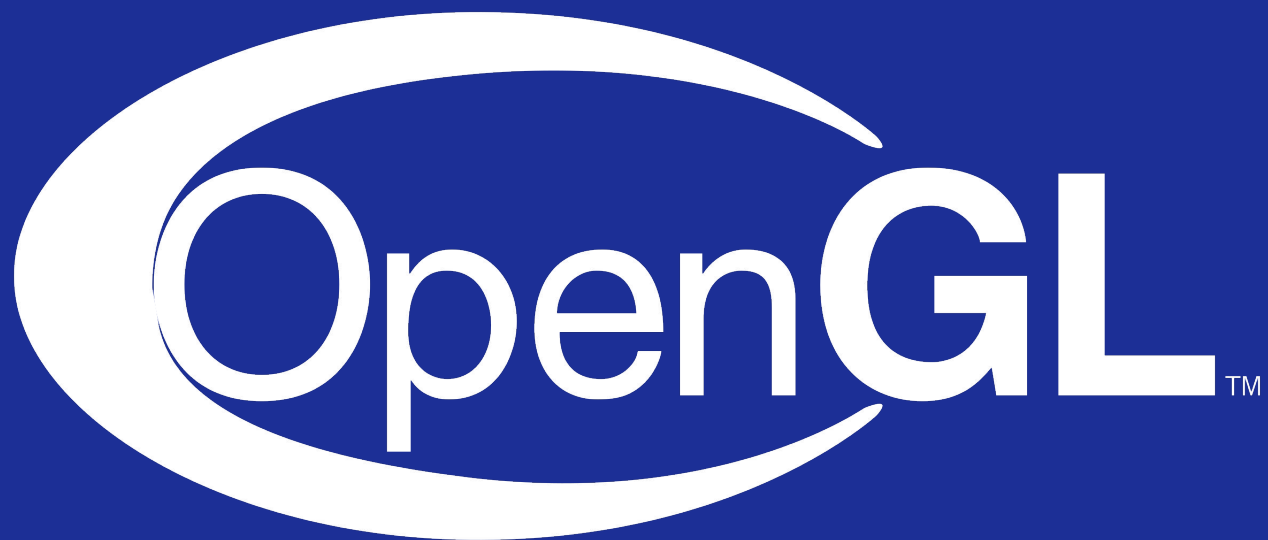




Zachodniopomorski  
Uniwersytet Technologiczny  
w Szczecinie

# Gry Komputerowe Laboratorium 1

## Zajęcia organizacyjne Animacja z uwzględnieniem czasu



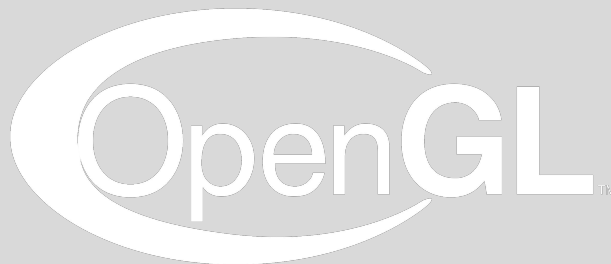
Wydział  
Informatyki

mgr inż. Michał Chwesiuk



Zachodniopomorski  
Uniwersytet Technologiczny  
w Szczecinie

# Plan zajęć projektowych



## Plan zajęć

Konfiguracja

Animacja z  
uwzględnieniem czasu

- Zajęcia projektowe składają się z 15 godzin, co daje nam 7 zajęć (plus jedno zajęcie godzinne).
- Zajęcia polegają na nauce podstawowych technik programowania grafiki czasu rzeczywistego w OpenGL.
- Celem zajęć jest implementacja własnej gry komputerowej korzystając z poznanych technik.
- Na każdym zajęciach będzie do wykonania zadanie, które ma pomóc w implementacji gry. Łącznie będzie omówione 6 tematów.
- Wymagana frekwencja na przyzwoitym poziomie.
- **Warunkiem koniecznym do otrzymania oceny pozytywnej z projektu jest oddanie gry!**



Wydział  
Informatyki



Zachodniopomorski  
Uniwersytet Technologiczny  
w Szczecinie

# Plan zajęć projektowych

## Plan zajęć

Konfiguracja

Animacja z  
uwzględnieniem czasu



- **Warunkiem koniecznym do otrzymania oceny pozytywnej z projektu jest oddanie gry!**
- Pod koniec semestru należy oddać paczkę zawierającą kody źródłowe (bez zbędnych plików), oraz skompilowaną wersję aplikacji.
- Gra musi wykorzystywać **OpenGL**, oraz być napisana w C++ w środowisku Qt Creator
- Korzystanie z frameworków, bibliotek wpływających na strukturę projektu oraz wszelkich gotowych silników jest zakazane. Silnik należy zaimplementować własnoręcznie.
- Korzystanie z bibliotek do obsługi dźwięku oraz silników fizycznych z kolei będzie mile widziane.



