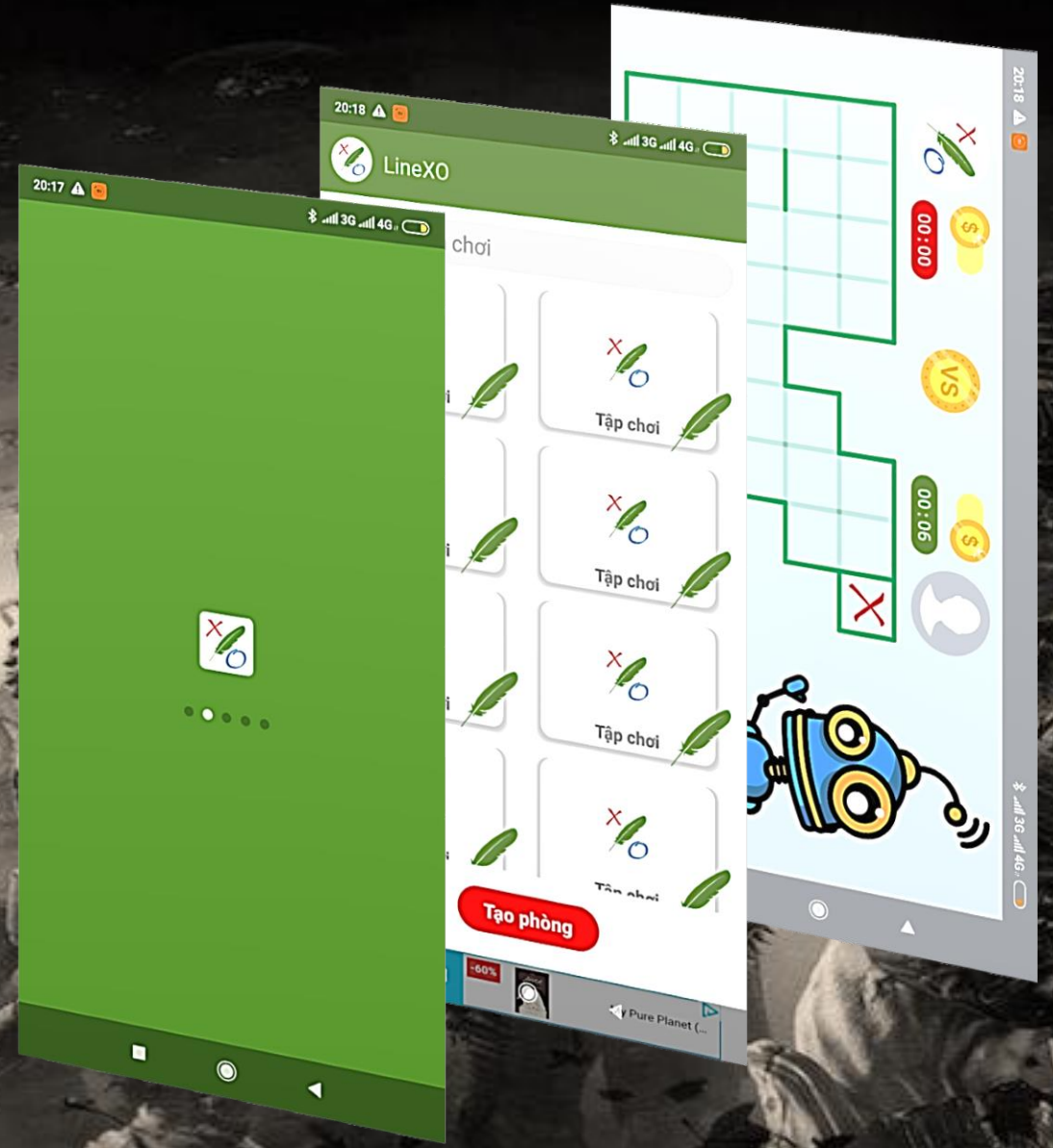
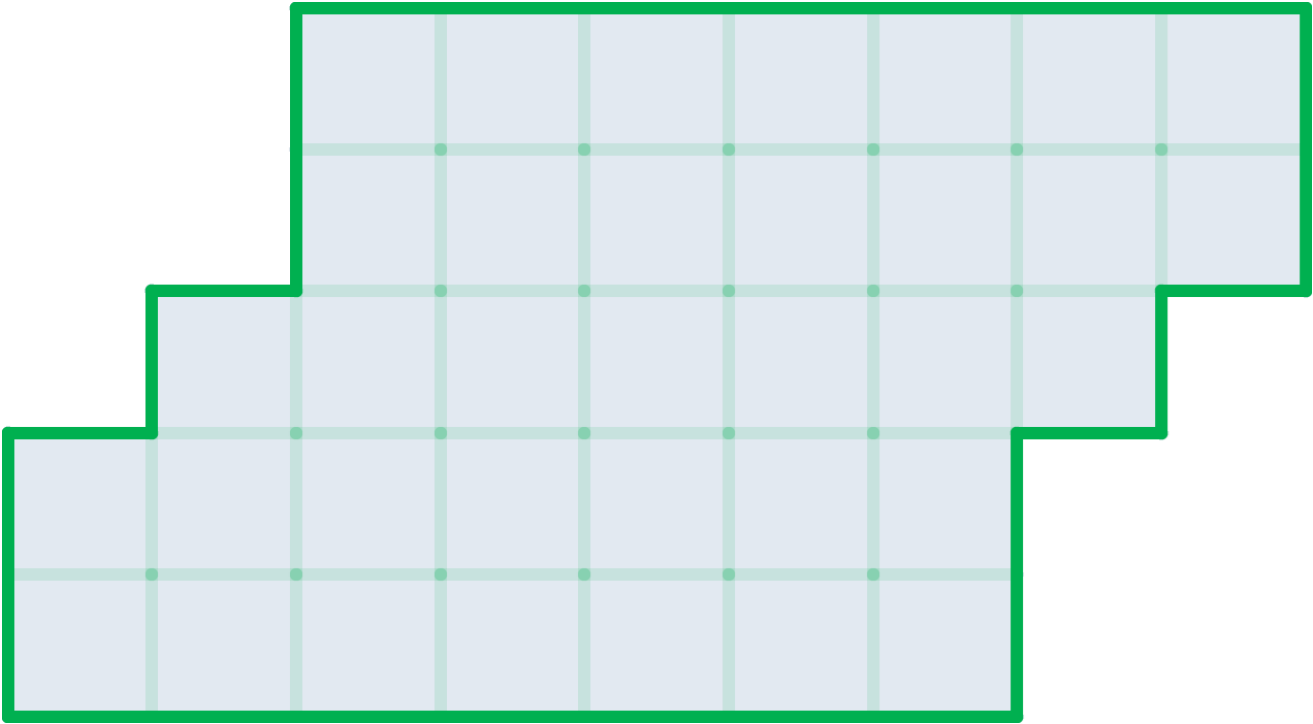


