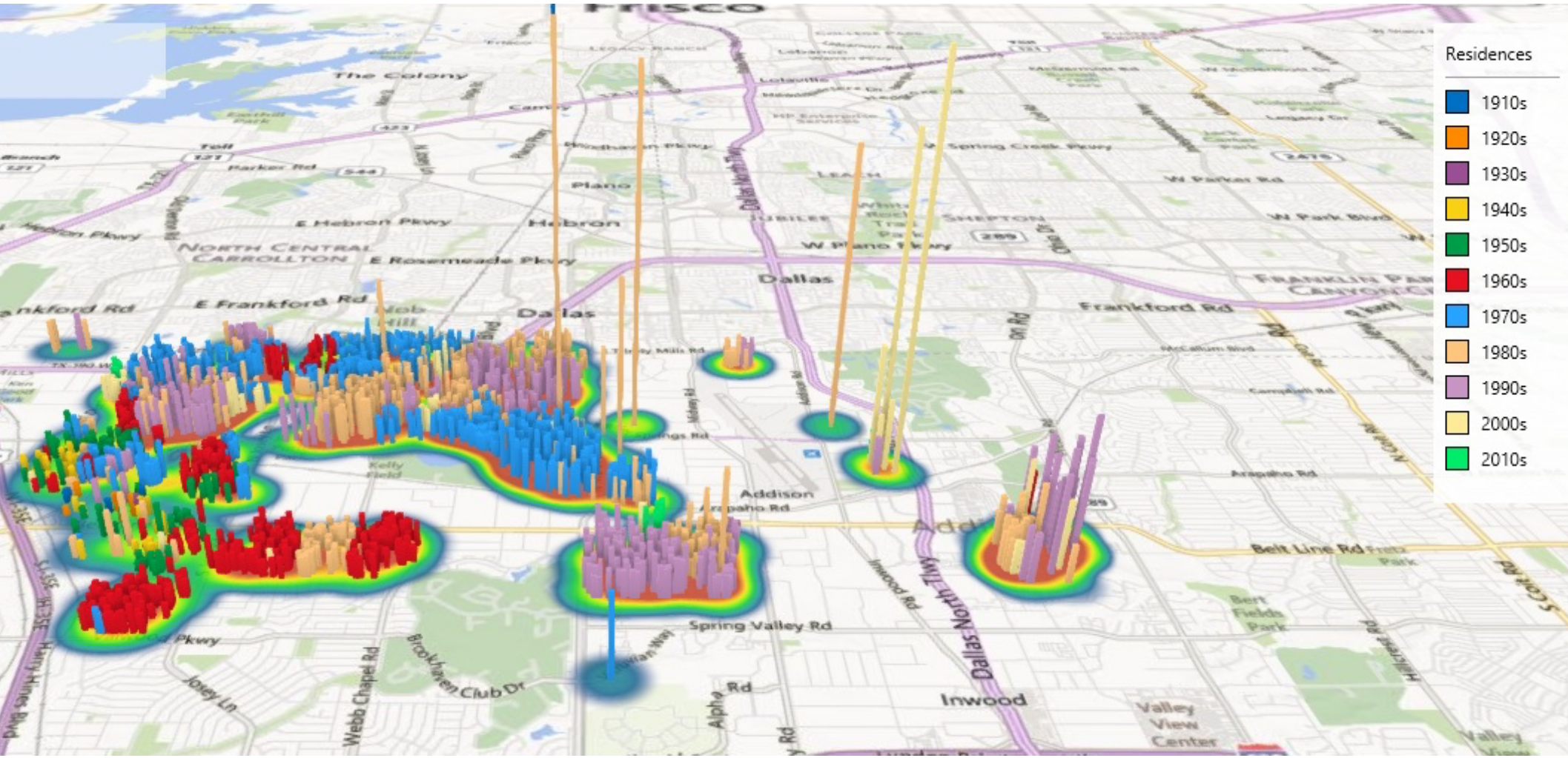
- Ús de paletes diferents

http://colorbrewer2.org



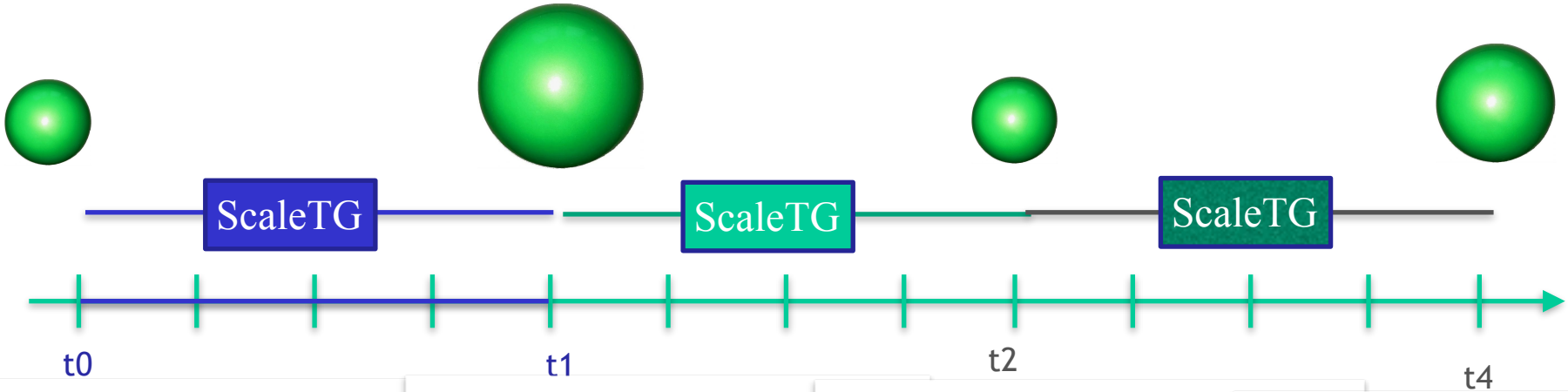
Visualization mapping (extensions de la pràctica)

- n propietats (localització i gizmos)



Visualization mapping (extensions de la pràctica)

- animacions de les dades (sèries temporals)




```
2.1621234 41.4056165  1000000
2.1677698 41.4034215  290000
2.135380995 41.389714424  307596
2.1564047 41.4062419  36500
2.185382 41.3855443  94830
2.178407934 41.381142750  307920
2.16018 41.40352  4214
2.175268003 41.379703771  450787
2.164072884 41.379806535  132024
2.1550105 41.3984803  51072
2.149045108 41.394592939  1300000
2.191080074 41.397724772  882703
2.169583037 41.375154356  725000
2.191159281 41.399440359  191125
2.175174697 41.380544047  1242221
2.175268003 41.379703771  818323
```

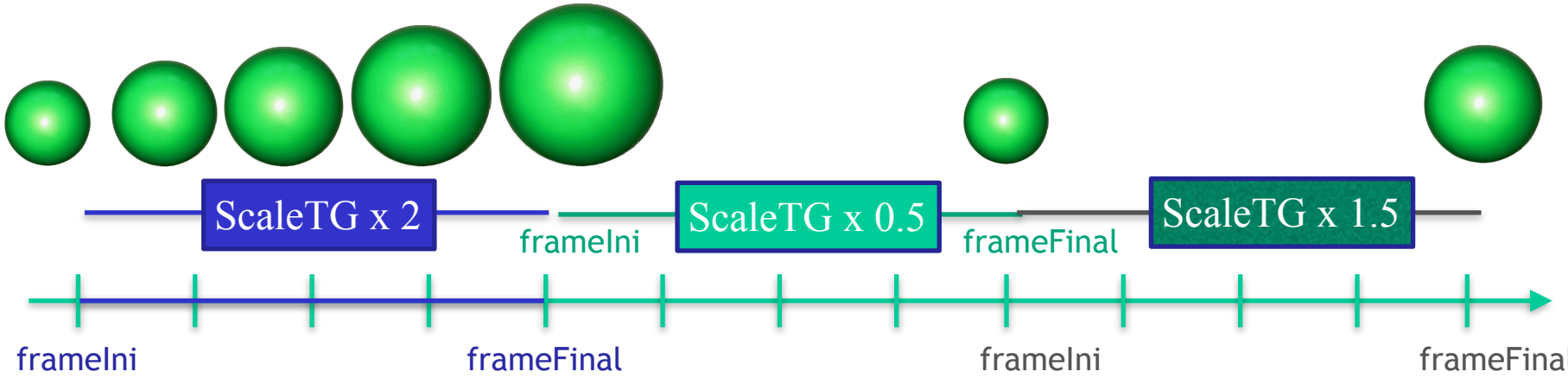
```
2.1621234 41.4056165  50003
2.1677698 41.4034215  190012
2.135380995 41.389714424  27
2.1564047 41.4062419  61650
2.185382 41.3855443  122483
2.178407934 41.381142750  50
2.16018 41.40352  232143
2.175268003 41.379703771  21
2.164072884 41.379806535  53
2.1550105 41.3984803  251724
