ESCAPE 824064

Development Practice

Pierre Aubert

Introduction

Introduction

Change mind on **implementation**

Introduction

Change mind on **implementation**

New **Idea**

Introduction

Change mind on **implementation**

New **Idea**

Learning :

- New **Concept**
- New **Language**

Introduction

Change mind on **implementation**

New **Idea**

Learning :

- New **Concept**
- New **Language**

Better view of work to be done

Introduction

Change mind on **implementation**

New **Idea**

Learning :

- New **Concept**
- New **Language**

Better view of work to be done

Good advice of a collegue

Introduction

Change mind on **implementation**

New **Idea**

Learning :

- New **Concept**
- New **Language**

Better view of work to be done

Good advice of a collegue

So we **change** our way of **implementing** stuff

Introduction

Change mind on **implementation**

New **Idea**

Learning :
- New **Concept**
- New **Language**

Better view of work to be done

Good advice of a collegue

So we **change** our way of **implementing** stuff

Generaly : **Nice Improvement**

Introduction

Change mind on **implementation**

New **Idea**

Learning :
- New **Concept**
- New **Language**

Better view of work to be done

Good advice of a collegue

So we **change** our way of **implementing** stuff

Generaly : **Nice Improvement**

Sometimes : **Break**

Introduction

Change mind on **implementation**

New **Idea**

Learning :

- New **Concept**
- New **Language**

Better view of work to be done

Good advice of a collegue

So we **change** our way of **implementing** stuff

Generaly : **Nice Improvement**

Sometimes : **Break**

Why :

- Start with an **idea**
- Finish with **another idea**
- But **not compatible**

How ?

How ?

Documentation

How ?

Documentation

Versioning :
- **History** of modifications
- **Restore** previous versions
- **Prepare** future versions

How ?

Documentation

Versioning :
- **History** of modifications
- **Restore** previous versions
- **Prepare** future versions

Unit Tests :
- **Check** if something is wrong
- Can be **automatised**

How ?

Documentation

Versioning :
- **History** of modifications
- **Restore** previous versions
- **Prepare** future versions

Unit Tests :
- **Check** if something is wrong
- Can be **automatised**

Development Method :
- Simplify **new features**
- Ease **bug fix**
- Enhence **reusability**

How ?

Development Method :
- Simplify **new features**
- Ease **bug fix**
- Enhence **reusability**

Documentation

Versioning :
- **History** of modifications
- **Restore** previous versions
- **Prepare** future versions

Software Architecture :
- Interfaces

Unit Tests :
- **Check** if something is wrong
- Can be **automatised**

How ?

Documentation

Versioning :
- **History** of modifications
- **Restore** previous versions
- **Prepare** future versions

Unit Tests :
- **Check** if something is wrong
- Can be **automatised**

Development Method :
- Simplify **new features**
- Ease **bug fix**
- Enhence **reusability**

Software Architecture :
- Interfaces

Profiling :
- Time functions

Documentation

readme.md :
- Showcase
- Depedencies
- Installation / Compitation
- Use cases

Development Documentation :
- Doxygen
- Sphinx

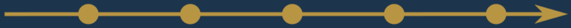
Versionning

Versionning

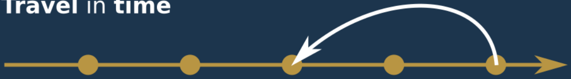
Save **all changes** in the project

Versionning

Save **all changes** in the project



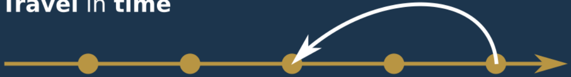
Travel in **time**



Versionning

Save **all changes** in the project

Travel in **time**

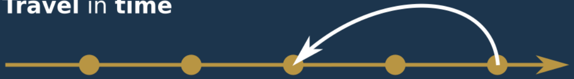


Coherent save of the project
- **As many local** / **remote saves** you need

Versionning

Save **all changes** in the project

Travel in **time**



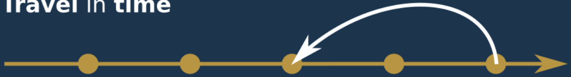
Coherent save of the project
- **As many local** / **remote saves** you need

Prevent breaking projet with **old modification**

Versionning

Save **all changes** in the project

Travel in **time**



Coherent save of the project
- **As many local / remote saves** you need

Prevent breaking projet with **old modification**

Versionning

Save **all changes** in the project

Travel in **time**

Coherent save of the project
- **As many local / remote saves** you need

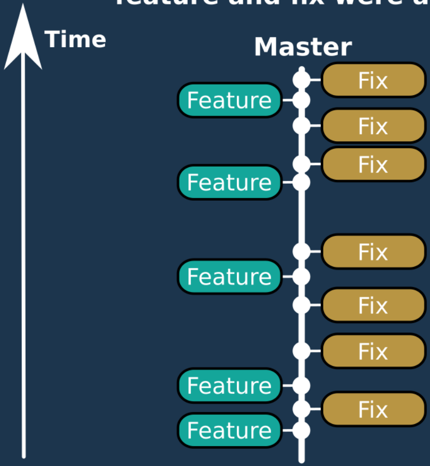
Prevent breaking projet with **old modification**