Wydział  
Informatyki



Zachodniopomorski  
Uniwersytet Technologiczny  
w Szczecinie

# Plan zajęć projektowych

## Plan zajęć

Konfiguracja

Animacja z  
uwzględnieniem czasu



- **Gra musi zawierać scenę 3D, teksturowanie, oświetlenie, interakcje z graczem, animacje/ruch obiektów na scenie i wykrywanie kolizji.**
- Koniecznym jest aby gra była grywalna, to znaczy żeby zasady rozgrywki były jasno określone i występowały możliwości wygranej i przegranej (w uzasadnionych przypadkach nie jest to konieczne).
- **Gra musi być wynikiem pracy indywidualnej!**
- Koncepcja gry musi być przedstawiona najpóźniej na trzecich laboratoriach.
- Bardzo mile widziana jest kreatywność! Labirynty i kolejne klony gier, tj. mario, space invaders pojawiają się w ilościach hurtowych co roku. Każde urozmaicenie rozgrywki skutkuje większą zabawą podczas pracy oraz lepszą oceną.



Wydział  
Informatyki



Zachodniopomorski  
Uniwersytet Technologiczny  
w Szczecinie

## Plan zajęć

Konfiguracja

Animacja z  
uwzględnieniem czasu



Wydział  
Informatyki

# Plan zajęć projektowych

- **Gry są oceniane w trygwiadzkowej skali w czterech kategoriach:**
  - **Efekty** - użyte techniki grafiki czasu rzeczywistego (np. shadery, efekty cząsteczkowe, odbicia, VBO, ...)
  - **Wygląd** - użycie własnych elementów, dopracowanie wizualne, spójność, czasochłonność prac nad wyglądem
  - **Grywalność** - logika rozgrywki, dopracowanie zasad, czy graczowi chce się grać
  - **Kod** - jakość kodu, architektura aplikacji, obiektowość
- **Punkty za uzyskane gwiazdki w kategoriach:**
  - Punkty za jedną gwiazdkę - 0.50 pkt
  - Punkty za dwie gwiazdki - 1.00 pkt
  - Punkty za trzy gwiazdki - 1.50 pkt
- **Zdobyte punkty są równoznaczne ocenie.**
- **Zaliczenie zajęć projektowych wymaga zdobycia minimum trzech punktów.**





Zachodniopomorski  
Uniwersytet Technologiczny  
w Szczecinie

# Przykład gier

## Plan zajęć

Konfiguracja

Animacja z  
uwzględnieniem czasu



Wydział  
Informatyki



Zachodniopomorski  
Uniwersytet Technologiczny  
w Szczecinie

# Konspekty Gier

## Plan zajęć

Konfiguracja

Animacja z  
uwzględnieniem czasu



- Każdy powinien przygotować pomysł na grę maksymalnie do trzecich zajęć projektowych.
- Pomysł może być przedstawiony w ramach konspektu, przedstawienia wizji na forum grupy, lub w rozmowie prywatnej z prowadzącym zajęcia.



Wydział  
Informatyki



Zachodniopomorski  
Uniwersytet Technologiczny  
w Szczecinie

## Plan zajęć

Konfiguracja

Animacja z  
uwzględnieniem czasu

# Tematy zajęć laboratoryjnych



1. Animacja z uwzględnieniem czasu rzeczywistego
2. Kamera FPP/TPP
3. Organizacja sceny, kolizje
4. Oświetlenie
5. Teksturowanie