LineX

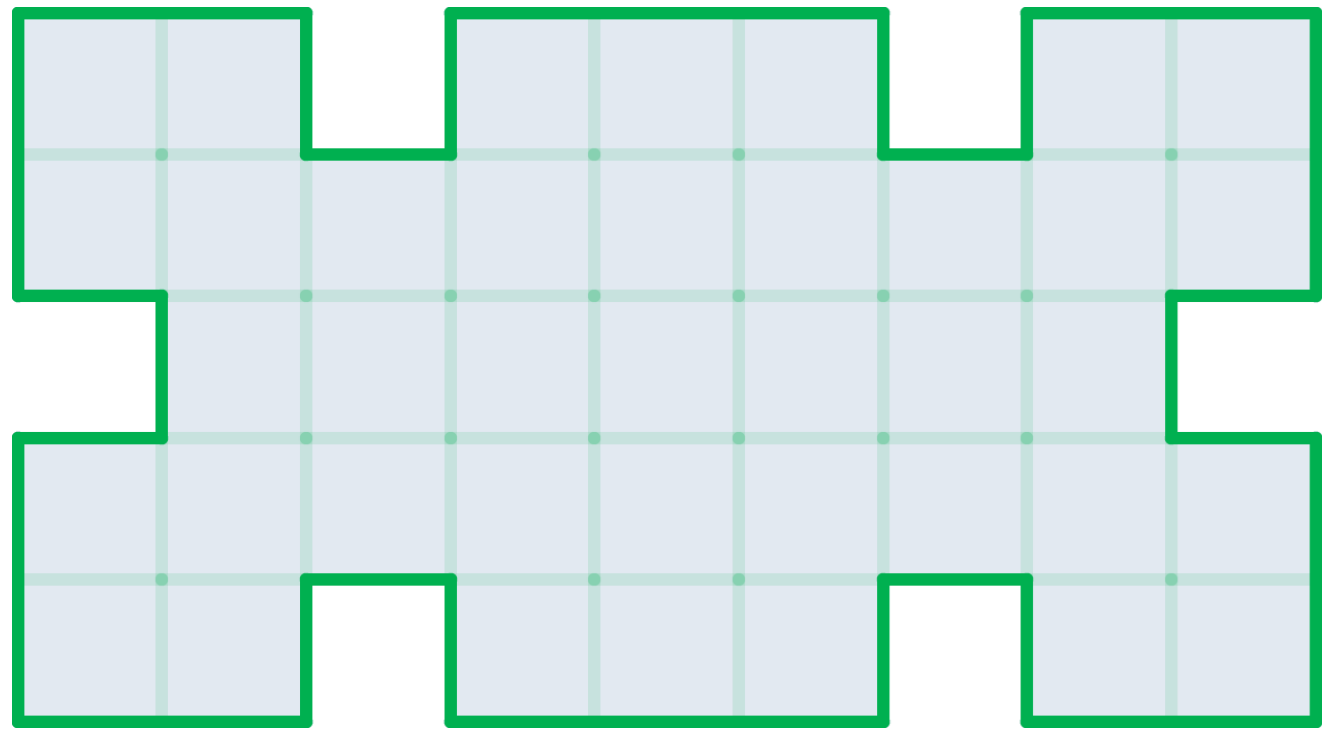


Anh Hùng Sang Trọng

