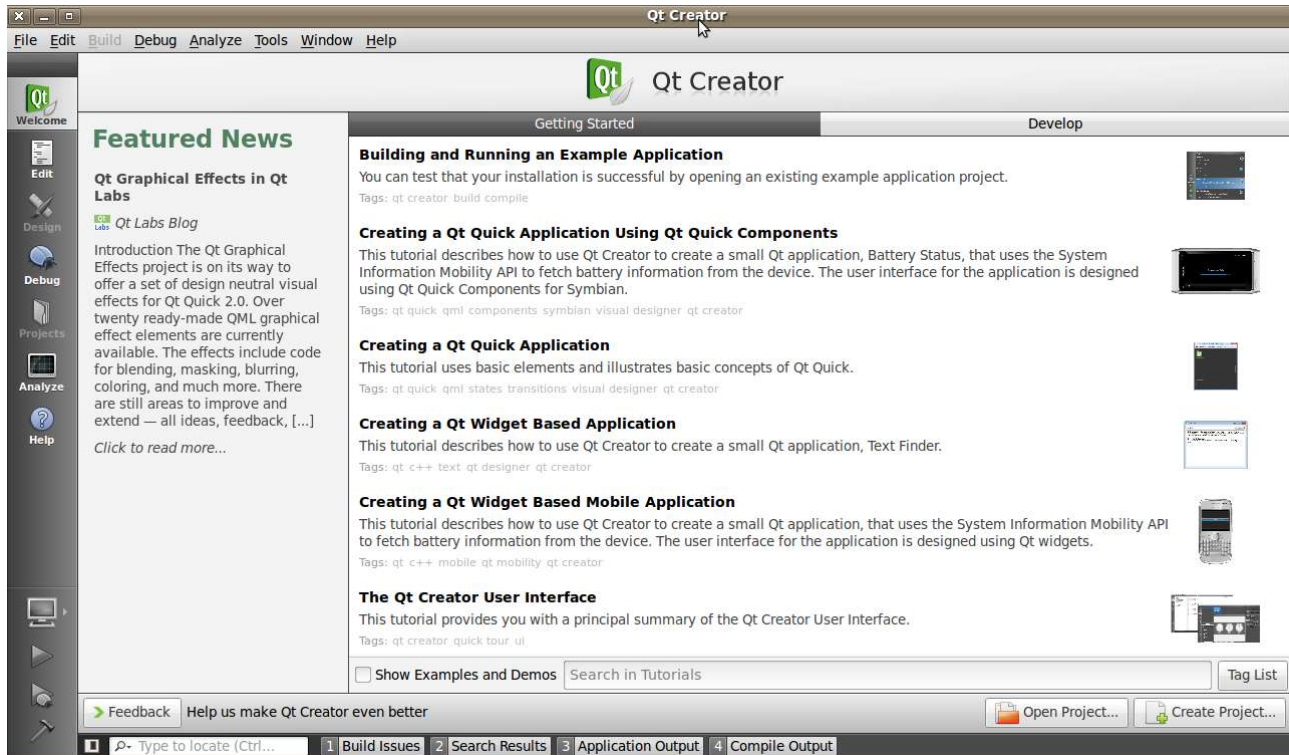


## GiVD - curs 2013-2014 – Exercici C++ i QtCreator

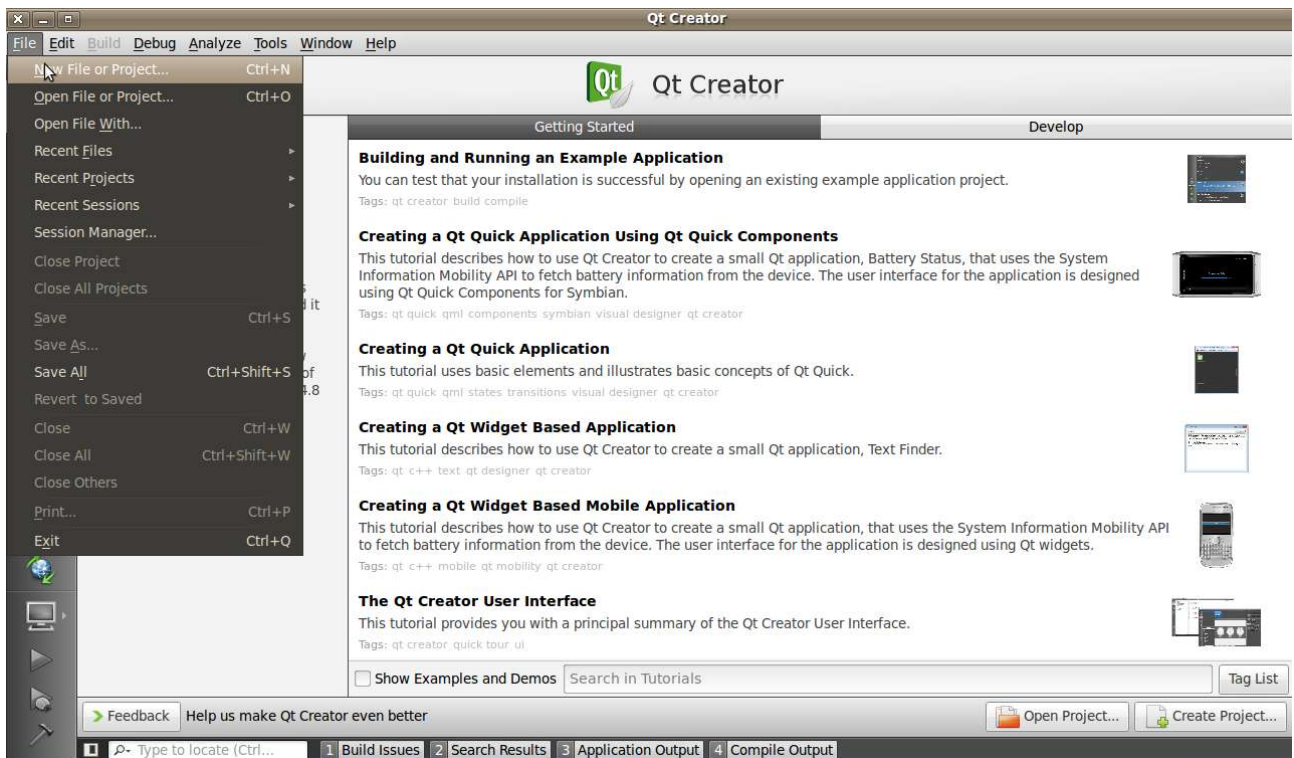
Per a executar el QtCreator heu d'anar a Aplicacions->Eines de programació i executar el QtCreator. Us sortirà una pantalla de l'estil:



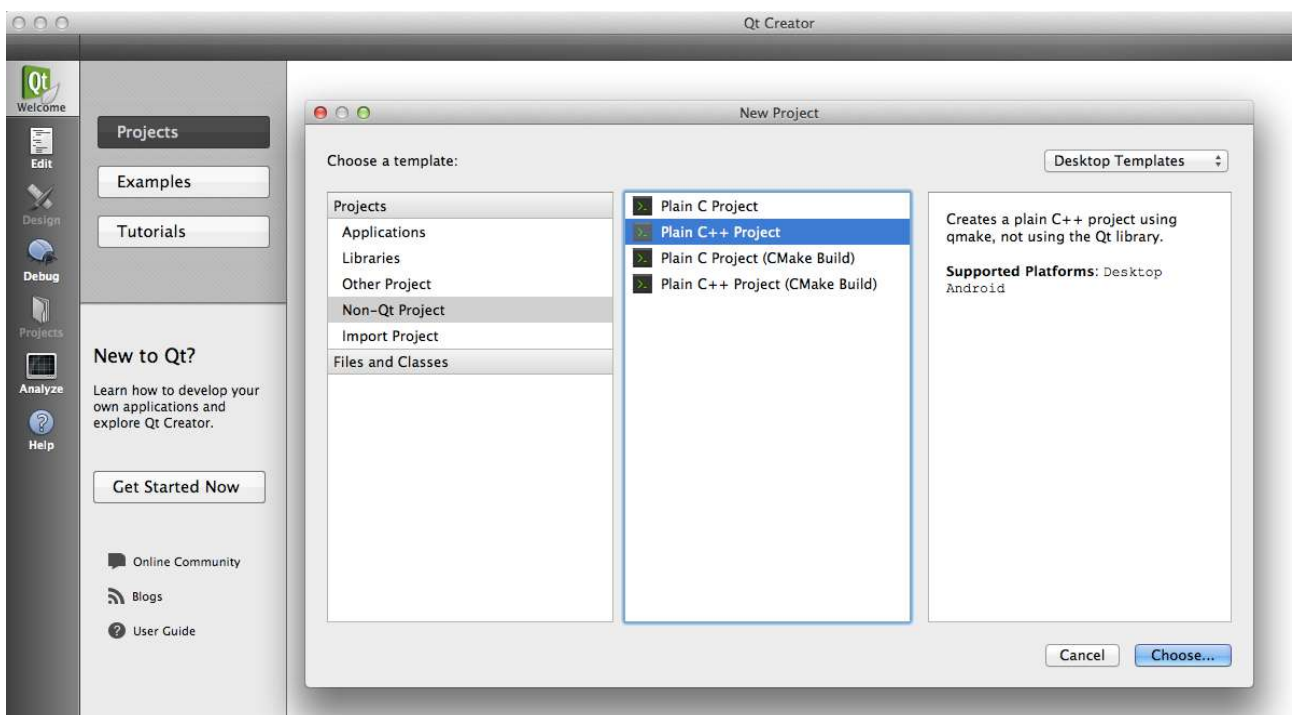
Des del mateixa plana principal de QtCreator podeu repassar alguns tutorials per introduir-vos amb més detall en aquest IDE. A continuació, fem una repassada de com començar un nou projecte, del tipus QtApplication i farem un primer HelloQt utilitzant un exemple ja creat amb Qt i OpenGL.