Wydział  
Informatyki





Zachodniopomorski  
Uniwersytet Technologiczny  
w Szczecinie

Plan zajęć

### Konfiguracja

Animacja z  
uwzględnieniem czasu

# Konfiguracja i przygotowanie projektu

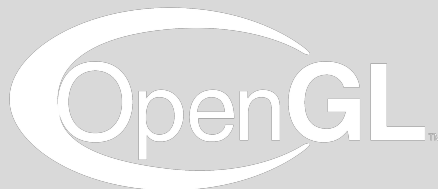


Wydział  
Informatyki



Zachodniopomorski  
Uniwersytet Technologiczny  
w Szczecinie

# OpenGL - Open Graphics Library



Plan zajęć

## Konfiguracja

Animacja z  
uwzględnieniem czasu

- **Open Graphics Library**
- **API** pozwalające na wykorzystanie akceleracji sprzętowej do **renderowania grafiki czasu rzeczywistego**.
- OpenGL **nie jest biblioteką!** Jest to zbiór funkcji i numerycznych stałych, które są interpretowane przez system operacyjny i/lub sterownik karty graficznej.
- Działanie OpenGL można rozumieć jako **maszynę stanu**, raz ustawiona wartość parametru systemu pozostaje taka sama do momentu jej zmiany.
- Główną zaletą jest **wieloplatformowość**.
- Konkurencja: Microsoft **DirectX** i Khronos Group **Vulkan**.
- **Nie obejmuje obsługi wejścia i integracji z systemem operacyjnym w celu utworzenia kontekstu.**



Wydział  
Informatyki



Zachodniopomorski  
Uniwersytet Technologiczny  
w Szczecinie

# Środowisko QT



Plan zajęć

## Konfiguracja

Animacja z  
uwzględnieniem czasu

- **Qt** - zestaw przenośnych bibliotek i narzędzi programistycznych dedykowanych dla języków **C++**, QML i Java.
- Środowisko Qt jest dostępne dla platform: X11 (m.in. GNU/Linux, BSD, Solaris), Windows, Mac OS X, Haiku oraz dla urządzeń wbudowanych opartych na Linuksie (Qt Extended), Windows CE, Symbian, Android.
- Biblioteki Qt zawierają niezależne od platformy systemowej moduły obsługi procesów, plików, sieci, grafiki trójwymiarowej (**OpenGL**), baz danych (SQL), języka XML, lokalizacji, wielowątkowości, zaawansowanej obsługi napisów oraz wtyczek.
- <https://www.qt.io>