Gitlab

https://gitlab.in2p3.fr

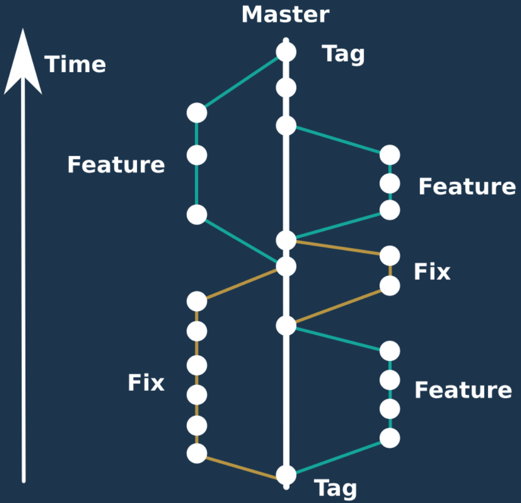
Git History

Tool to know when feature and fix were added



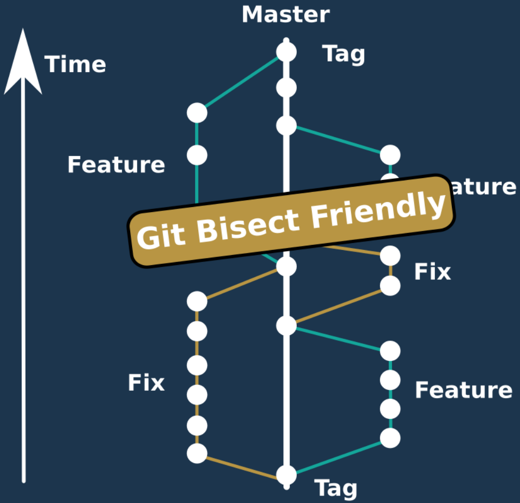
Branches

Classical Workflow



Branches

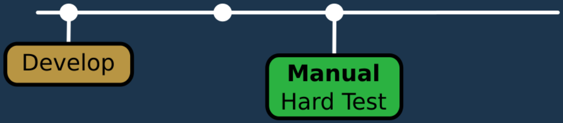
Classical Workflow



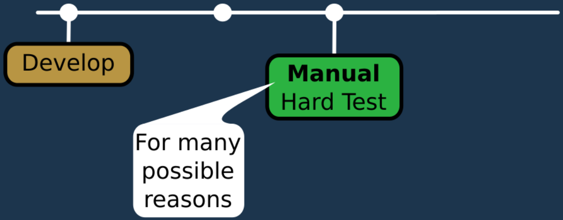
Git Bisect

Git Bisect

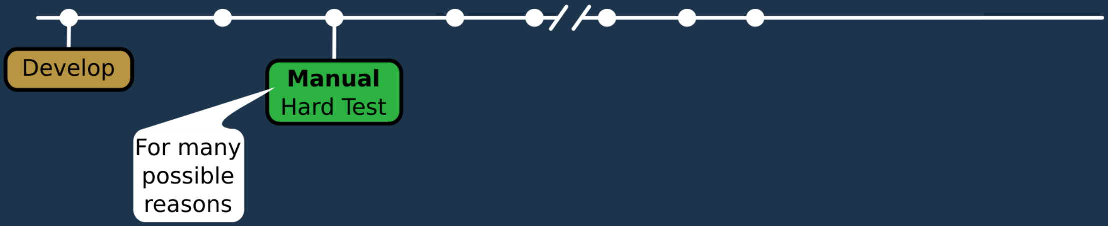
Git Bisect



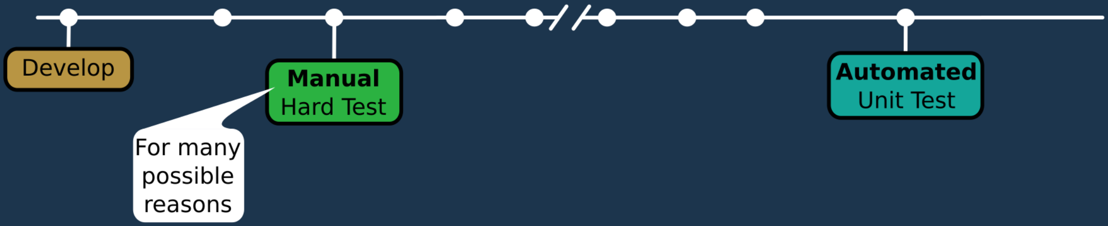
Git Bisect



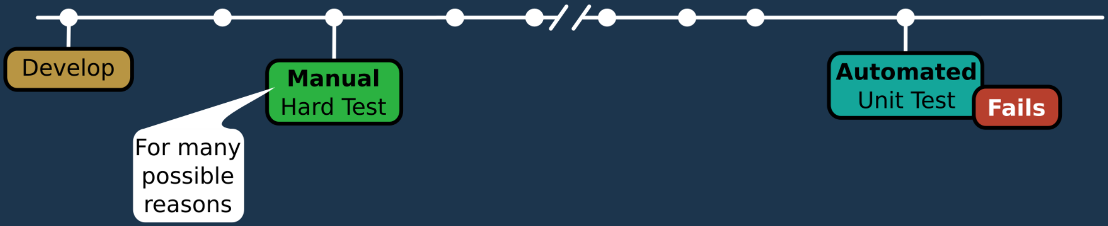
Git Bisect



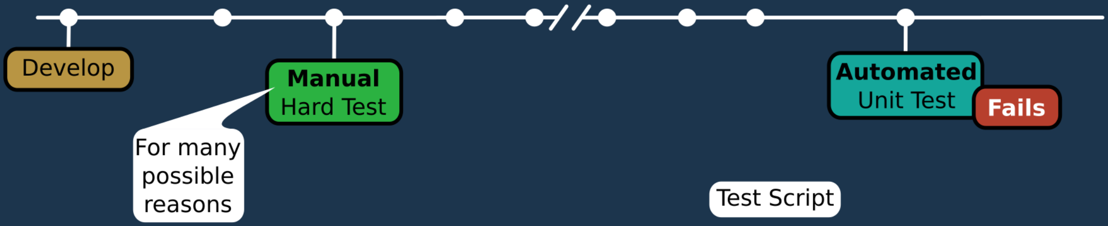
Git Bisect


Develop
Manual Hard Test
For many possible reasons
Automated Unit Test

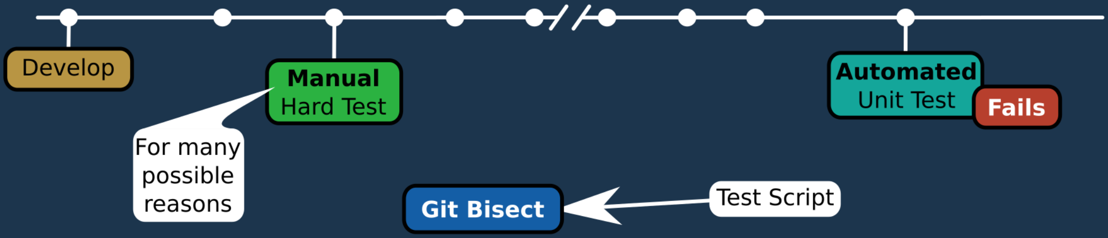
Git Bisect


Develop
Manual Hard Test
For many possible reasons
Automated Unit Test
Fails

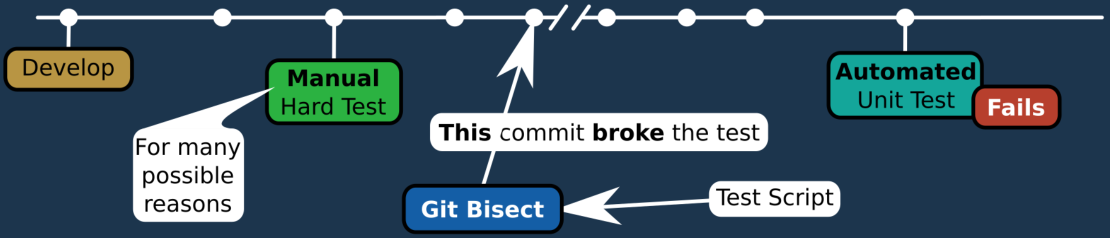
Git Bisect


Develop
Manual
Hard Test
For many possible reasons
Automated
Unit Test
Fails
Test Script

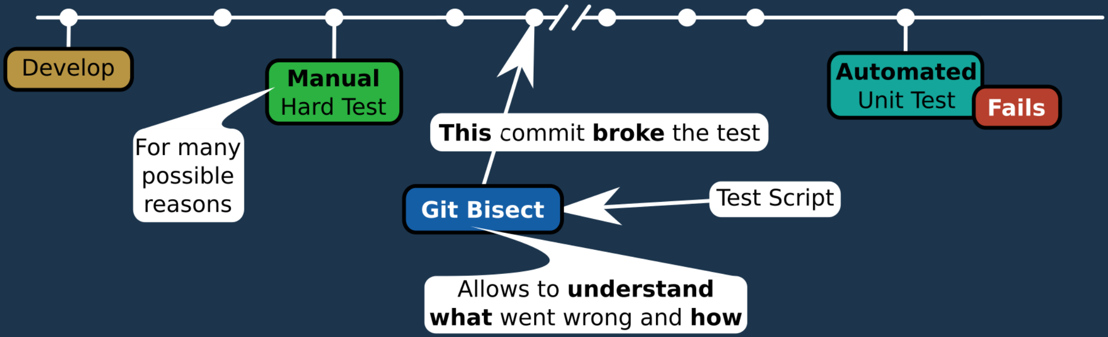
Git Bisect


Develop
Manual Hard Test
For many possible reasons
Automated Unit Test
Fails
Git Bisect
Test Script

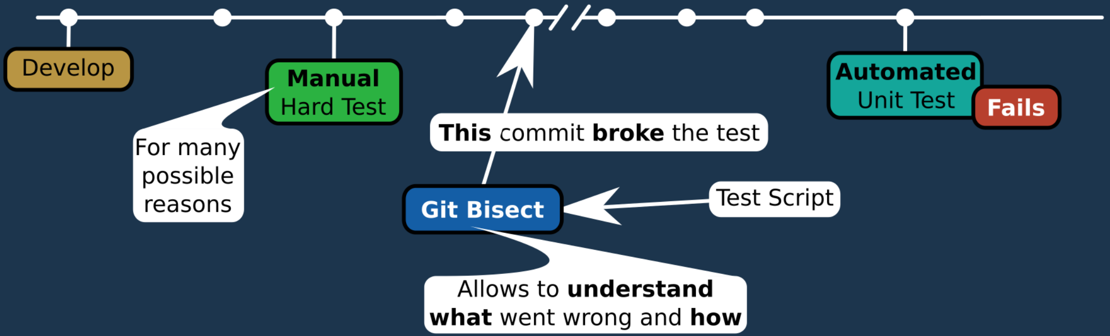
Git Bisect


Develop
Manual Hard Test
For many possible reasons
This commit broke the test
Git Bisect
Test Script
Automated Unit Test
Fails

Git Bisect

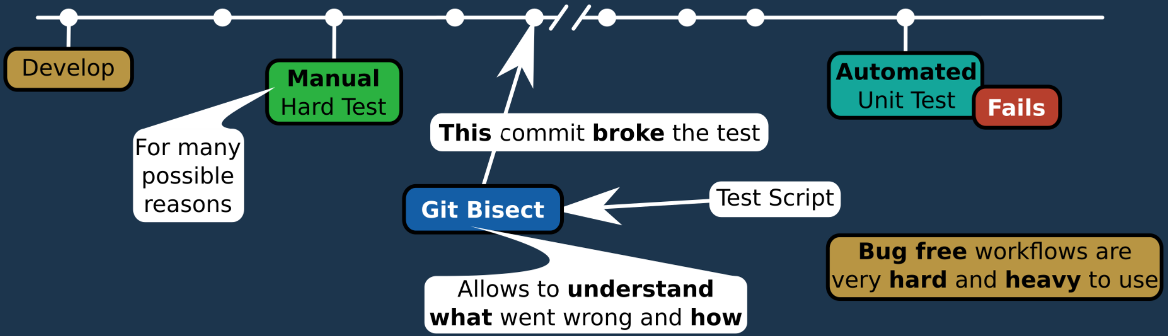

Develop
Manual Hard Test
For many possible reasons
This commit broke the test
Automated Unit Test
Fails
Git Bisect
Test Script
Allows to understand what went wrong and how

