Symulacja dyskretna systemów złożonych
Symulacja ogrzewania pomieszczenia

Emilia Mączka, Weronika Wisz

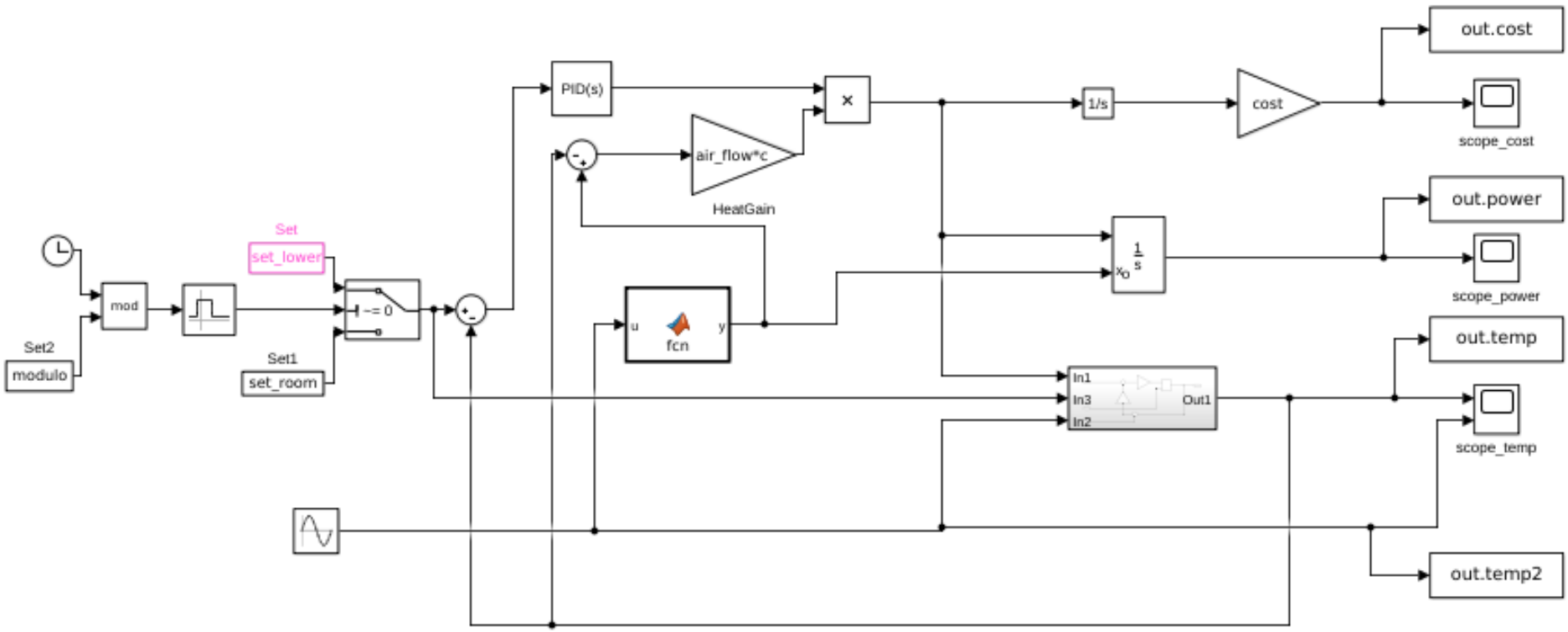
Akademia Górniczo-Hutnicza
Wydział EAIiIB
Informatyka