Wydział  
Informatyki



## Plan zajęć

## Konfiguracja

## Animacja z uwzględnieniem czasu

- <https://www.qt.io>



Wydział  
Informatyki





Zachodniopomorski  
Uniwersytet Technologiczny  
w Szczecinie

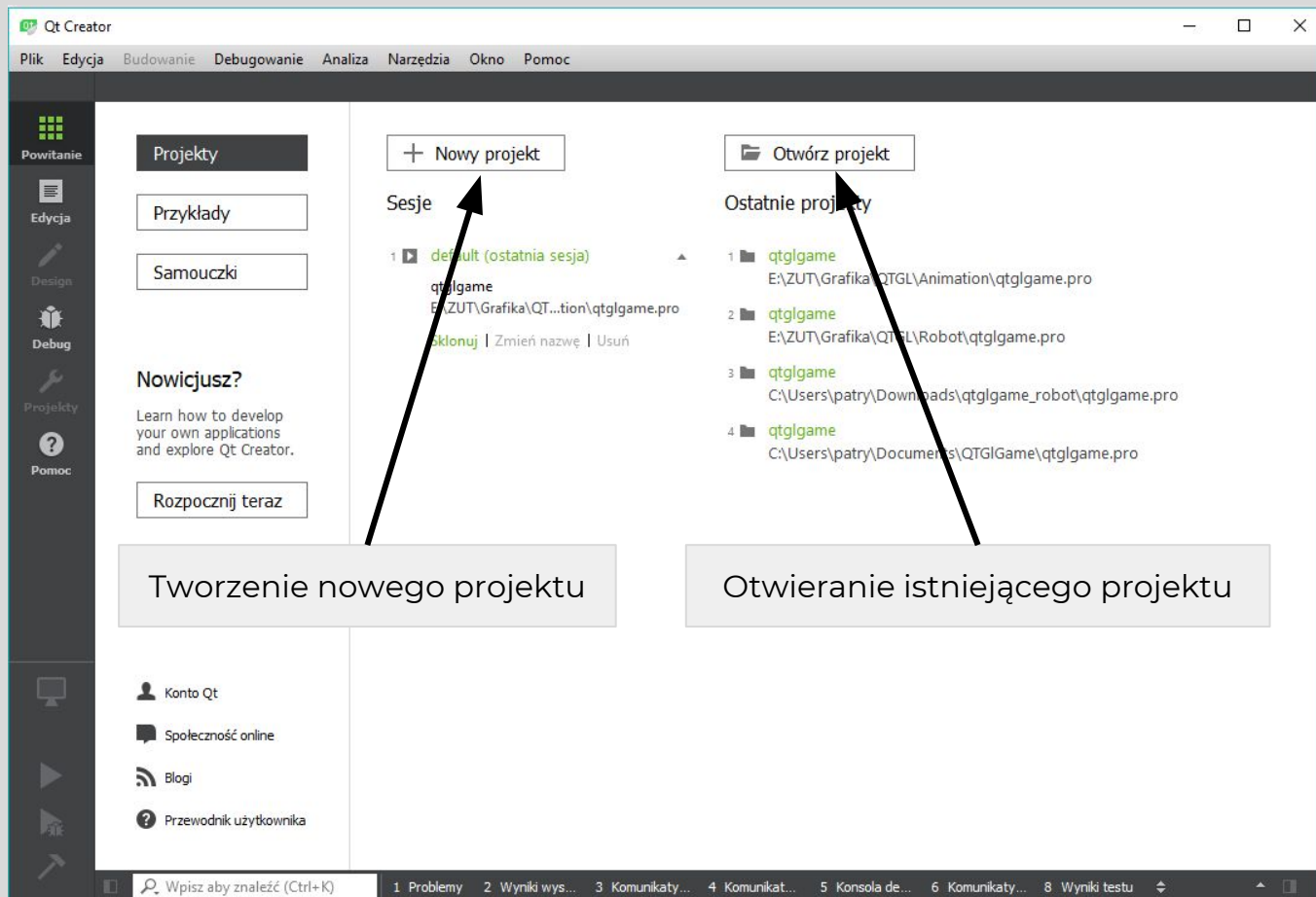
# Widok okna QT Creator

## Wczytanie projektu

Plan zajęć

Konfiguracja

Animacja z  
uwzględnieniem czasu



Wydział  
Informatyki





Zachodniopomorski  
Uniwersytet Technologiczny  
w Szczecinie

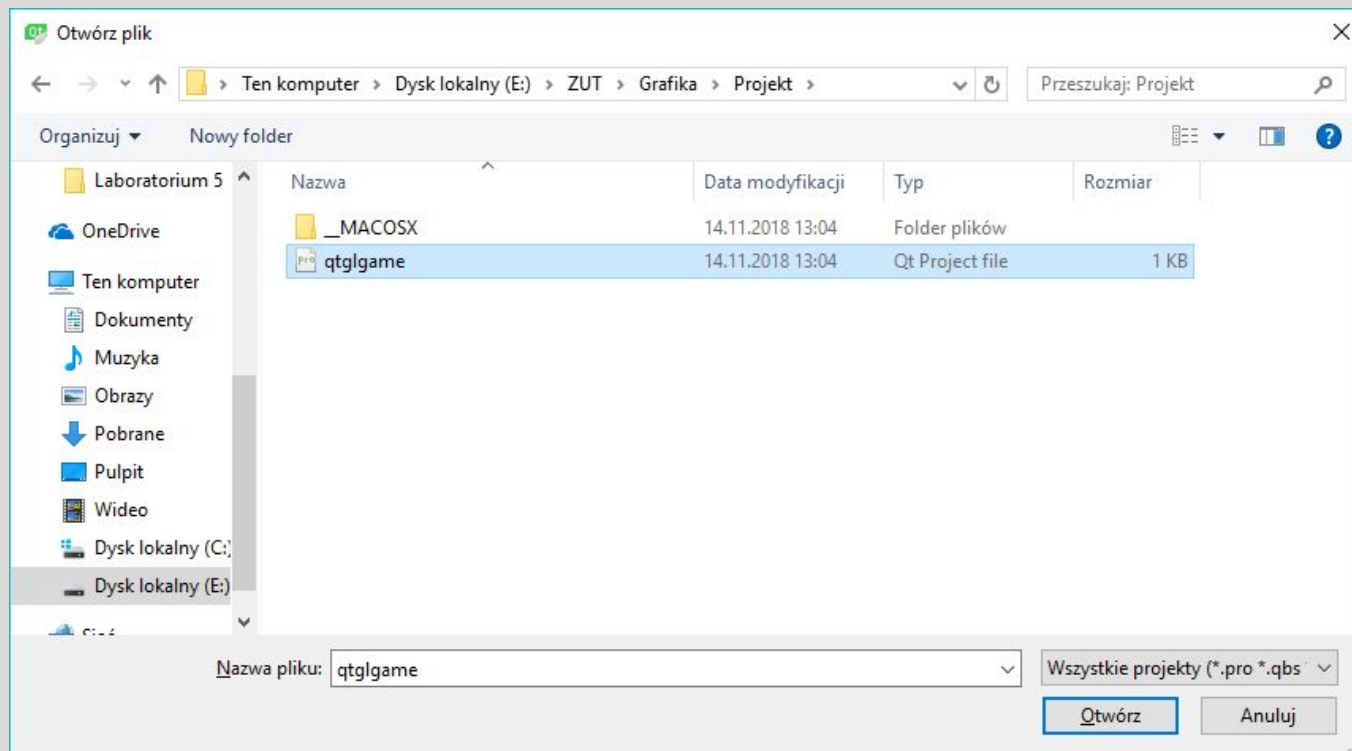
# Widok okna QT Creator

## Wczytanie projektu

Plan zajęć