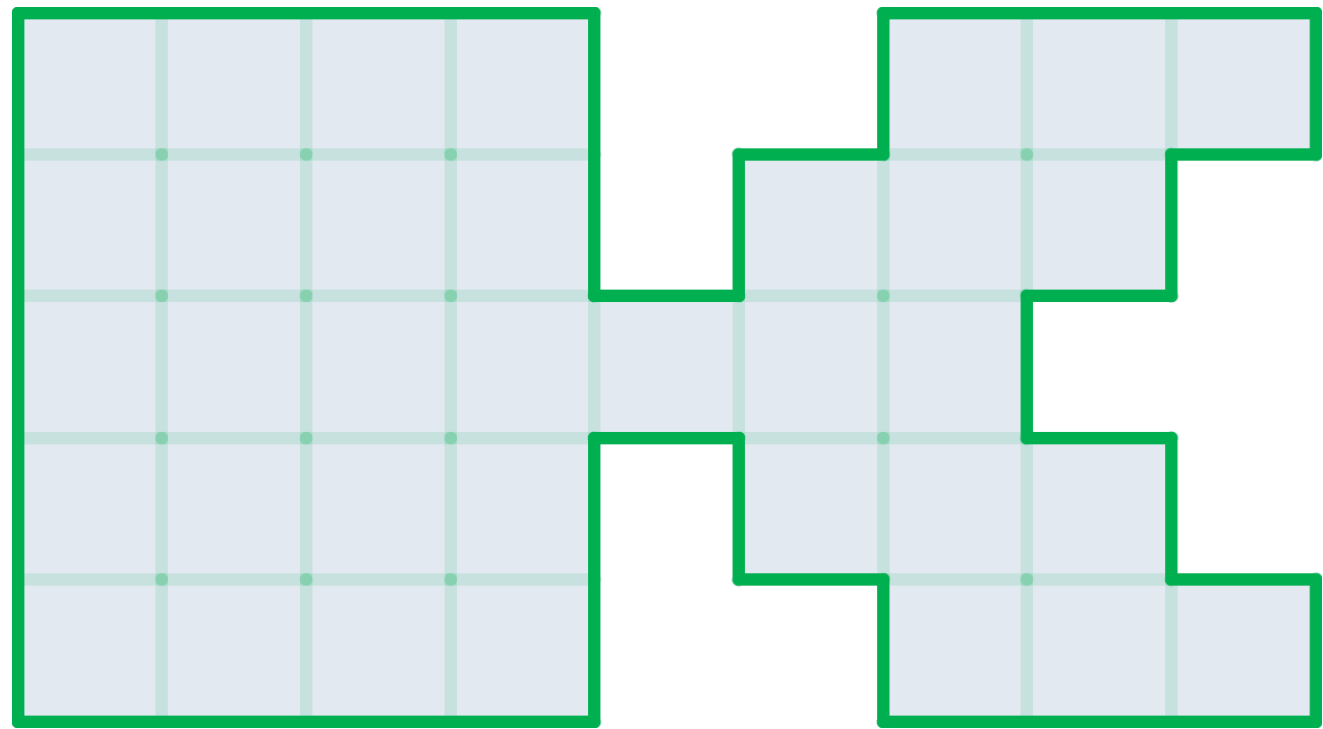
Luật chơi