Git Bisect


Develop
Manual Hard Test
For many possible reasons
This commit broke the test
Automated Unit Test
Fails
Git Bisect
Test Script
Allows to understand what went wrong and how

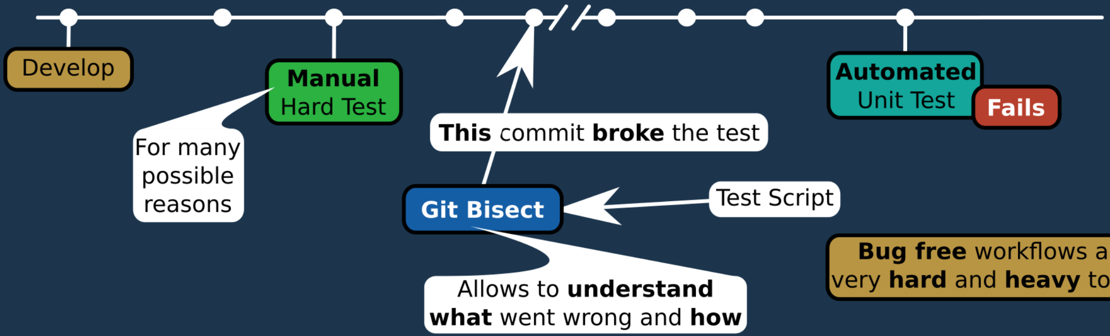
Development :
- **Not only** about adding **features** and finding **bugs**
- **Understanding why** a bug appeared and **how** to prevent it

Git Bisect


Develop
Manual Hard Test
For many possible reasons
This commit broke the test
Automated Unit Test
Fails
Git Bisect
Test Script
Allows to understand what went wrong and how
Bug free workflows are very hard and heavy to use

Development :
- **Not only** about adding **features** and finding **bugs**
- **Understanding why** a bug appeared and **how** to prevent it

Git Bisect


Develop
Manual Hard Test
For many possible reasons
This commit **broke** the test
Automated Unit Test
Fails
Git Bisect
Test Script
Allows to **understand** **what** went wrong and **how**

Bug free workflows are very **hard** and **heavy** to use

Some workflows can **fasten** bugs **research** and **fix**

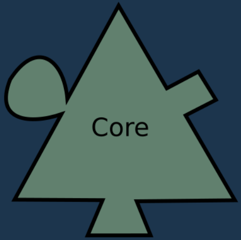
Development :
- **Not only** about adding **features** and finding **bugs**
- **Understanding why** a bug appeared and **how** to prevent it