14.05.2020

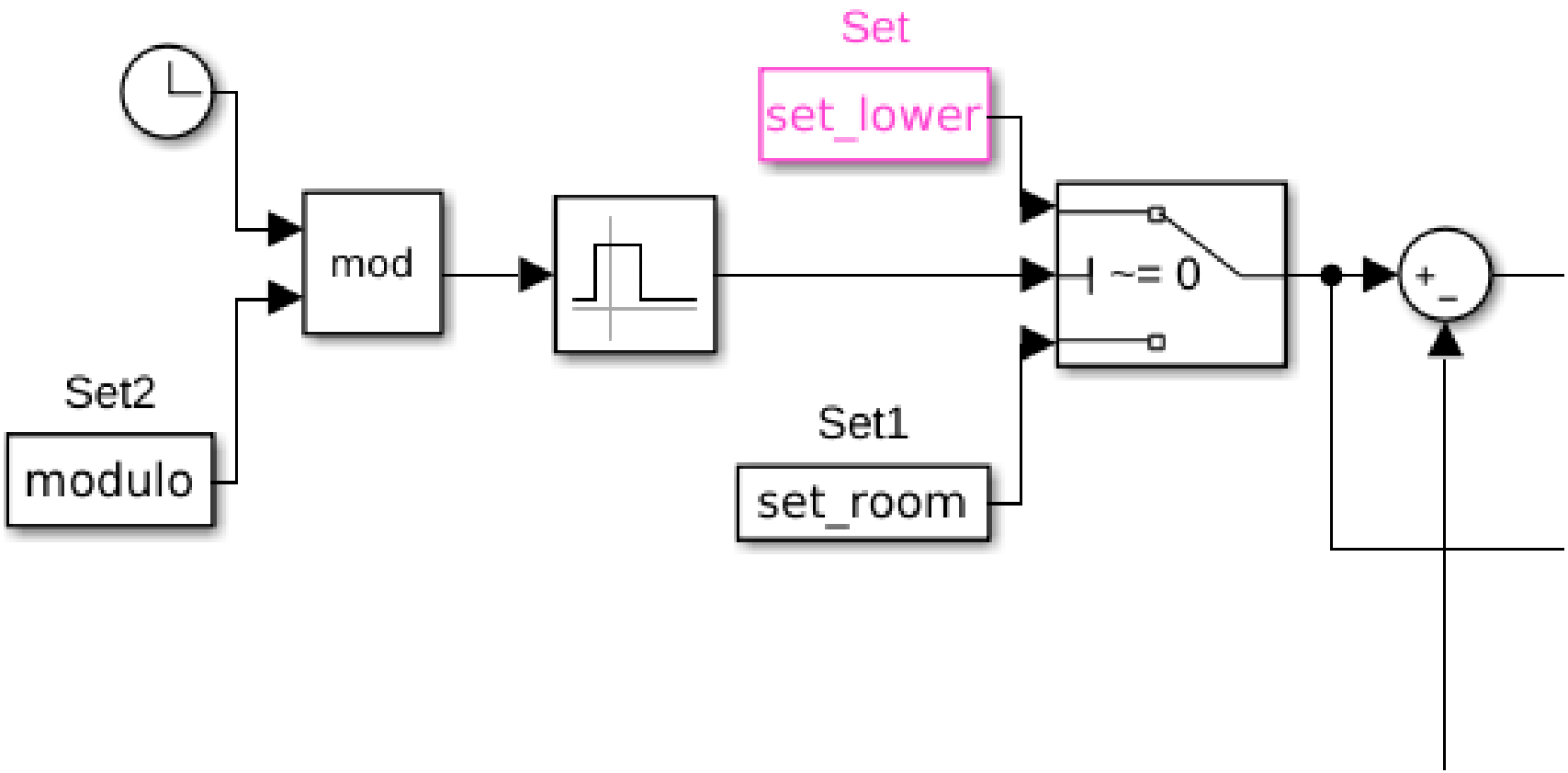
Bieżące postępy

- Ulepszenie modelu, dodanie wykresu ilość zużytego ciepła, oraz możliwości obniżania temperatury pomieszczenia w określonych godzinach.
- Stworzenie dokumentacji ze wszystkimi aspektami fizycznymi potrzebnymi do stworzenia modelu oraz wykorzystanymi technikami, które stosuje się w intalacjach ogrzewania.
- Stworzenie aplikacji obsługującej model, która umożliwia użytkownikowi przprowadzenie interesujących go symulacji, szczegóły opisane w dalszej części prezentacji.