Fixeu-vos que a la part esquerra teniu diferents botons: Welcome (activa els tutorials bàsics per a començar a treballar amb QtCreator), Edit (activa el mode d'edició del codi), Design (activa el QtDesigner que serveix per a dissenyar interfícies gràfiques) i Debug (que serveix per entrar a depurar el codi que esteu desenvolupant). El botó de Help activa tot el llistat de suport per a desenvolupar projectes i per a accedir als principals exemples de codi.

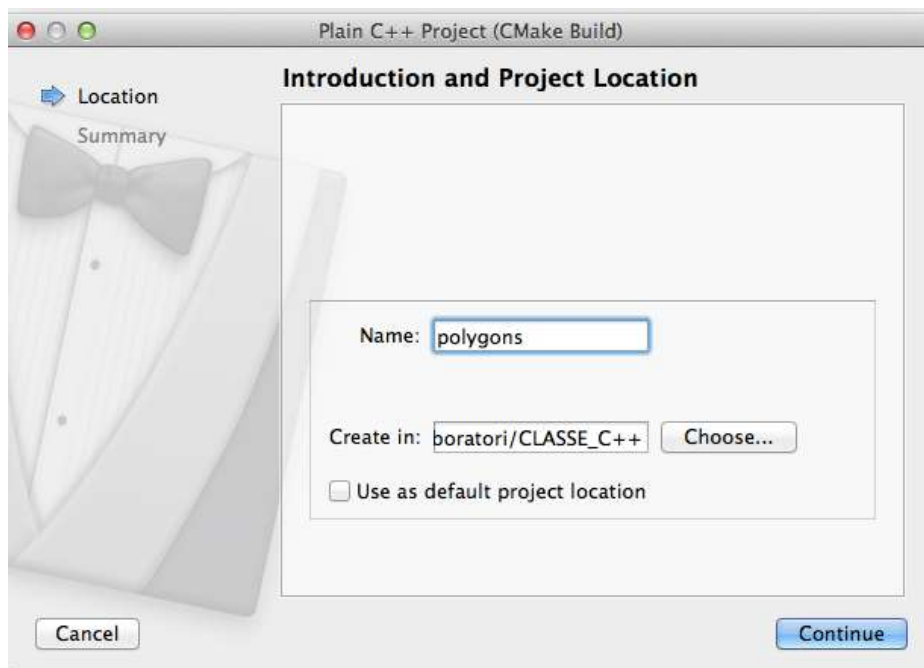
Per a començar un projecte buit picarem al menú **File->New Project** de la part superior esquerra i anirem seguint els passos que ens va indicant l'IDE.