Interfaces

Interface : Border

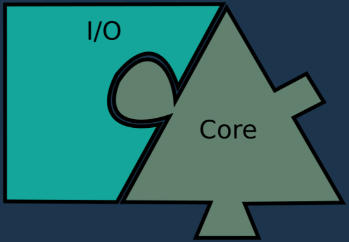
Interfaces

Interface : Border



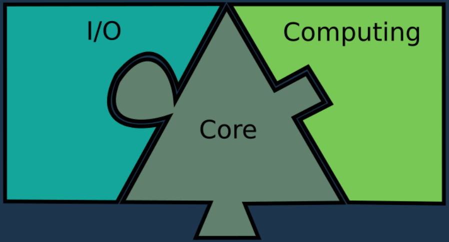
Interfaces

Interface : Border



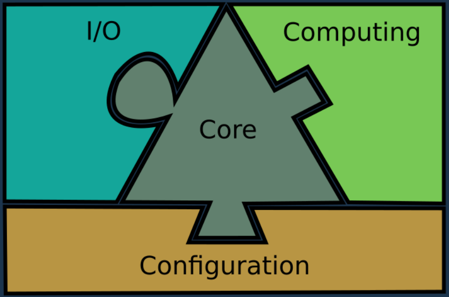
Interfaces

Interface : Border



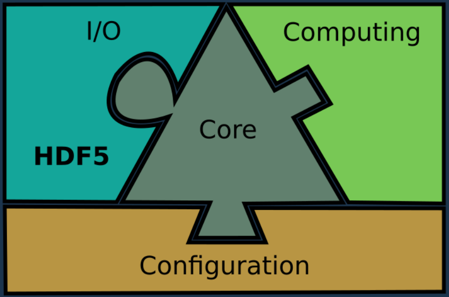
Interfaces

Interface : Border



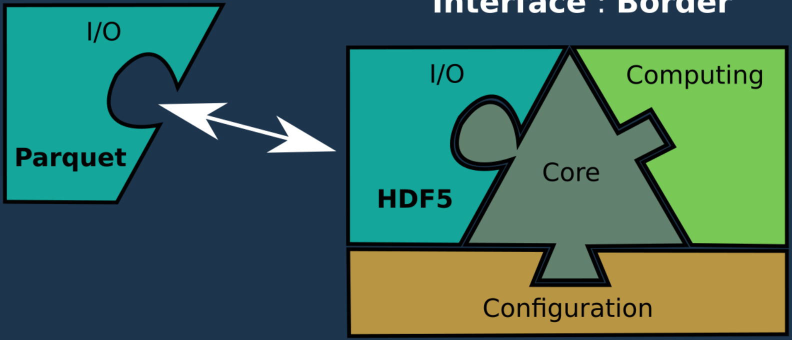
Interfaces

Interface : Border



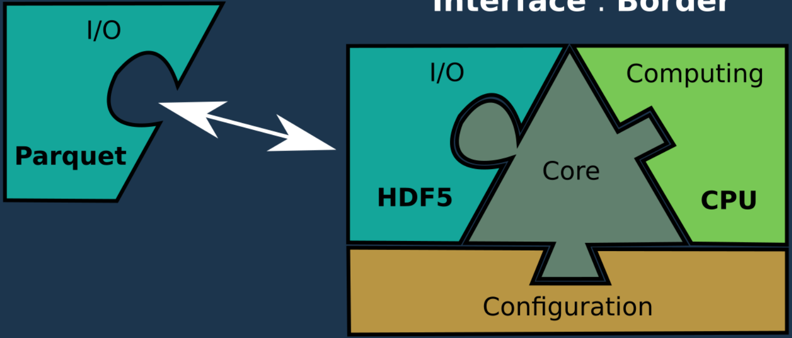
Interfaces

Interface : Border



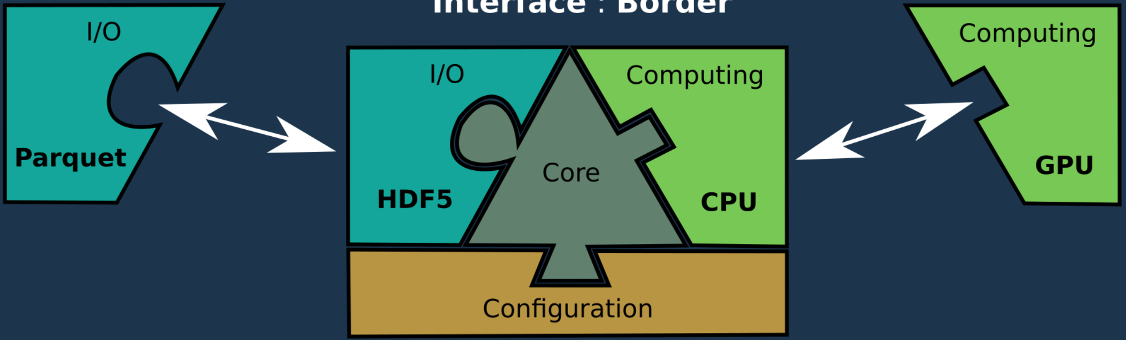
Interfaces

Interface : Border



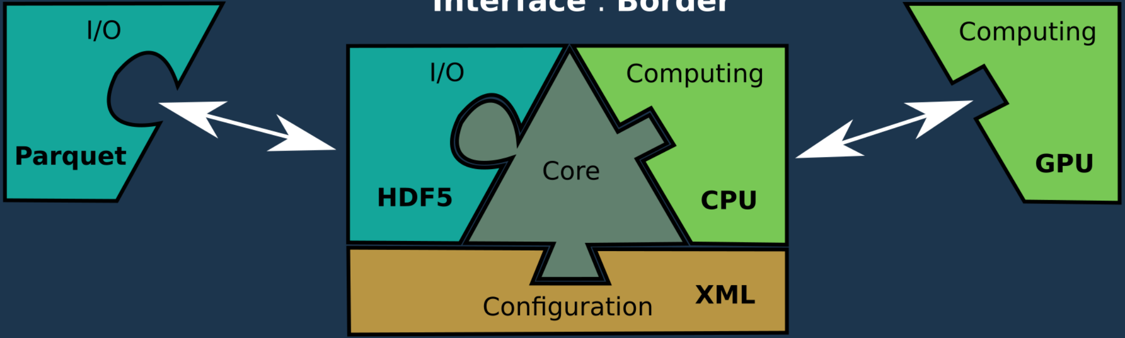
Interfaces

Interface : Border



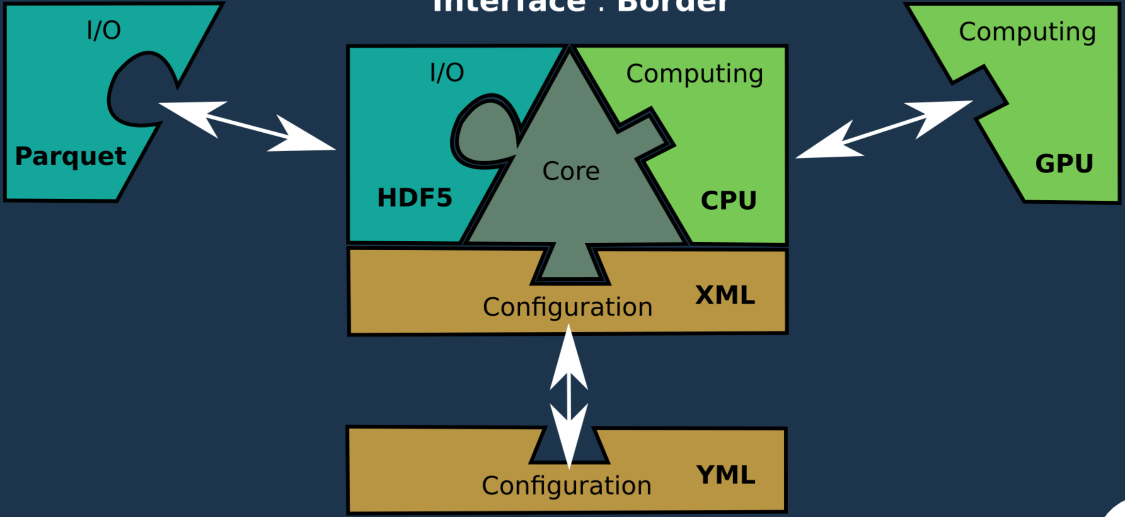
Interfaces

Interface : Border



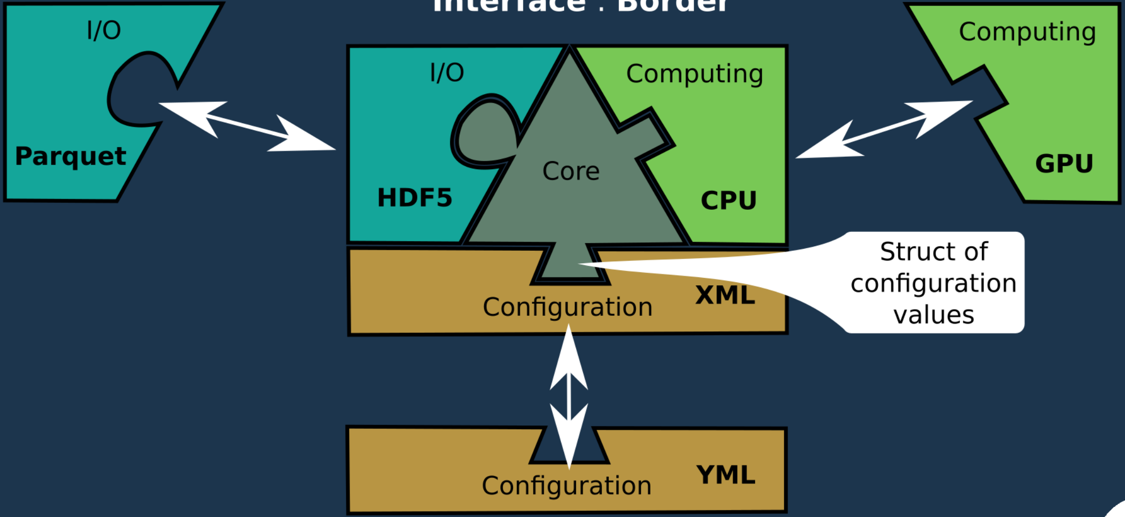
Interfaces

Interface : Border



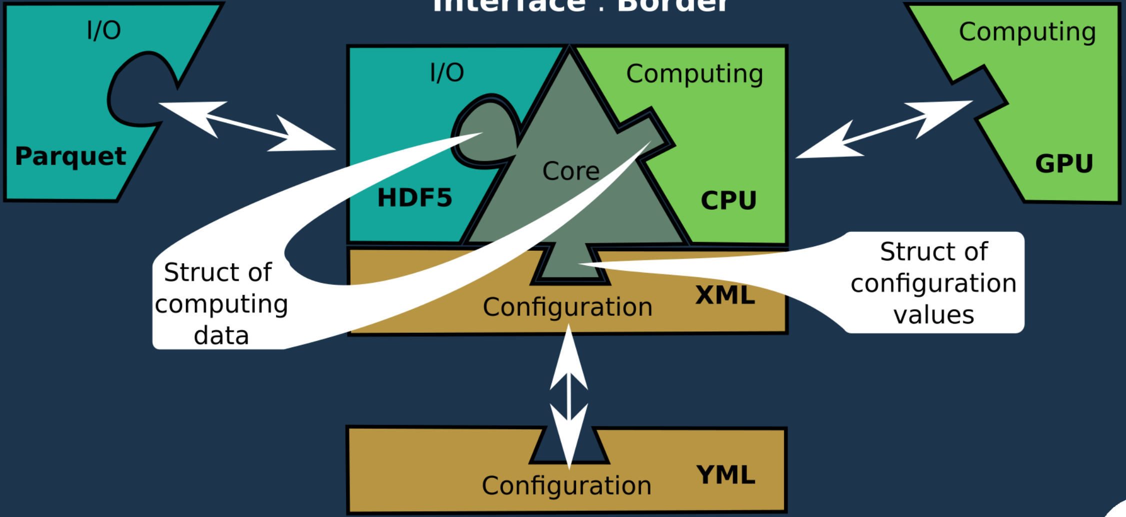
Interfaces

Interface : Border



Interfaces

Interface : Border

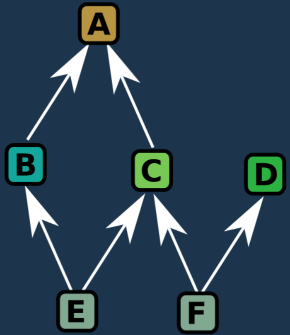


Constellation of projects

Increase **reusability** of projects

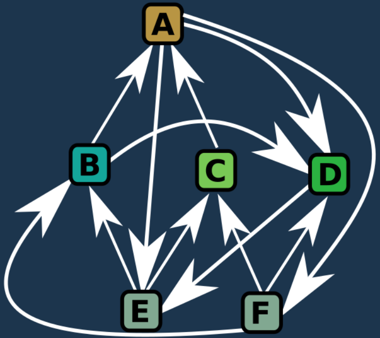
Constellation of projects

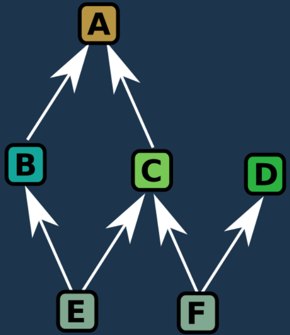
Increase **reusability** of projects



Constellation of projects

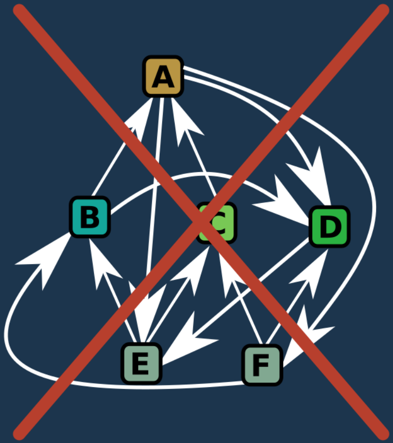
Increase **reusability** of projects

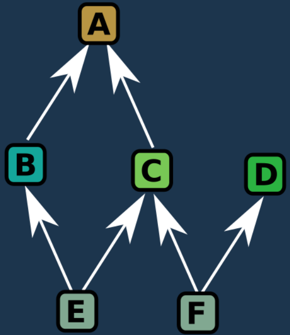




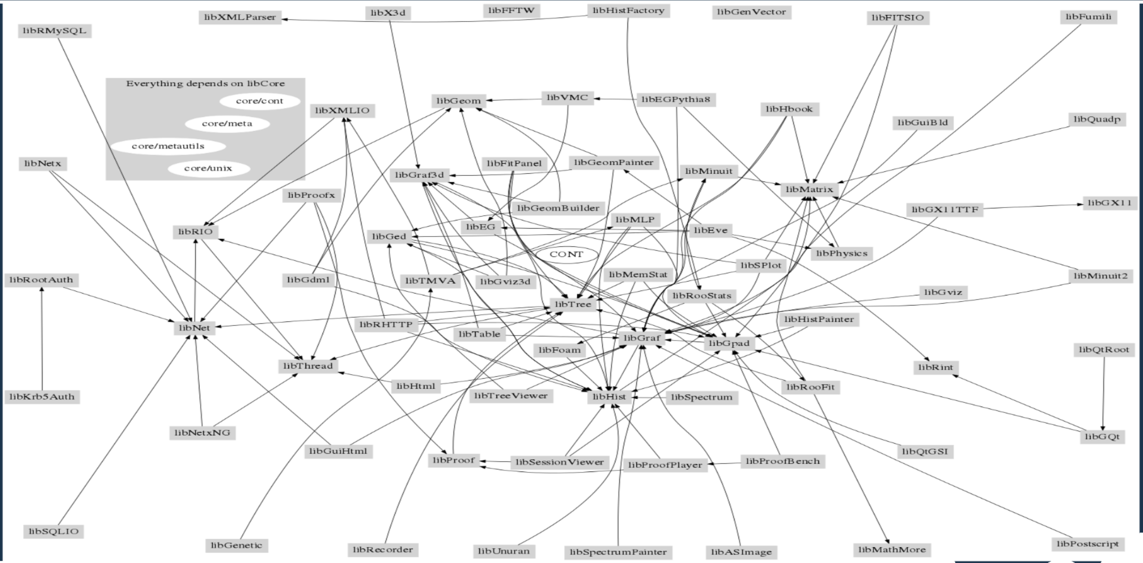
Constellation of projects

Increase **reusability** of projects





Do not do that !!!



What does this code do ?

```
float a, q, p, r, s;for(size_t _(0ul);_<m_;++_){a_[_]=0;b[_]=0