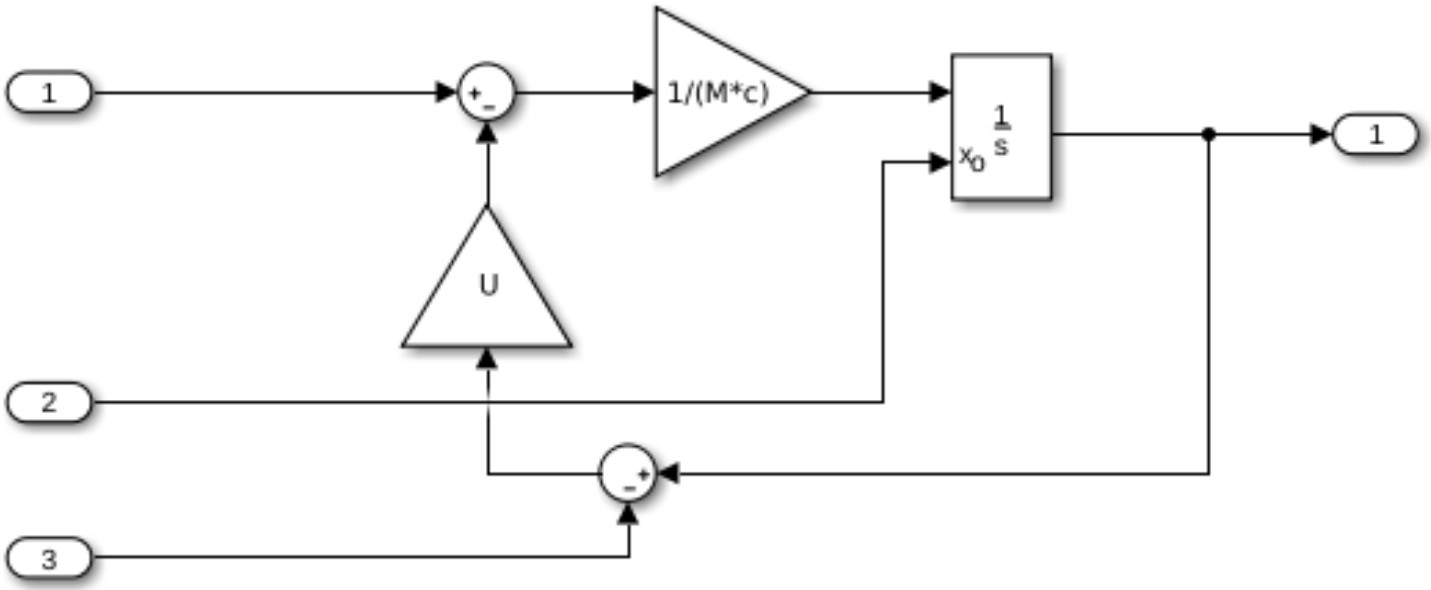
Model ogrzewania



Możliwość obniżania temperatury



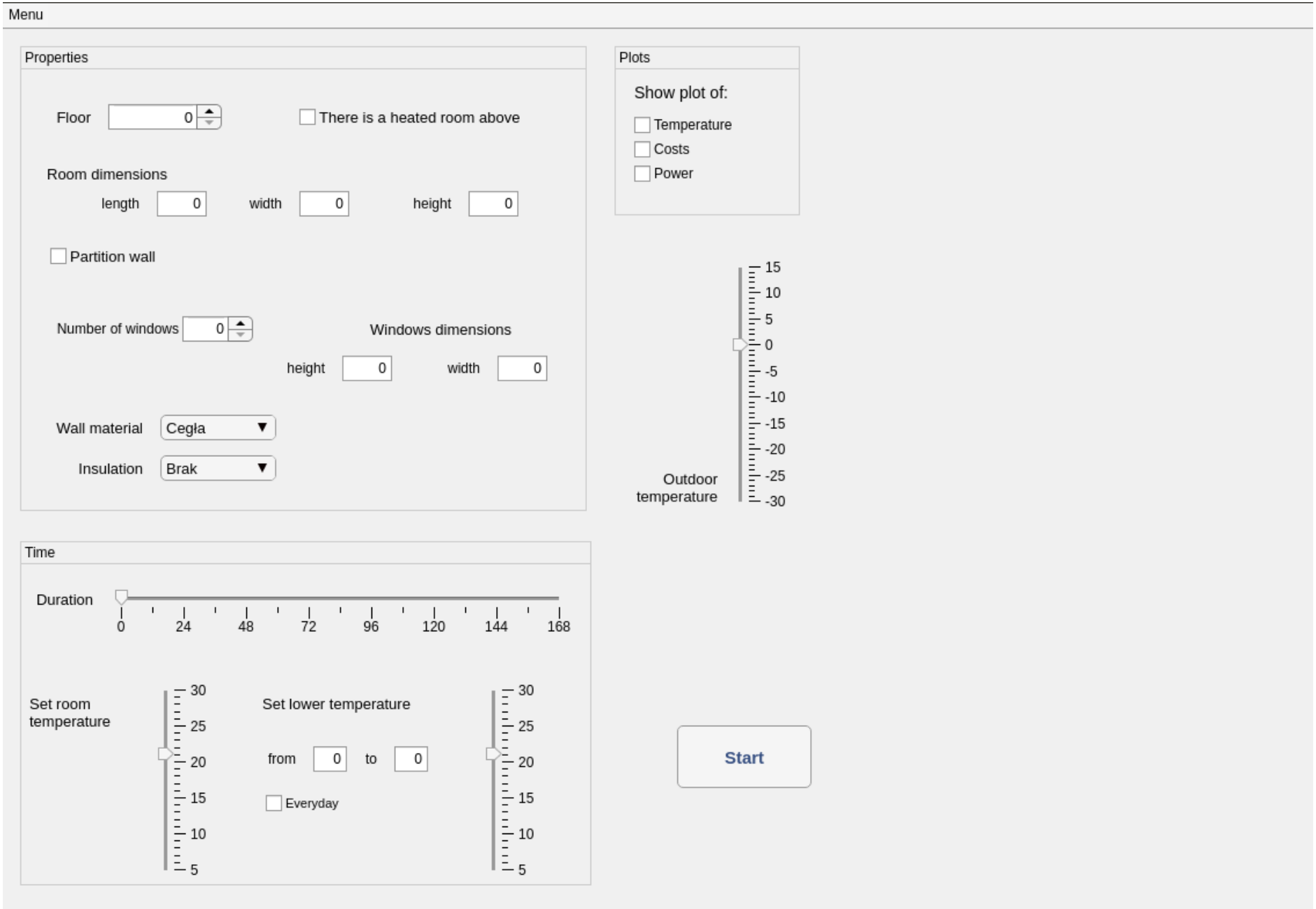
Uwzględnianie strat cieplnych

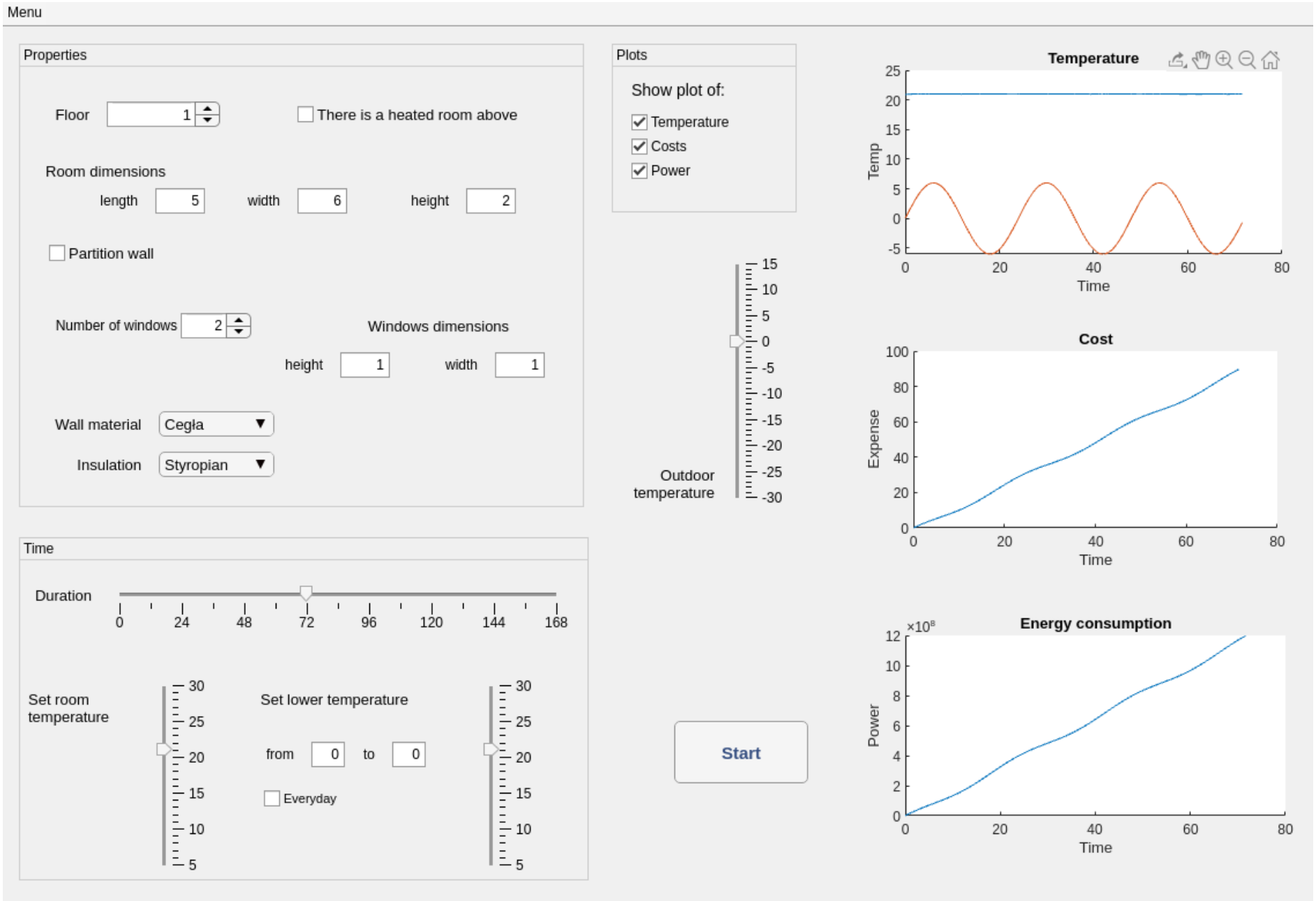


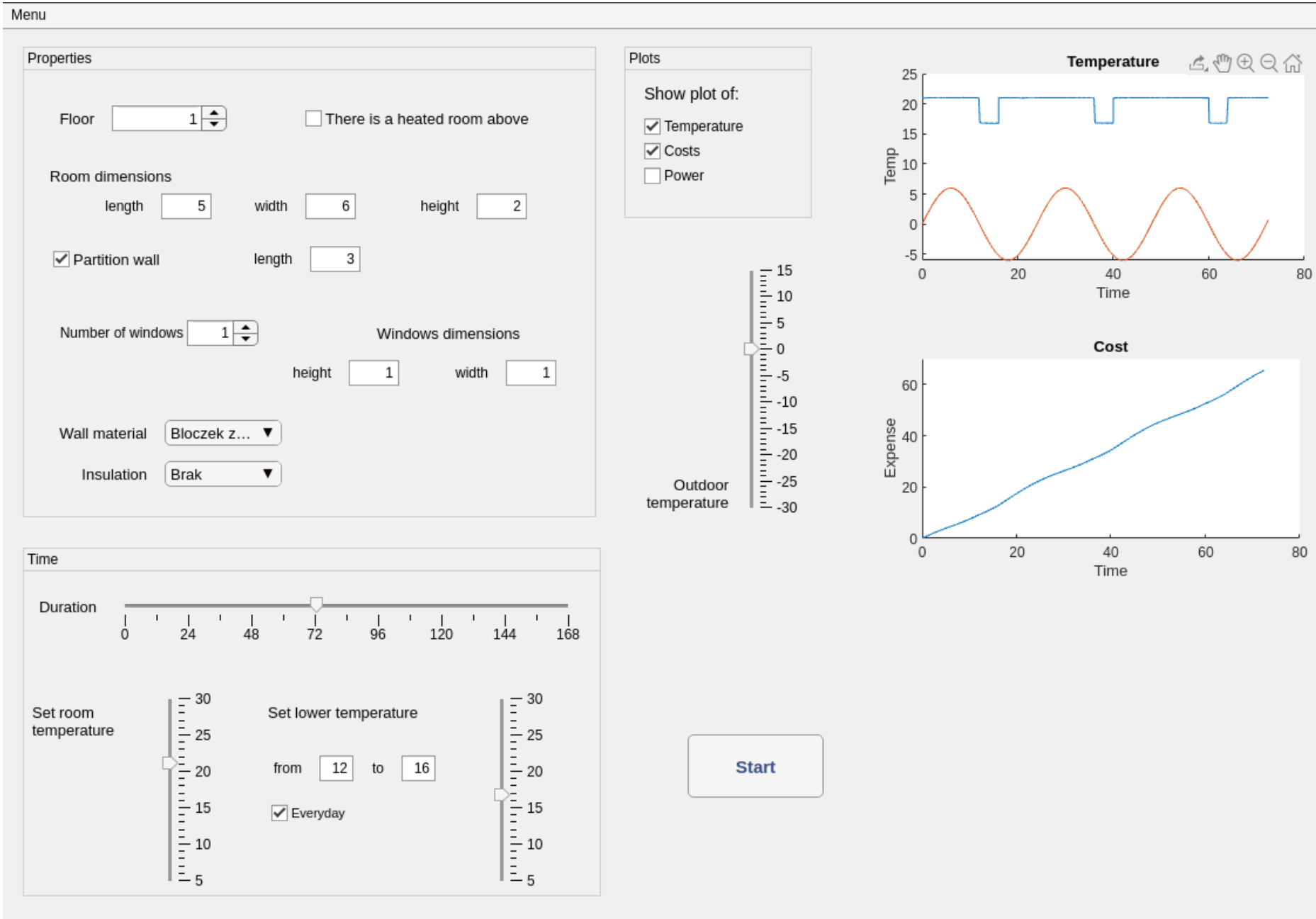
Aplikacja

Aplikacja została stworzona za pomocą MATLAB App Designer. Można w niej przeprowadzić symulacje z wykorzystaniem stworzonego przez nas modelu. Użytkownik ma możliwość wprowadzenia danych o swoim pomieszczeniu oraz innych czynnikach takich jak:

- wymiary pokoju,
- wymiary i ilość okien,
- materiał z jakiego zbudowane są ściany, oraz ocieplenie,
- jaka temperatura w pokoju ma być utrzymywana, a także czy w określonych godzinach chcemy ją obniżać,
- średnią temperaturę na zewnątrz,
- które z wykresów chcemy wyświetlać, temperatury, kosztów, ilości zużytego ciepła.


Menu
Properties
Floor
0
There is a heated room above