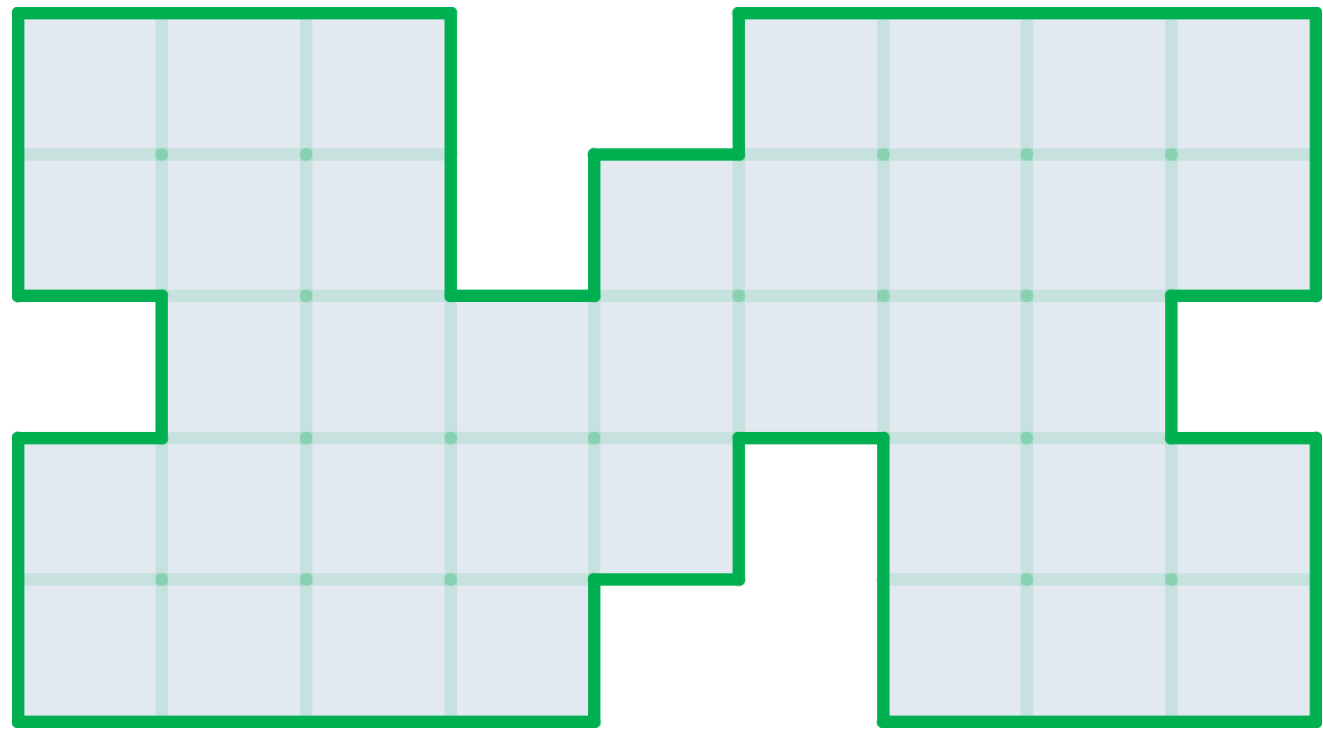
Luật chơi



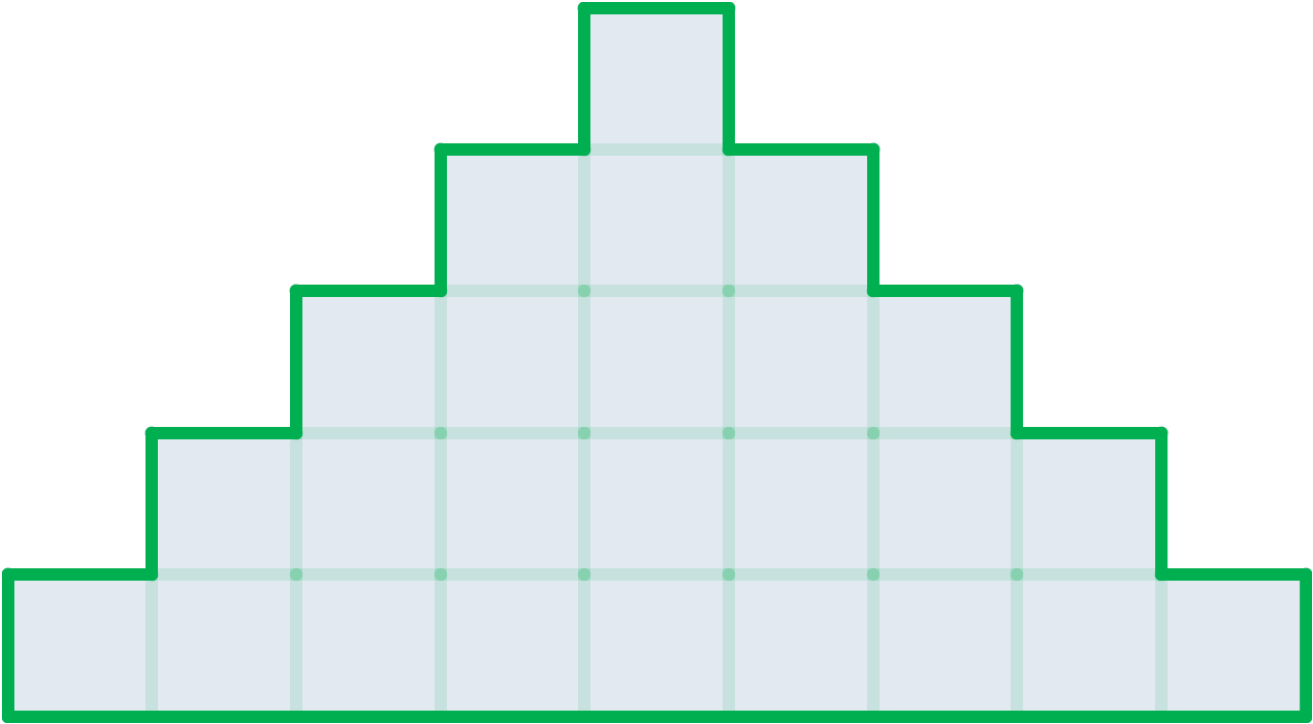