2.149045108 41.394592939  12
2.191080074 41.397724772  32
2.169583037 41.375154356  73
2.191159281 41.399440359  92
2.175174697 41.380544047  22
2.175268003 41.379703771  11
```

```
2.1621234 41.4056165  81040
2.1677698 41.4034215  490200
2.135380995 41.389714424  475
2.1564047 41.4062419  316502
2.185382 41.3855443  294833
2.178407934 41.381142750  247
2.16018 41.40352  1121400
2.175268003 41.379703771  135
2.164072884 41.379806535  131
2.1550105 41.3984803  81724
2.149045108 41.394592939  140
2.191080074 41.397724772  112
2.169583037 41.375154356  890
2.191159281 41.399440359  119
2.175174697 41.380544047  122
2.175268003 41.379703771  898
```

```
2.1621234 41.4056165  101001
2.1677698 41.4034215  292001
2.135380995 41.389714424  137596
2.1564047 41.4062419  265012
2.185382 41.3855443  14833
2.178407934 41.381142750  179320
2.16018 41.40352  422142
2.175268003 41.379703771  1157387
2.164072884 41.379806535  202224
2.1550105 41.3984803  101072
2.149045108 41.394592939  1200000
2.191080074 41.397724772  1282703
2.169583037 41.375154356  954000
2.191159281 41.399440359  69235
2.175174697 41.380544047  132221
2.175268003 41.379703771  898313
```