### Konfiguracja

Animacja z  
uwzględnieniem czasu



- Otwórz projekt -> wybieramy plik **qtglgame.pro** z wypakowanego projektu



Wydział  
Informatyki



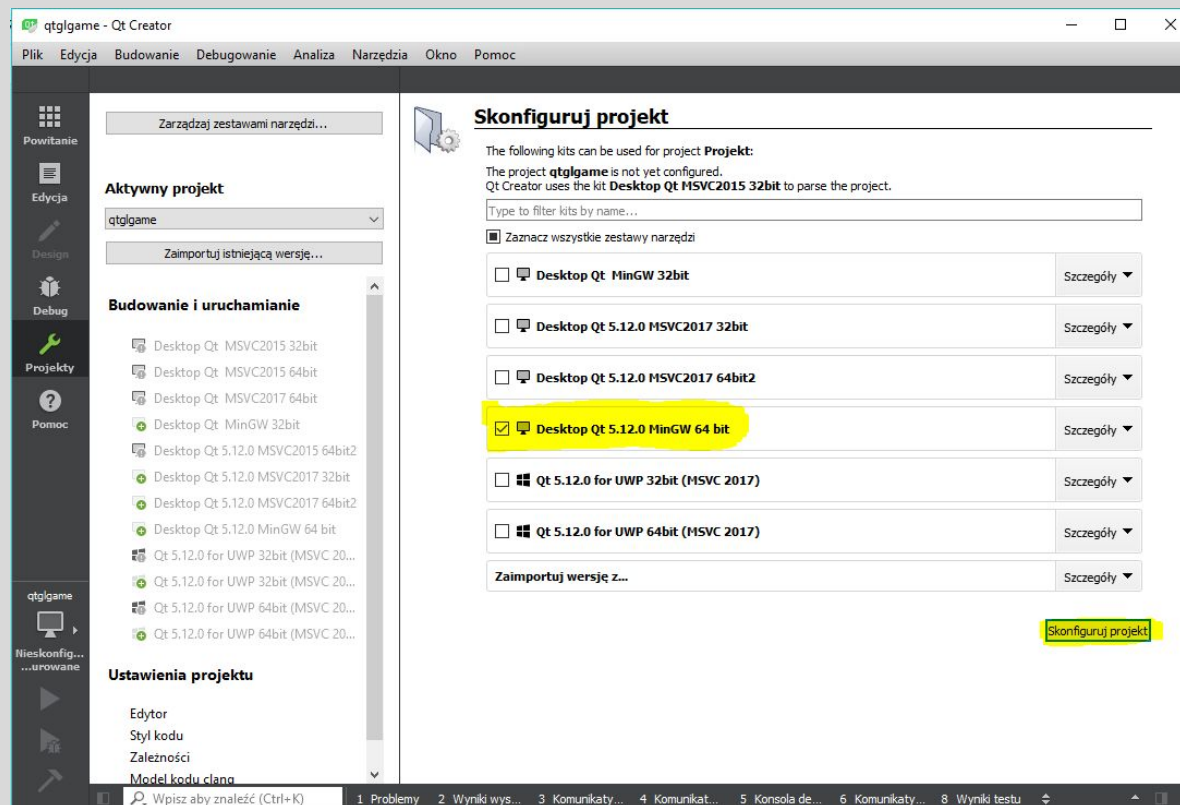
Zachodniopomorski  
Uniwersytet Techniczny  
w Szczecinie

# Widok okna QT Creator Wczytanie projektu

Plan zajęć

## Konfiguracja

Animacja z  
uwzględnieniem czasu



- Następnie należy zaznaczyć odpowiedni zestaw narzędzi Desktop GT Wersja MinGW 32/64 bit.
- Potwierdzić wybór przyciskiem *Skonfiguruj projekt.*