Room dimensions
length
0
width
0
height
0
Partition wall
Number of windows
0
Windows dimensions
height
0
width
0
Wall material
Cegła
Insulation
Brak
Plots
Show plot of:
Temperature
Costs
Power
15
10
5
0
-5
-10
-15
-20
-25
-30
Outdoor temperature
Time
Duration
0
24
48
72
96
120
144
168
Set room temperature
30
25
20
15
10
5
Set lower temperature
from
0
to
0
Everyday
30
25
20
15
10
5
Start


Menu
Properties
Floor
1
There is a heated room above
Room dimensions
length
5
width
6
height
2
Partition wall
Number of windows
2
Windows dimensions
height
1
width
1
Wall material
Cegła
Insulation
Styropian
Time
Duration
0
24
48
72
96
120
144
168
Set room temperature
30
25
20
15
10
5
Set lower temperature
from
0
to
0
Everyday
Plots
Show plot of:
Temperature
Costs
Power
Outdoor temperature
15
10
5
0
-5
-10
-15
-20
-25
-30
Start
Temperature
Temp
Time
Cost
Expense
Time
Energy consumption
×10^8
Power
Time


Menu
Properties
Floor
1
There is a heated room above
Room dimensions
length
5
width
6
height
2
Partition wall
length
3
Number of windows
1
Windows dimensions
height
1
width
1
Wall material
Bloczek z...
Insulation
Brak
Plots
Show plot of:
Temperature
Costs
Power
Outdoor temperature
15
10
5
0
-5
-10
-15
-20
-25
-30
Temperature
Temp
Time
Cost
Expense
Time
Time
Duration
0
24
48
72
96
120
144
168
Set room temperature
30
25
20
15
10
5
Set lower temperature
from
12
to
16
Everyday
Start

Skrypt

```
% Callbacks that handle component events
methods (Access = private)

    % Code that executes after component creation
    function startupFcn(app)
        app.TempAxes.Visible = 'off';
        app.CostAxes.Visible = 'off';
        app.PowerAxes.Visible = 'off';

    end

    % Button pushed function: StartButton
    function StartButtonPushed(app, event)

        % Set temperature

        lower_from = app.LowerFrom.Value;
        lower_to = app.LowerTo.Value;

        if(app.EverydayCheckBox.Value == 1)
            assignin('base','modulo',24);
        else
            assignin('base','modulo',app.DurationSlider.Value);
        end

        if(lower_from < lower_to)
            assignin('base', 'lower_from', lower_from);
            assignin('base', 'lower_to', lower_to);
            assignin('base',"set_room", app.RoomTempSlider.Value);
            assignin('base',"set_lower", app.LowerTempSlider.Value);
        elseif(lower_from > lower_to)
            assignin('base', 'lower_to', lower_from);
            assignin('base', 'lower_from', lower_to);
            assignin('base',"set_room", app.LowerTempSlider.Value);
            assignin('base',"set_lower", app.RoomTempSlider.Value);

        end
```