;c[_]=0;d[_]=0;e[_]=0;f[_]=0;g[_]=0;h[_]=0;i[_]=0;j[_]=0:k[_]=
0;l[_]=0;m[_]=0;n[_]=0;o[_]=0;for(size_t __(0ul);__<n;++__){a=
w[_*n+__];a_[_]+=a;q=u[__],p=v[__];r=a*q;s=a*p;b[_]+=r;c[_]+=s
;d[_]+=q*r;f[_]+=s*p;g[_]+=q*q*a*q;j[_]+=s*p*p;k[_]+=r*q*q*q;o
[_]+=p*p*a*p*p;e[_]+=r*p;h[_]+=q*a*q*p;i[_]+=p*q*p*a;l[_]+=p*q
*q*a*q;n[_]+=p*a*p*q*p;m[_]+=q*p*s*q;}}
```

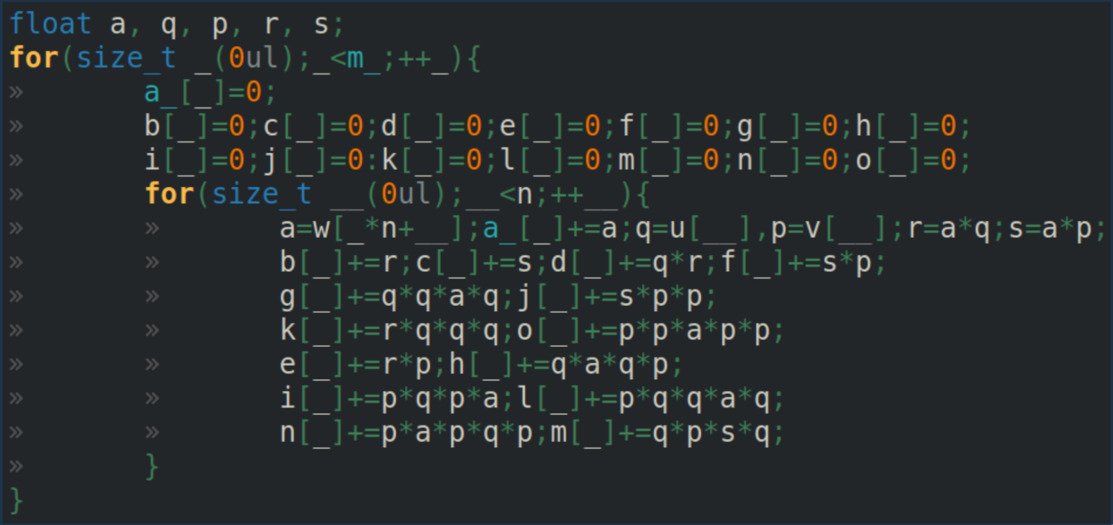
What does this code do ?

```
float a, q, p, r, s;for(size_t _(0ul);_<m_;++_){a_[_]=0;b[_]=0
;c[_]=0;d[_]=0;e[_]=0;f[_]=0;g[_]=0;h[_]=0;i[_]=0;j[_]=0:k[_]=
0;l[_]=0;m[_]=0;n[_]=0;o[_]=0;for(size_t ___(0ul);___<n;++___){a=
w[_*n+___];a_[_]+=a;q=u[___],p=v[___];r=a*q;s=a*p;b[_]+=r;c[_]+=s
;d[_]+=q*r;f[_]+=s*p;g[_]+=q*q*a*q;j[_]+=s*p*p;k[_]+=r*q*q*q;o
[_]+=p*p*a*p*p;e[_]+=r*p;h[_]+=q*a*q*p;i[_]+=p*q*p*a;l[_]+=p*q
*q*a*q;n[_]+=p*a*p*q*p;m[_]+=q*p*s*q;}}
```

What does this code do ?

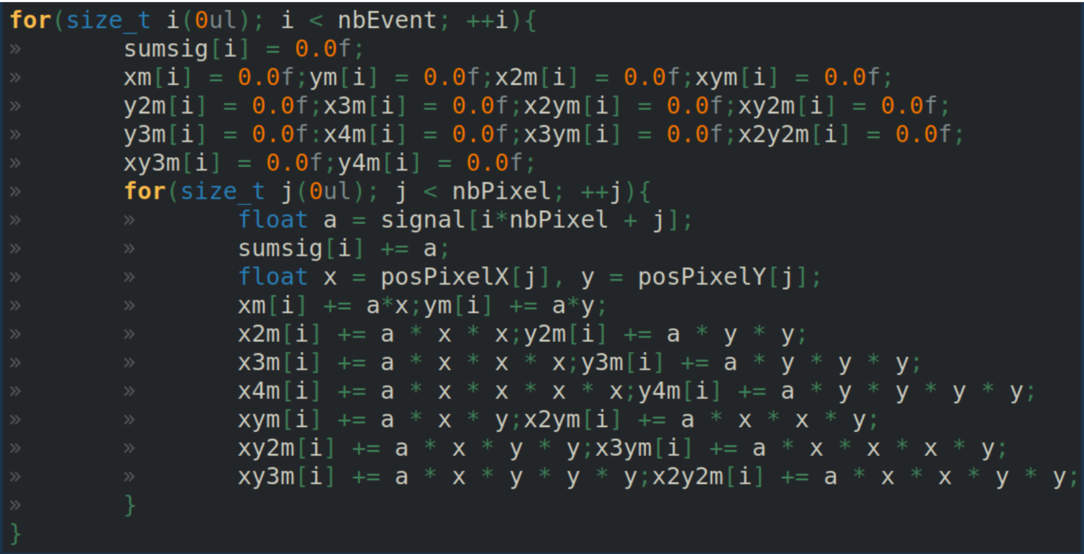
```
float a, q, p, r, s;
for(size_t _(0ul);_<m_;++_){
»       a_[_]=0;
»       b[_]=0;c[_]=0;d[_]=0;e[_]=0;f[_]=0;g[_]=0;h[_]=0;
»       i[_]=0;j[_]=0:k[_]=0;l[_]=0;m[_]=0;n[_]=0;o[_]=0;
»       for(size_t __(0ul);__<n;++__){
»       »       a=w[_*n+__];a_[_]+=a;q=u[__],p=v[__];r=a*q;s=a*p;
»       »       b[_]+=r;c[_]+=s;d[_]+=q*r;f[_]+=s*p;
»       »       g[_]+=q*q*a*q;j[_]+=s*p*p;
»       »       k[_]+=r*q*q*q;o[_]+=p*p*a*p*p;
»       »       e[_]+=r*p;h[_]+=q*a*q*p;
»       »       i[_]+=p*q*p*a;l[_]+=p*q*q*a*q;
»       »       n[_]+=p*a*p*q*p;m[_]+=q*p*s*q;
»       }
}
```

What does this code do ?