Wydział  
Informatyki



Zachodniopomorski  
Uniwersytet Technologiczny  
w Szczecinie

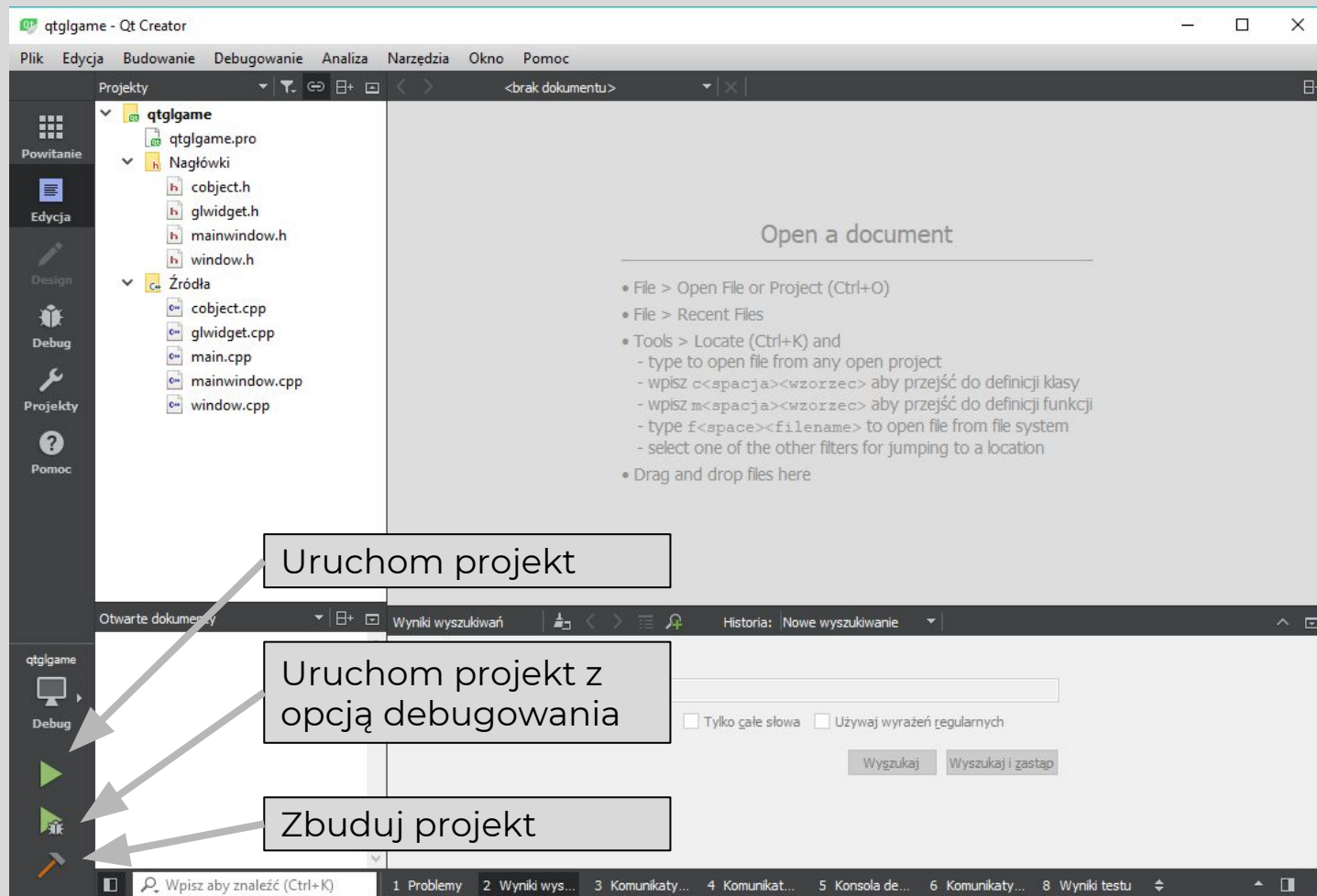
# Widok okna QT Creator

## Widok projektu

Plan zajęć

Konfiguracja

Animacja z  
uwzględnieniem czasu



Wydział  
Informatyki



Zachodniopomorski  
Uniwersytet Technologiczny  