Skrypt

```
% Room dimensions

%length [m]
length = app.RoomLength.Value;
%width [m]
width = app.RoomWidth.Value;
%height [m]
height = app.RoomHeight.Value;
% radians to degrees
%r2d = 180/pi;
%roof pitch
%roof_pitch = 40/r2d;

% window area
win_num = app.WindowsNum.Value;
% Height of windows = 1 m
win_height = app.WindowsHeight.Value;
% Width of windows = 1 m
win_width = app.WindowsWidth.Value;
win_area = win_num * win_height * win_width;


% wall area
wall_area = 2 * length * height + 2 * width * height + 2 * length * width  - win_area;

% window resistance
win_lambda = 0.78;
win_d = 0.01;
win_res = win_d /(win_lambda * 3600 * win_area);
```

Skrypt

```
% wall resistance
% (włokno szklane do ocieplenia budynku)
% hour is the time unit
% [k] = J/s/m/C
lambda = app.WallMaterial.Value;
if(strcmp(lambda,"Cegła"))
    wall_lambda = 0.77;
elseif(strcmp(lambda,"Pustak"))
    wall_lambda = 0.4;
elseif(strcmp(lambda,"Żelbet"))
    wall_lambda = 1.7;
else
    wall_lambda = 0.2;
end

wall_d = 0.2;
wall_res = wall_d/wall_lambda;

lambda_i = app.Insulation.Value;
if(strcmp(lambda_i,"Brak"))
    ins_lambda = 0;
elseif(strcmp(lambda_i,"Styropian"))
    ins_lambda = 0.04;
elseif(strcmp(lambda_i,"Wełna mineralna"))
    ins_lambda = 0.18;
else
    ins_lambda = 0.17;
end
```

Skrypt

```
% Heat flux density q [W/m^2]
% q = U(Ti-Te)   U - material conductivity [W/(K*m^2)

%convective heat transfer
if(ins_lambda ~= 0)
    ins_d = 0.1;
    ins_res = ins_d/ins_lambda;
    total_wall_res = (wall_res + ins_res)/(3600 * wall_area);
else
     total_wall_res = wall_res /(3600 * wall_area);
end


U = 1/win_res + 1/total_wall_res;
assignin('base','U',U);


%density of air [kg/m^3]
dens_air = 1.2250;
%air mass
M = length * width * height * dens_air;
assignin('base','M',M);
%cp of air (273 K) [J/kgK]
c = 1005.4;
assignin('base','c',c);
%air flow rate [kg/hr]
air_flow = 3600;
assignin('base','air_flow',air_flow);

% 1 kW-hr = 3.6e6 J
% cost = 0.27 zl / 3.6e6 J
cost = 0.27/3.6e6;
assignin('base',"cost",cost);
```

Skrypt

```
% Draw plots
assignin('base','init_temp', app.OutdoorTempSlider.Value);
simout = sim('model','StopTime',num2str(app.DurationSlider.Value));

% Plot of temperature outisde and inside the room
if(app.TempCheckBox.Value == 1)
    app.TempAxes.Visible = 'on';
    plot(app.TempAxes, simout.temp.Time ,simout.temp.Data);
    hold(app.TempAxes,'on');
    plot(app.TempAxes, simout.temp2.Time, simout.temp2.Data);
    hold(app.TempAxes,'off');
    app.TempAxes.YLim = [-inf,25];
else
    app.TempAxes.Visible = 'off';
    app.TempAxes.cla;
end

% Consumption cost plot
if(app.CostsCheckBox.Value == 1)
    app.CostAxes.Visible = 'on';
    plot(app.CostAxes,simout.cost.Time, simout.cost.Data);
else
    app.CostAxes.Visible = 'off';
    app.CostAxes.cla;
end

% Plot of produced energy
if(app.PowerCheckBox.Value == 1)
    app.PowerAxes.Visible = 'on';
    plot(app.PowerAxes, simout.power.Time,simout.power.Data);
else
    app.PowerAxes.Visible = 'off';
    app.PowerAxes.cla;
end
```