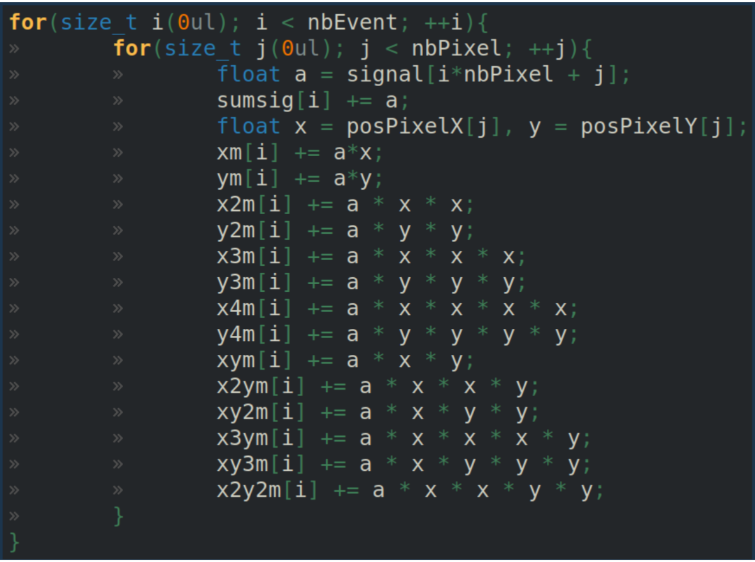
```
float a, q, p, r, s;
for(size_t _(0ul);_<m_;++_){
»       a_[_]=0;
»       b[_]=0;c[_]=0;d[_]=0;e[_]=0;f[_]=0;g[_]=0;h[_]=0;
»       i[_]=0;j[_]=0:k[_]=0;l[_]=0;m[_]=0;n[_]=0;o[_]=0;
»       for(size_t __(0ul);__<n;++__){
»       »       a=w[_*n+__];a_[_]+=a;q=u[__],p=v[__];r=a*q;s=a*p;
»       »       b[_]+=r;c[_]+=s;d[_]+=q*r;f[_]+=s*p;
»       »       g[_]+=q*q*a*q;j[_]+=s*p*p;
»       »       k[_]+=r*q*q*q;o[_]+=p*p*a*p*p;
»       »       e[_]+=r*p;h[_]+=q*a*q*p;
»       »       i[_]+=p*q*p*a;l[_]+=p*q*q*a*q;
»       »       n[_]+=p*a*p*q*p;m[_]+=q*p*s*q;
»       }
}
```

What does this code do ?



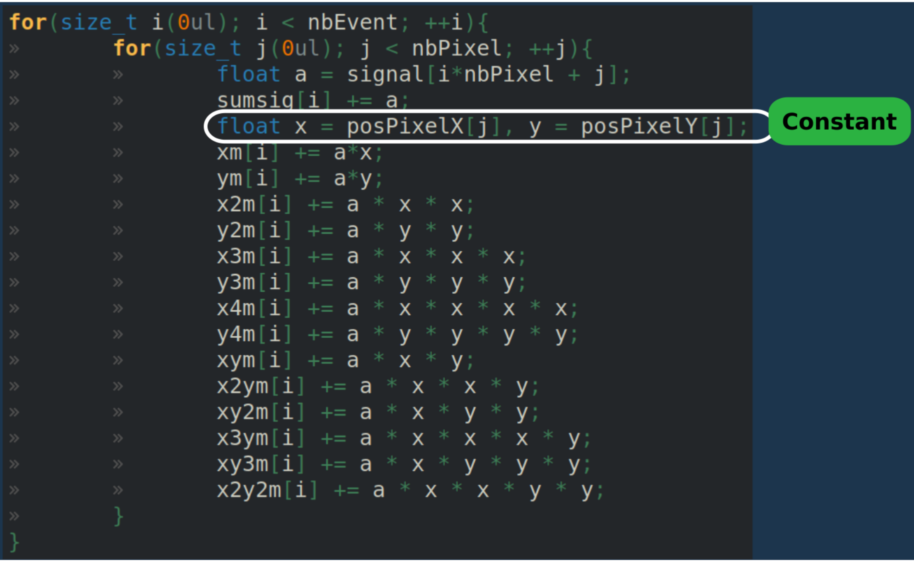
```
for(size_t i(0ul); i < nbEvent; ++i){
»       sumsig[i] = 0.0f;
»       xm[i] = 0.0f;ym[i] = 0.0f;x2m[i] = 0.0f;xym[i] = 0.0f;
»       y2m[i] = 0.0f;x3m[i] = 0.0f;x2ym[i] = 0.0f;xy2m[i] = 0.0f;
»       y3m[i] = 0.0f:x4m[i] = 0.0f;x3ym[i] = 0.0f;x2y2m[i] = 0.0f;
»       xy3m[i] = 0.0f;y4m[i] = 0.0f;
»       for(size_t j(0ul); j < nbPixel; ++j){
»       »       float a = signal[i*nbPixel + j];
»       »       sumsig[i] += a;
»       »       float x = posPixelX[j], y = posPixelY[j];
»       »       xm[i] += a*x;ym[i] += a*y;
»       »       x2m[i] += a * x * x;y2m[i] += a * y * y;
»       »       x3m[i] += a * x * x * x;y3m[i] += a * y * y * y;
»       »       x4m[i] += a * x * x * x * x;y4m[i] += a * y * y * y * y;
»       »       xym[i] += a * x * y;x2ym[i] += a * x * x * y;
»       »       xy2m[i] += a * x * y * y;x3ym[i] += a * x * x * x * y;
»       »       xy3m[i] += a * x * y * y * y;x2y2m[i] += a * x * x * y * y;
»       }
}
```

What does this code do ?



```
for(size_t i(0ul); i < nbEvent; ++i){
        for(size_t j(0ul); j < nbPixel; ++j){
                float a = signal[i*nbPixel + j];
                sumsig[i] += a;
                float x = posPixelX[j], y = posPixelY[j];
                xm[i] += a*x;
                ym[i] += a*y;
                x2m[i] += a * x * x;
                y2m[i] += a * y * y;
                x3m[i] += a * x * x * x;
                y3m[i] += a * y * y * y;
                x4m[i] += a * x * x * x * x;
                y4m[i] += a * y * y * y * y;
                xym[i] += a * x * y;
                x2ym[i] += a * x * x * y;
                xy2m[i] += a * x * y * y;
                x3ym[i] += a * x * x * x * y;
                xy3m[i] += a * x * y * y * y;
                x2y2m[i] += a * x * x * y * y;
        }
}
```

What does this code do ?



```
for(size_t i(0ul); i < nbEvent; ++i){
	for(size_t j(0ul); j < nbPixel; ++j){
		float a = signal[i*nbPixel + j];
		sumsig[i] += a;
		float x = posPixelX[j], y = posPixelY[j];
		xm[i] += a*x;
		ym[i] += a*y;
		x2m[i] += a * x * x;
		y2m[i] += a * y * y;
		x3m[i] += a * x * x * x;
		y3m[i] += a * y * y * y;
		x4m[i] += a * x * x * x * x;
		y4m[i] += a * y * y * y * y;
		xym[i] += a * x * y;
		x2ym[i] += a * x * x * y;
		xy2m[i] += a * x * y * y;
		x3ym[i] += a * x * x * x * y;
		xy3m[i] += a * x * y * y * y;
		x2y2m[i] += a * x * x * y * y;
	}
}
```