w Szczecinie

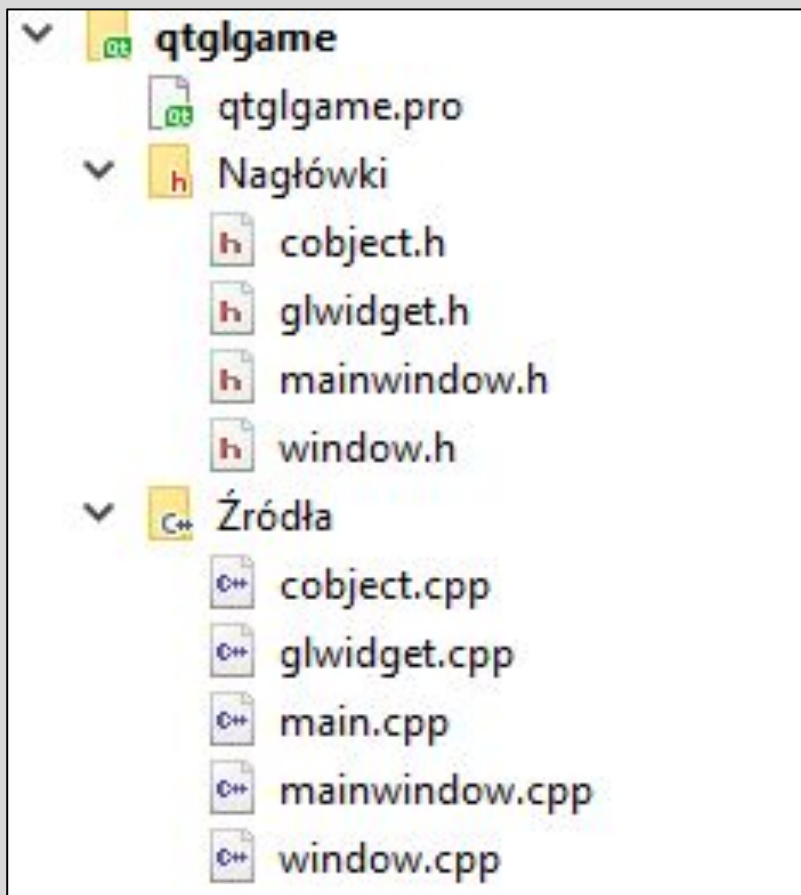
# Widok okna QT Creator

## Struktura projektu

Plan zajęć

### Konfiguracja

Animacja z  
uwzględnieniem czasu



Wydział  
Informatyki



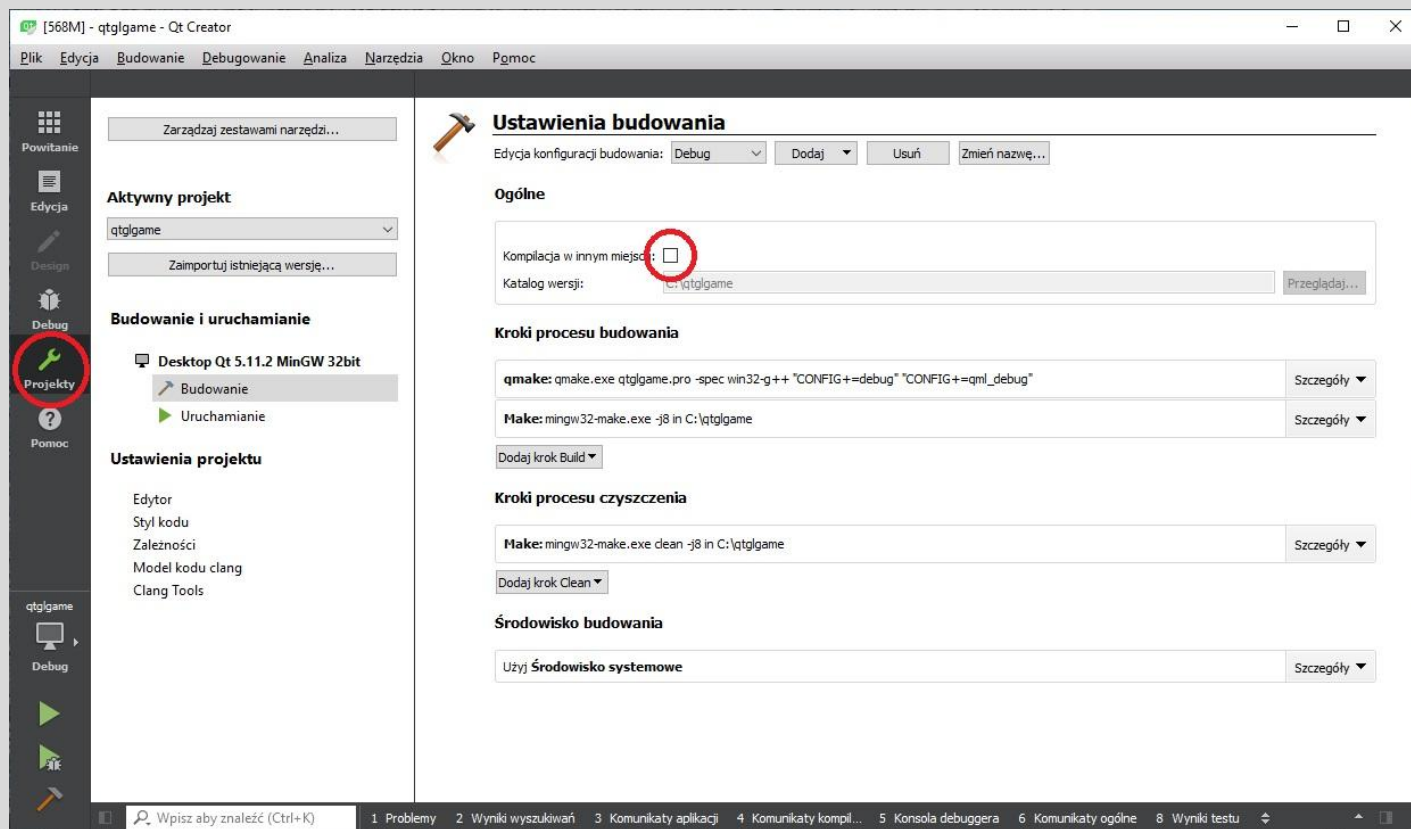
Zachodniopomorski  
Uniwersytet Techniczny  
w Szczecinie

