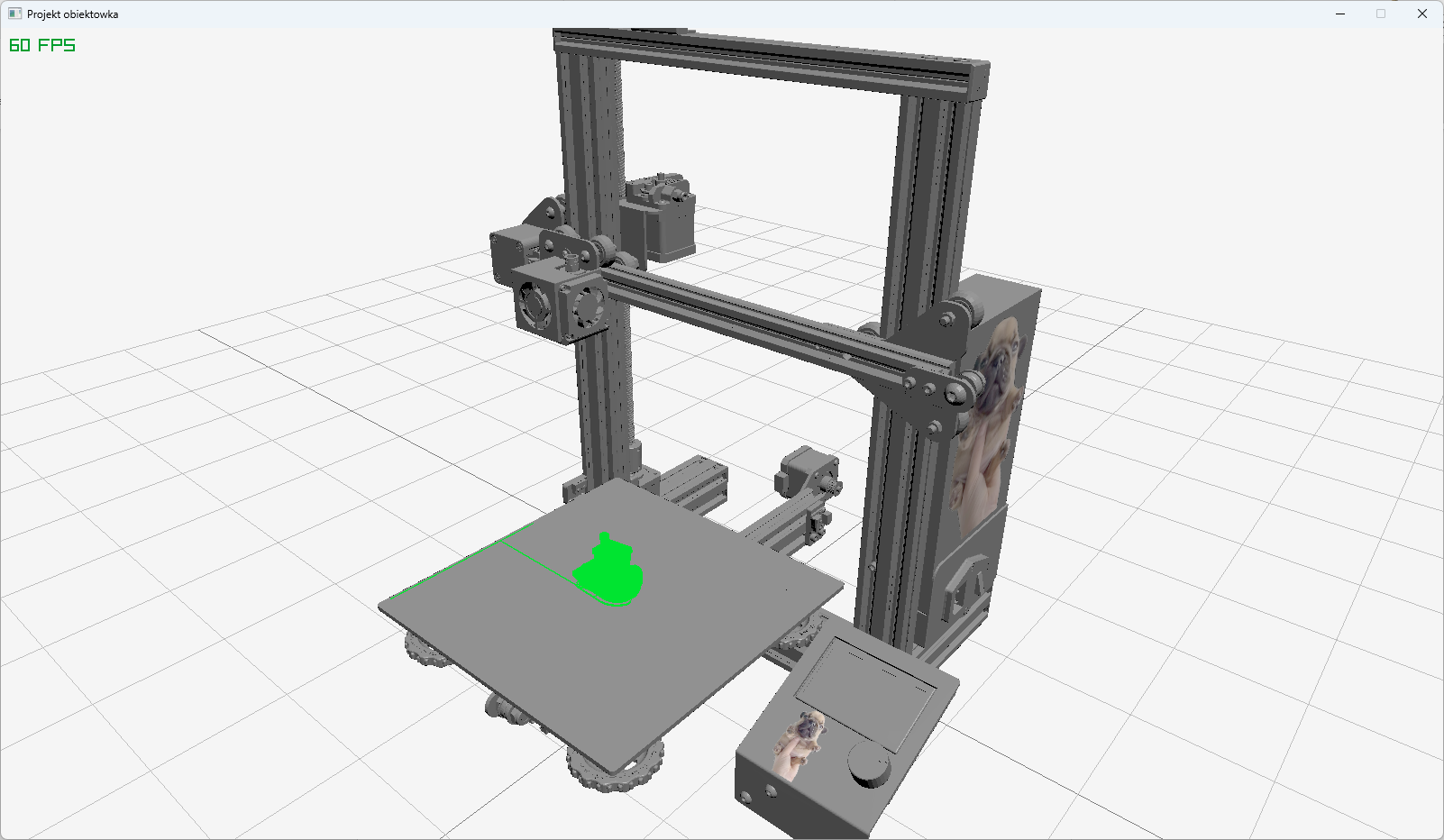
Temat projektu: Wizualizacja drukarki 3D

Projekt miał lekko zmieniony zakres w porównaniu z połowicznym sprawozdaniem po rozmowie z prowadzącym. Teraz program jest skupiony na wizualizacji druku 3D włącznie z generacją wydrukowanej części na podstawie plików gcode.

Obsługa programu jest bardzo prosta i intuitywna. Do kontroli kamery wykorzystuje się standardowe klawisze (WASD, spacja oraz CTRL). Dodatkowo zostały dodane guziki E i Q do kontroli roll oraz scroll do szybszego poruszania się w przód i w tył. Można zmieniać kąt oświetlenia sceny za pomocą strzałek oraz ładować pliki gcode przez przeciągnięcie ich do okna. Do zamknięcia aplikacji służy przycisk ESC, można też wyłączyć aplikację poprzez zamknięcie okna.

Program jest zbudowany z 4 klas (jedna jest polimorficznie połączona z inną), funkcji wspierających oraz z głównej funkcji main.  
W int main() zawarta jest główna pętla symulacji która powoduje kontynuowanie stanu drukarki 3D i drukowanego modelu.  
Klasa main\_model jest wykorzystywana do wgrywania modeli 3D drukarki, główną ramę bezpośrednio i pozostałe osie przez klasę modele za pomocą dziedziczenia. Zawiera konstruktor, dekonstruktor oraz funkcję do rysowania tych modeli.  
Klasa modele rozbudowuje poprzednią klasę o dodatkowe funkcje oraz zmienną prywatną. Te zmiany pozwalają tej klasie na wsparcie BoundingBox’ów które są użyteczne w detekcji kolizji.  
Klasa TargetPoint jest odpowiedzialna za wszystkie ruchy elementów co egzekwuje jej główna funkcja MoveToPoint. Wykonuje te ruchy w oparciu o wektor zawierający wszystkie parametry co do ruchów, które miałaby drukarka wykonać, w tym prędkość ruchów.  
Ostatnią klasą jest Extruder, która odpowiada za generowanie wizualizacji wydruku. Cały proces opiera się o generację meshy na podstawie ruchów głowicy oraz dodatkowej publicznej listy zawierającej informacje o rodzaju ruchu wykonywanym. Potrzebna jest generacja wielu meshy zamiast jednego, gdyż standardowa struktura mesha w raylibie przepełniała się nawet przy najmniejszych wydrukach. Generacja wielu meshy pozwala też na dobre zarządzanie pamięcią programu i oferuje przy tym też dobrą optymalizacje programu.  
Ostatnim ważnym elementem programu jest funkcja GcodeAnalizer, która odpowiada za interpretację plików pod drukarki 3D (gcode) i wygenerowanie na ich podstawie parametrów ruchu dla funkcji MoveToPoint.  
W trakcie druku można załadować nowy plik gcode bez konieczności ponownego odpalania aplikacji. Powoduje to wyczyszczenie danych z wektora posiadającego parametry ruchu i wyczyszczenie wszystkich wygenerowanych wcześniej meshy.