Visualization mapping (extensions de la pràctica)

- animacions de les dades (sèries temporals)




```
int frameIni;
int frameFinal;
shared_ptr<TG> transf;
```

```
int frameIni;
int frameFinal;
shared_ptr<TG> transf;
```

```
int frameIni;
int frameFinal;
shared_ptr<TG> transf;
```

TG serà una ScaleTG(1.189)

r=0.5 r=1

$\Delta S = \sqrt[4]{S}$

Animation

```
int frameIni;
int frameFinal;
shared_ptr<TG> transf;
```

Visualization mapping (extensions de la pràctica)

- animacions de les dades (sèries temporals)

Run Animation

```
void MainWindow::runAnimation() {

    int width, height;
    vector<QImage> frames;

    // Camera parameters from the Controller
    auto camera = Controller::getInstance()->getSetUp()->getCamera();
    width = camera->viewportX;
    height = camera->viewportY;

    // TODO: Canviar per incloure animacions diferents
    // Ara es crea aqui una escena amb una esfera que té animacions,
    // Caldria crear l'escena des de fitxer (opcionalment)
    // es crida el render i es guarden
    // tantes imatges com a frames s'han calculat (exemple amb MAXFRAMES = 5 a Animation.hh)

    Controller::getInstance()->createScene(MAXFRAMES);
    for (int i=0; i<MAXFRAMES; i++) {
        QImage *image = new QImage(width, height, QImage::Format_RGB888);
        Controller::getInstance()->update(i);
        Controller::getInstance()->rendering(image);
        frames.push_back(*image);
    }

    outputFile->saveAnimation(frames);
    frames.clear();
}
```

Visualization mapping (extensions de la pràctica)

- animacions de les dades (sèries temporals)

Run Animation

```
void MainWindow::runAnimation

    int width, height;
    vector<QImage> frames;

    // Camera parameters from
    auto camera = Controller:
    width = camera->viewportX
    height = camera->viewport

    // TODO: Canviar per incl
    // Ara es crea aqui una e
    // Caldria crear l'escena
    // es crida el render i es guarden
    // tantes imatges com a frames s'ha calculat (exemple amb MAXFRAMES = 5 a Animation.hh)

    Controller::getInstance()->createScene(MAXFRAMES);
    for (int i=0; i<MAXFRAMES; i++) {
        QImage *image = new QImage(width, height, QImage::Format_RGB888);
        Controller::getInstance()->update(i);
        Controller::getInstance()->rendering(image);
        frames.push_back(*image);
    }

    outputFile->saveAnimation(frames);
    frames.clear();
}
```

```
bool Controller::createScene(int nFrames) {

    //TO DO Fase 3 opcional: Codi exemple amb animacions però que es pot canviar
    // pel que creguis convenient

    auto sphere = make_shared<Sphere>(vec3(0, 0, -1), 0.5, 1.0);
    sphere->setMaterial(make_shared<Lambertian>(vec3(0.5, 0.2, 0.7)));

    shared_ptr<Animation> anim = make_shared<Animation>();
    anim->transf =  make_shared<TranslateTG>(vec3(0.2));
    sphere->addAnimation(anim);

    return true;
}
```