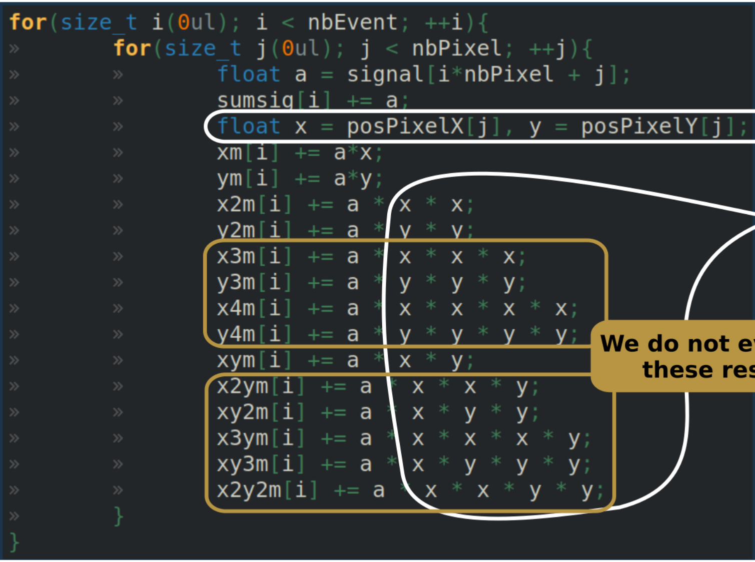
What does this code do ?

```
for(size_t i(0ul); i < nbEvent; ++i){
	for(size_t j(0ul); j < nbPixel; ++j){
		float a = signal[i*nbPixel + j];
		sumsig[i] += a;
		float x = posPixelX[j], y = posPixelY[j];
		xm[i] += a*x;
		ym[i] += a*y;
		x2m[i] += a * x * x;
		y2m[i] += a * y * y;
		x3m[i] += a * x * x * x;
		y3m[i] += a * y * y * y;
		x4m[i] += a * x * x * x * x;
		y4m[i] += a * y * y * y * y;
		xym[i] += a * x * y;
		x2ym[i] += a * x * x * y;
		xy2m[i] += a * x * y * y;
		x3ym[i] += a * x * x * x * y;
		xy3m[i] += a * x * y * y * y;
		x2y2m[i] += a * x * x * y * y;
	}
}
```

Constant

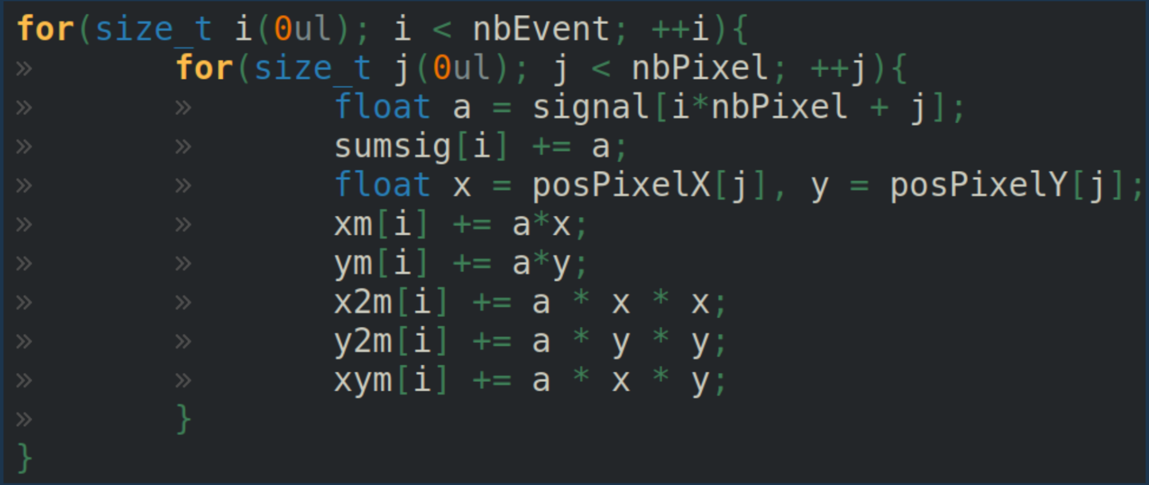
Useless computation

What does this code do ?



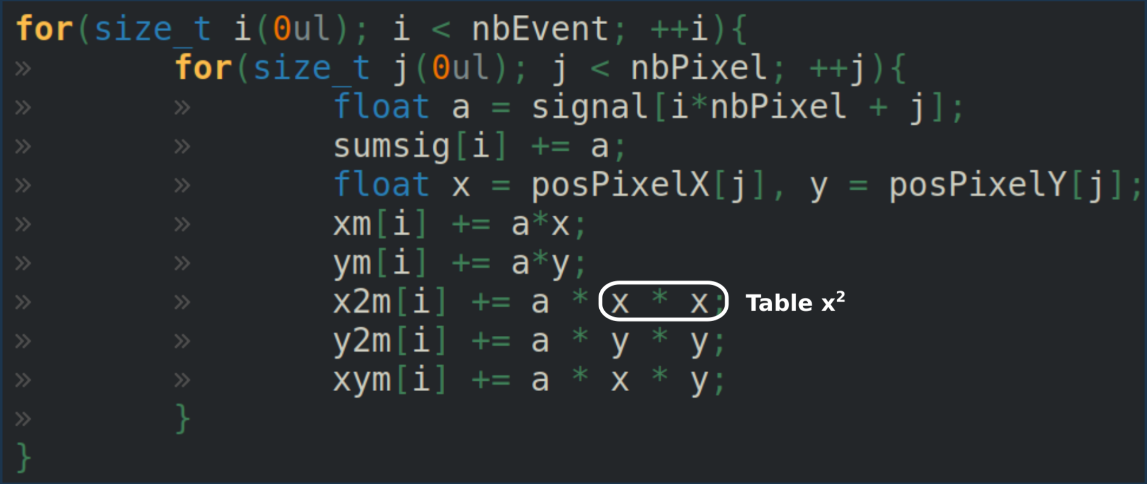
```
for(size_t i(0ul); i < nbEvent; ++i){
	for(size_t j(0ul); j < nbPixel; ++j){
		float a = signal[i*nbPixel + j];
		sumsig[i] += a;
		float x = posPixelX[j], y = posPixelY[j];
		xm[i] += a*x;
		ym[i] += a*y;
		x2m[i] += a * x * x;
		y2m[i] += a * y * y;
		x3m[i] += a * x * x * x;
		y3m[i] += a * y * y * y;
		x4m[i] += a * x * x * x * x;
		y4m[i] += a * y * y * y * y;
		xym[i] += a * x * y;
		x2ym[i] += a * x * x * y;
		xy2m[i] += a * x * y * y;
		x3ym[i] += a * x * x * x * y;
		xy3m[i] += a * x * y * y * y;
		x2y2m[i] += a * x * x * y * y;
	}
}
```

What does this code do ?



```
for(size_t i(0ul); i < nbEvent; ++i){
        for(size_t j(0ul); j < nbPixel; ++j){
                float a = signal[i*nbPixel + j];
                sumsig[i] += a;
                float x = posPixelX[j], y = posPixelY[j];
                xm[i] += a*x;
                ym[i] += a*y;
                x2m[i] += a * x * x;
                y2m[i] += a * y * y;
                xym[i] += a * x * y;
        }
}
```

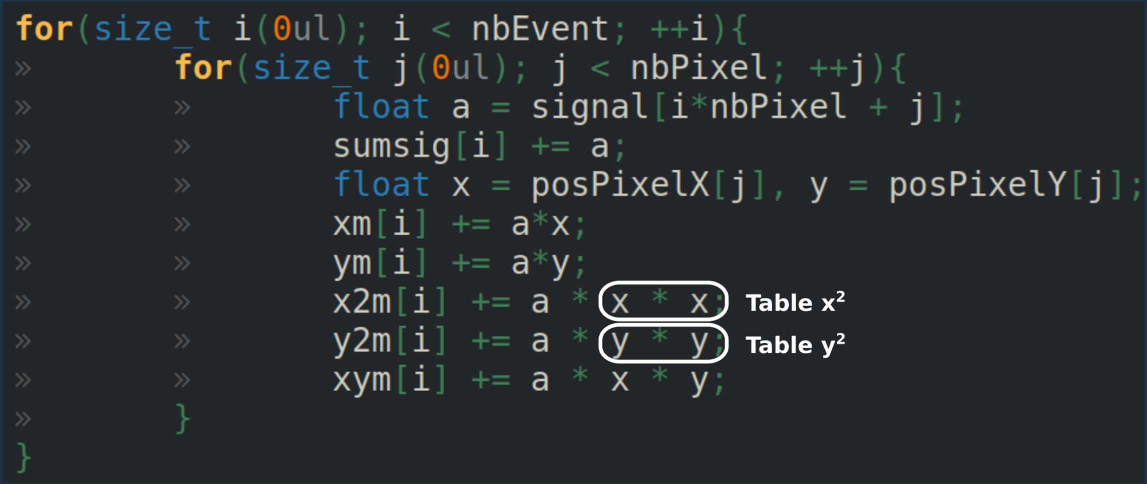
What does this code do ?



```
for(size_t i(0ul); i < nbEvent; ++i){
	for(size_t j(0ul); j < nbPixel; ++j){
		float a = signal[i*nbPixel + j];
		sumsig[i] += a;
		float x = posPixelX[j], y = posPixelY[j];
		xm[i] += a*x;
		ym[i] += a*y;
		x2m[i] += a * x * x;
		y2m[i] += a * y * y;
		xym[i] += a * x * y;
	}
}
```

Table x^2

What does this code do ?