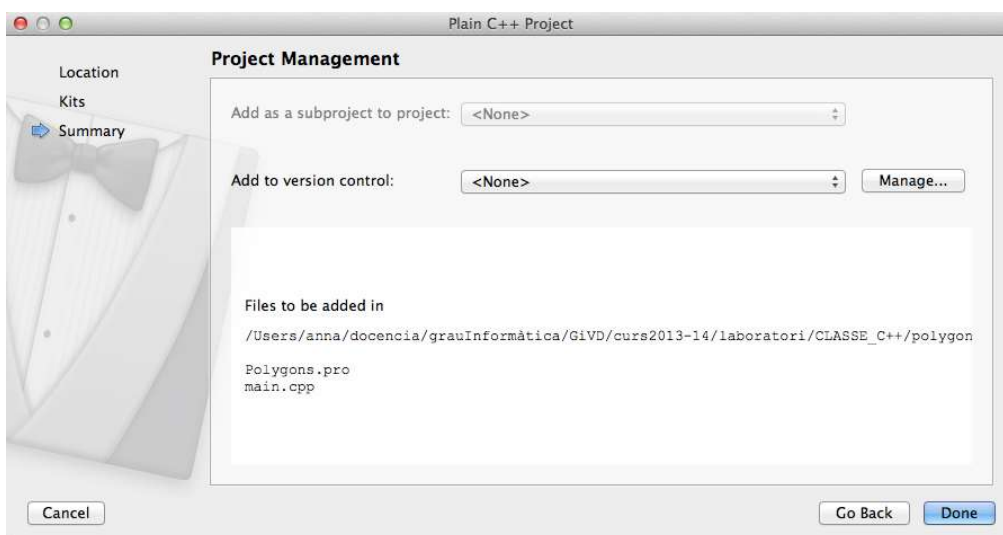
Aquí, seleccionarem de fer una aplicació de C++.



A partir d'aquesta selecció, ens demanarà on volem guardar el projecte i quines versions usarem de Qt (si en tenim més d'una instal·lada) i si volem fer la versió de Release o de Debug o ambdues. Començarem donant el nom del projecte nou, que l'anomenem Polygons, i generarem la versió amb Qt4.8.0