# Przed skompilowaniem qtglgame

Plan zajęć

## Konfiguracja

Animacja z  
uwzględnieniem czasu



Wydział  
Informatyki





Zachodniopomorski  
Uniwersytet Techniczny  
w Szczecinie

Plan zajęć

Konfiguracja

**Animacja z  
uwzględnieniem czasu**

# Animacja z uwzględnieniem czasu



Wydział  
Informatyki



# Ruch ręki robota

- W klasie **GLWidget** stwórz zmienną typu *float robotArmAngle*.

```
float robotArmAngle = 0;
```

- W funkcji **GLWidget::paintGL()** zamień kąt obrotu lewej ręki robota ze stałego na zależny od stworzonej zmiennej.

```
m_world.rotate(-15.0f, 0, 0, 1); // stary  
m_world.rotate(30*sin(robotArmAngle), 0, 0, 1);
```

- Na końcu funkcji **GLWidget::paintGL()** inkrementuj zmienną *robotArmAngle* (obok *timer++*)

```
robotArmAngle++;
```

- W wyniku robot powinien dynamicznie ruszać ręką.
- W załączonym przykładzie ruch ręki nie jest zależny od czasu, nie jesteśmy w stanie ustalić szybkości kątowej ruchu.
- Wynika to z faktu że rysowanie i update świata jest wykonywany jednocześnie. **Jest to błędne podejście.**

Plan zajęć

Konfiguracja

[Animacja z  
uwzględnieniem czasu](#)





# Funkcja aktualizacji świata - Update()

- W klasie **GLWidget** stwórz metodę *void updateGL()*

```
void updateGL();
```

- Zmień w klasie **GLWidget** *int m\_timer* na *QElapsedTimer timer*, oraz stwórz zmienną *float lastUpdateTime* (wymagany `#include <QElapsedTimer>`), także zmienną *float FPS*.

```
QElapsedTimer timer;  
float lastUpdateTime;  
float FPS;
```

- Na końcu funkcji **GLWidget::initializeGL()** zamień *timer = 0*.

```
lastUpdateTime = 0;  
timer.start();  
FPS = 60;
```

- W pliku **glwidget.cpp** stwórz funkcję ciało funkcji *void GLWidget::updateGL()* i w środku inkrementuj *robotArmAngle*.

```
void GLWidget::updateGL()  
{  
    robotArmAngle = robotArmAngle + 1;  
}
```

Plan zajęć

Konfiguracja

[Animacja z  
uwzględnieniem czasu](#)





# Funkcja aktualizacji świata - Update()

- W metodzie **GLWidget::paintGL()** zastąp dwie linijki

```
m_timer++;  
robotArmAngle++;
```

- poniższym kodem :

```
float timerTime = timer.elapsed() * 0.001f;  
float deltaTime = timerTime - lastUpdateTime;  
if(deltaTime >= (1.0f / FPS))  
{  
    updateGL();  
    lastUpdateTime = timerTime;  
}
```

- Dzięki zastosowanym zmianom, funkcja Update() będzie uruchamiana 60 razy na sekundę, co daje nam większą kontrolę nad aktualizacją świata.