Visualization mapping (extensions de la pràctica)

- animacions de les dades (sèries temporals)

Run Animation

```
void MainWindow::runAnimation() {

    int width, height;
    vector<QImage> frames;

    // Camera parameters from the Controller
    auto camera = Controll
    width = camera->viewpo
    height = camera->viewp

    // TODO: Canviar per i
    // Ara es crea aqui un
    // Caldria crear l'esc
    // es crida el render
    // tantes imatges com

    Controller::getInstance()->createScene(MA RAMES);
    for (int i=0; i<MAXFRAMES; i++) {
        QImage *image = new QImage(width, height, QImage::Format_RGB888);
        Controller::getInstance()->update(i);
        Controller::getInstance()->rendering(image);
        frames.push_back(*image);
    }

    outputFile->saveAnimation(frames);
    frames.clear();
}
```

Scene

```
void Scene::update(int nframe) {
    for (unsigned int i = 0; i< objects.size(); i++) {
        objects[i]->update(nframe);
    }
}
```

Visualization mapping (extensions de la pràctica)

- animacions de les dades (sèries temporals)


```
void Animable::addAnimation(shared_ptr<Animation> anim) {
    animFrames.push_back(anim);
}
```

```
void Animable::update(int nframe) {

    bool trobat = false;
    int i;
    for (i = 0; i<animFrames[animFrames.size()-1]->frameFinal && !trobat; i++)
        trobat = animFrames[i]->frameFinal>=nframe;

    aplicaTG(animFrames[i-1]->transf);
}
```

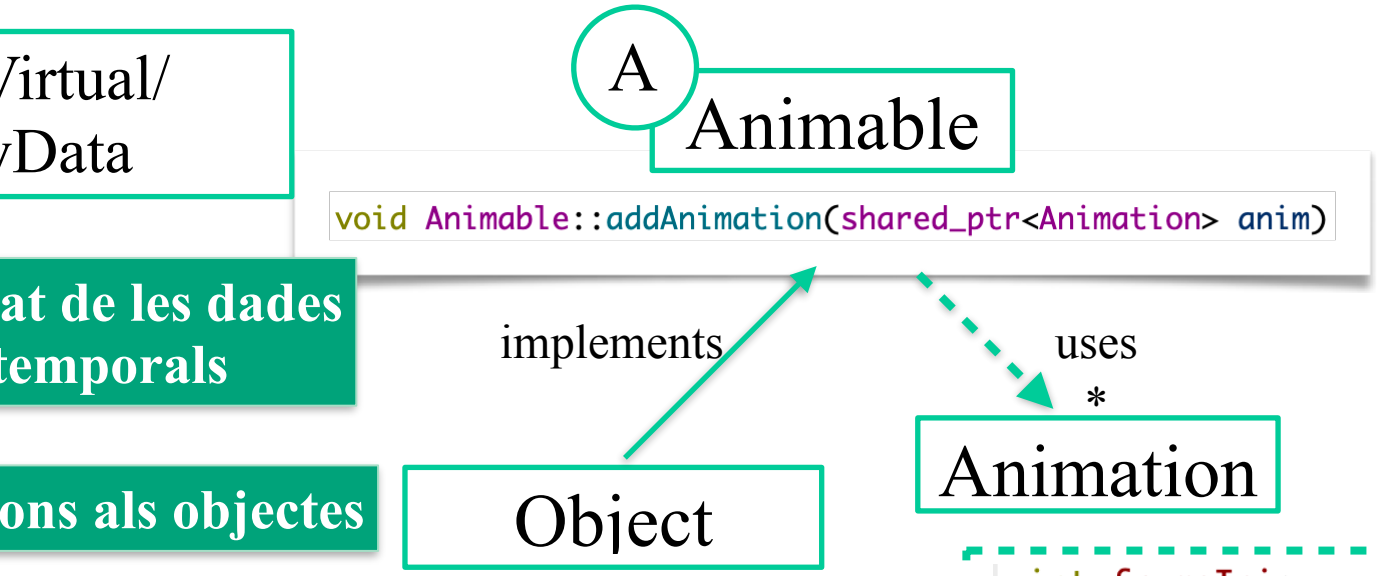
Visualization mapping (extensions de la pràctica)