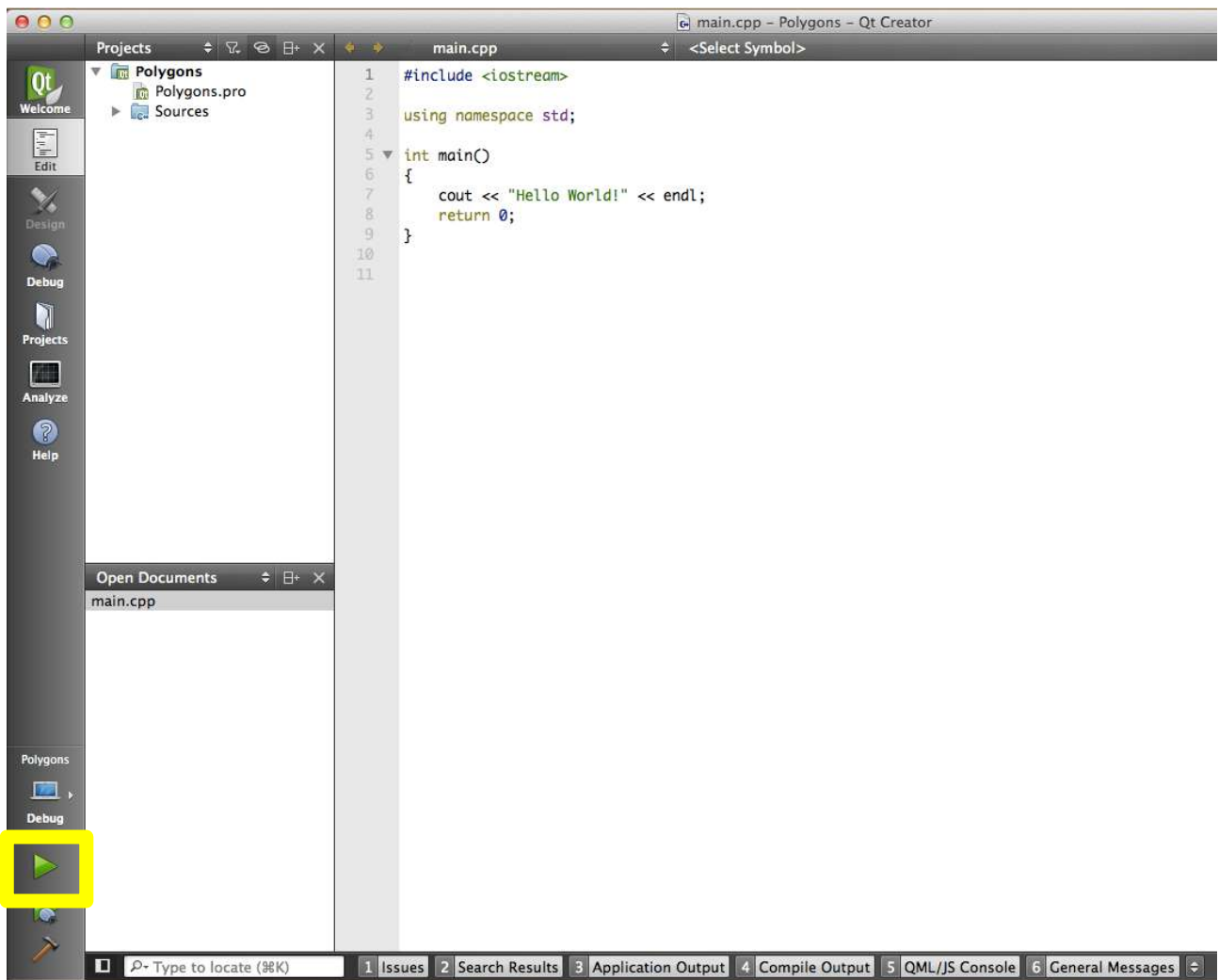


Com a detall adicional, ens diu quina es la classe que contindrà el *main* de l'aplicació.



```
main.cpp
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      cout << "Hello World!" << endl;
8      return 0;
9  }
10
11
```

Per a compilar i executar, heu de premer el butó de "Play".



El resultat sortirà per consola:

```
anna — qtcreator_proces — 80x24
Last login: Fri Feb 14 20:52:27 on ttys002
cd /; /opt/Qt/Qt\ Creator.app/Contents/MacOS/../Resources/qtcreator_process_stub
run /var/folders/kt/klcq2452nq3wzx2vqgd8j9h0000gn/T/QtCreator.t94555/stub-sock
et Press\ \<RETURN>\ to\ close\ this\ window... /Users/anna/build-prova2-Deskt
p_Qt_5_2_0_clang_64bit-Debug /var/folders/kt/klcq2452nq3wzx2vqgd8j9h0000gn/T/Qt
Creator.C94555 /Users/anna/build-prova2-Desktop_Qt_5_2_0_clang_64bit-Debug/prova
2
MacBook-Air-de-Anna:~ anna$ cd /; /opt/Qt/Qt\ Creator.app/Contents/MacOS/../Reso
urces/qtcreator_process_stub run /var/folders/kt/klcq2452nq3wzx2vqgd8j9h0000gn/
T/QtCreator.t94555/stub-socket Press\ \<RETURN>\ to\ close\ this\ window... /Us
ers/anna/build-prova2-Desktop_Qt_5_2_0_clang_64bit-Debug /var/folders/kt/klcq24
52nq3wzx2vqgd8j9h0000gn/T/QtCreator.C94555 /Users/anna/build-prova2-Desktop_Qt_5
_2_0_clang_64bit-Debug/prova2
Hello World!
Press <RETURN> to close this window...
```

**Exercici 1:** Afegeix una classe Square en el projecte amb un mètode anomenat `getPerimeter()` i:

- Afegeix una classe Figure al projecte. Aquesta ha de tenir un mètode abstracte `void getPerimeter()` que retorna el perímetre de la figura.
- Fes que la classe Square hereti de la classe Figure.
- Afegeix una classe Circle al projecte, que també ha d'heretar de Figure.
- En el main crea objectes Square i Circle i obté els seus perímetres.

**Exercici 2:**

- Afegeix al projecte una classe FigureContainer. Aquesta ha de tenir:
  - Un `std::vector` privat de punters a objectes Figure.
  - Un mètode públic **`void addFigure (Figure*)`** que afegirà al vector la figura que se li passi per paràmetre.
  - Un mètode públic **`double getPerimeters ()`** que retornarà la suma dels perímetres de les figures contingudes en el vector.
- En el main crea un objecte FigureContainer i afegeix els Square i Circle creats prèviament, obté la suma dels seus perímetres.

**Exercici 3:**

- Afegeix un mètode **`void setRadius (double)`** a la classe Circle, que canviarà el valor del radi de l'objecte Circle.
- En el main, després d'obtenir la suma dels perímetres de les figures en FigureContainer, canvia el radi de l'objecte Circle ja creat i torna a obtenir la suma dels perímetres afegits al contenidor de figures. Observa el resultat executant-lo en l'entorn de QtCreator.
- Ressegueix amb el debugger els diferents passos de l'execució posant un breakpoint al mètode `setRadius`.