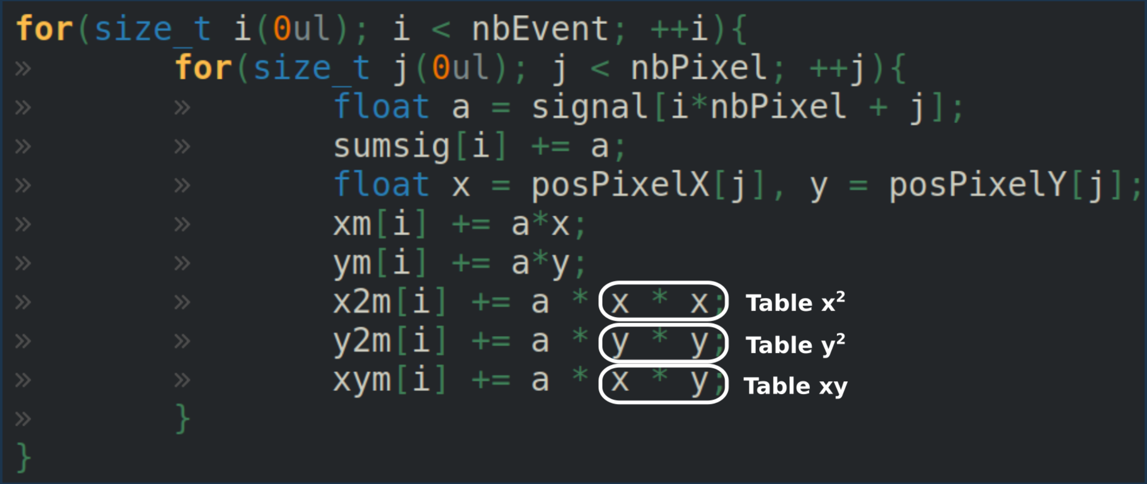


```
for(size_t i(0ul); i < nbEvent; ++i){
	for(size_t j(0ul); j < nbPixel; ++j){
		float a = signal[i*nbPixel + j];
		sumsig[i] += a;
		float x = posPixelX[j], y = posPixelY[j];
		xm[i] += a*x;
		ym[i] += a*y;
		x2m[i] += a * x * x;
		y2m[i] += a * y * y;
		xym[i] += a * x * y;
	}
}
```

Table x^2

Table y^2

What does this code do ?



```
for(size_t i(0ul); i < nbEvent; ++i){
	for(size_t j(0ul); j < nbPixel; ++j){
		float a = signal[i*nbPixel + j];
		sumsig[i] += a;
		float x = posPixelX[j], y = posPixelY[j];
		xm[i] += a*x;
		ym[i] += a*y;
		x2m[i] += a * x * x;
		y2m[i] += a * y * y;
		xym[i] += a * x * y;
	}
}
```

What does this code do ?

```
for(size_t i(0ul); i < nbEvent; ++i){
	for(size_t j(0ul); j < nbPixel; ++j){
		float a = signal[i*nbPixel + j];
		sumsig[i] += a;
		float x = posPixelX[j], y = posPixelY[j];
		xm[i] += a*x;
		ym[i] += a*y;
		x2m[i] += a * x * x;
		y2m[i] += a * y * y;
		xym[i] += a * x * y;
	}
}
```

Table x^2

Table y^2

Table xy

These are dot products

What does this code do ?

```cpp
for(size_t i(0ul); i < nbEvent; ++i){
	for(size_t j(0ul); j < nbPixel; ++j){
		float a = signal[i*nbPixel + j];
		sumsig[i] += a;
		float x = posPixelX[j], y = posPixelY[j];
		xm[i] += a*x;
		ym[i] += a*y;
		x2m[i] += a * x * x;
		y2m[i] += a * y * y;
		xym[i] += a * x * y;
	}
}
```

Table x^2

Table y^2

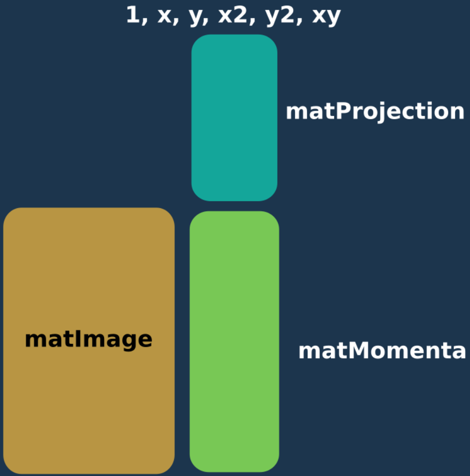
Table xy

These are dot products

This is a Matrix Multiplication

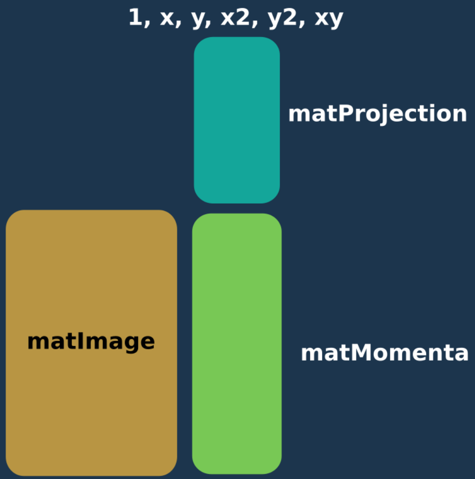
What does this code do ?

```
sgemm(matMomenta, matImage, matProjection);
```



What does this code do ?

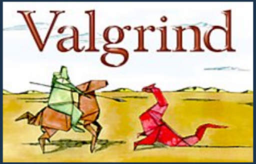
```
sgemm(matMomenta, matImage, matProjection);
```



Profiling Tools

Profiling Tools

- **Valgrind** :
  - Dynamic binary **Profiler**
  - **Memory** binary Checker



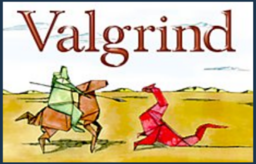
Profiling Tools

- **Valgrind** :
    - Dynamic binary **Profiler**
    - **Memory** binary Checker

- **Maqao** :
    - **Static** binary profiler
    - **Dynamic** binary profiler

Profiling Tools

- **Valgrind** :
    - Dynamic binary **Profiler**
    - **Memory** binary Checker

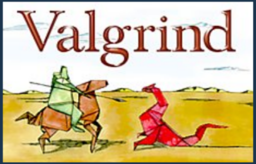


- **Maqao** :
    - **Static** binary profiler
    - **Dynamic** binary profiler

- **Auto profiling** library :
    - Only **selected functions**
    - But has to be **part of the software**

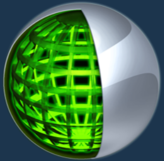
Profiling Tools

- **Valgrind** :
    - Dynamic binary **Profiler**
    - **Memory** binary Checker



- **Maqao** :
    - **Static** binary profiler
    - **Dynamic** binary profiler

- **Auto profiling** library :
    - Only **selected functions**
    - But has to be **part of the software**



Conclusion

Conclusion

git

Conclusion

Gitlab

Conclusion

Gitlab

Project builder

Unit tests

Conclusion

Gitlab

Project builder **Unit tests**

readme.md :
- Showcase
- Depedencies
- Installation / Compitation
- Use cases

Development Documentation :
- Doxygen
- Sphynx

Conclusion

Gitlab

Project builder **Unit tests**

readme.md :
- Showcase
- Depedencies
- Installation / Compitation
- Use cases

Development Documentation :
- Doxygen
- Sphynx

Profiling

NSight

Conclusion

git

Gitlab

Project builder **Unit tests**

CMake
Cross-platform Make

readme.md :
- Showcase
- Depedencies
- Installation / Compitation
- Use cases

Development Documentation :
- Doxygen
- Sphynx

doxygen

Implement **small functions**

Profiling

Valgrind

MAQAO

NSight