- animacions de les dades (sèries temporals)

SceneFactoryVirtual/
SceneFactoryData

Modificar el format de les dades per tenir n sèries temporals

Afegir les animacions als objectes




```
auto sphere = make_ptr<Sphere>(vec3(0, 0, -1), 0.5, v);

auto anim = make_ptr<Animation>();
anim->transf = make_ptr<ScaleTG>(vec3(0.2));
sphere->addAnimation(anim);
```

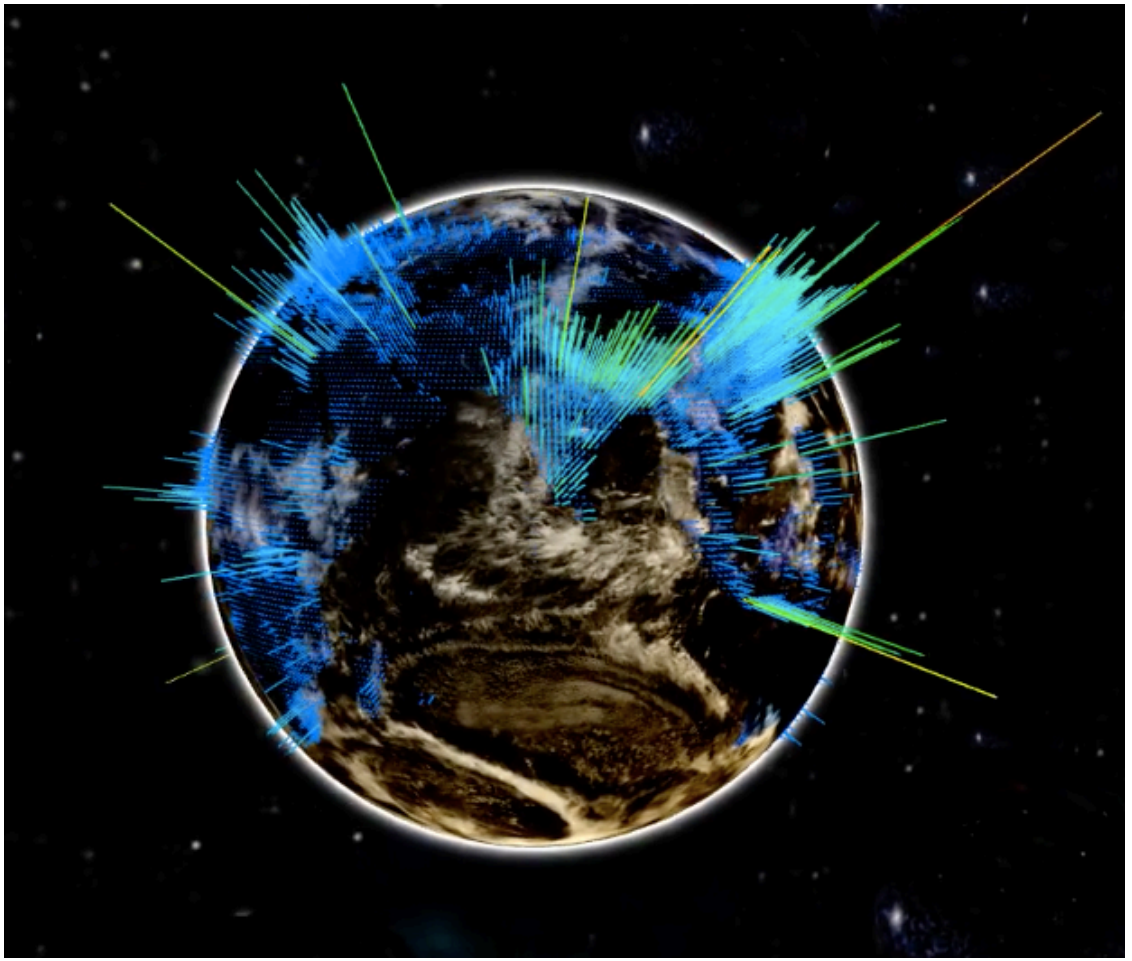
.json

TEMPORALVW o TEMPORALDATA

```
"numInstants": 10
"translation": [ -1, -2, 3],
"scale": [2.0],
"rotation": [30, 1, 0, 0]
```

Visualization mapping (extensions de la pràctica)

- Mapeig de les dades en una esfera



https://www.covidvisualizer.com