Luật chơi



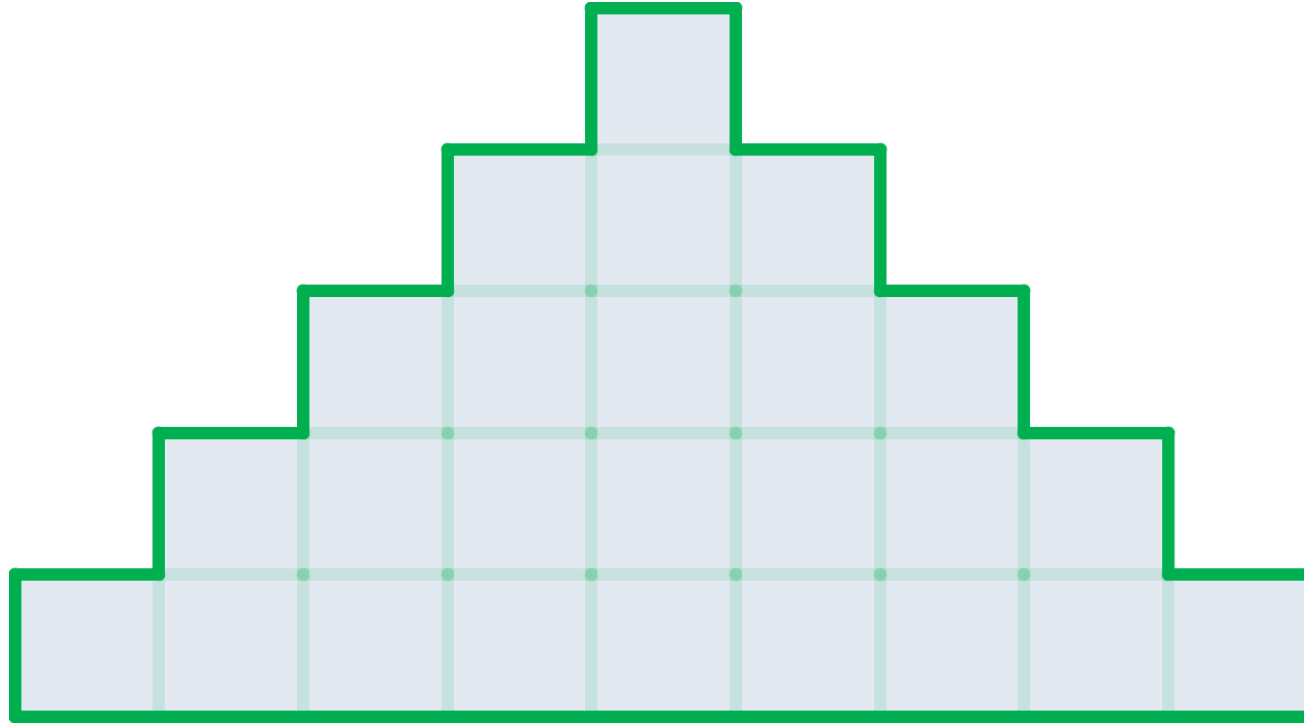
Luật chơi



Luật chơi

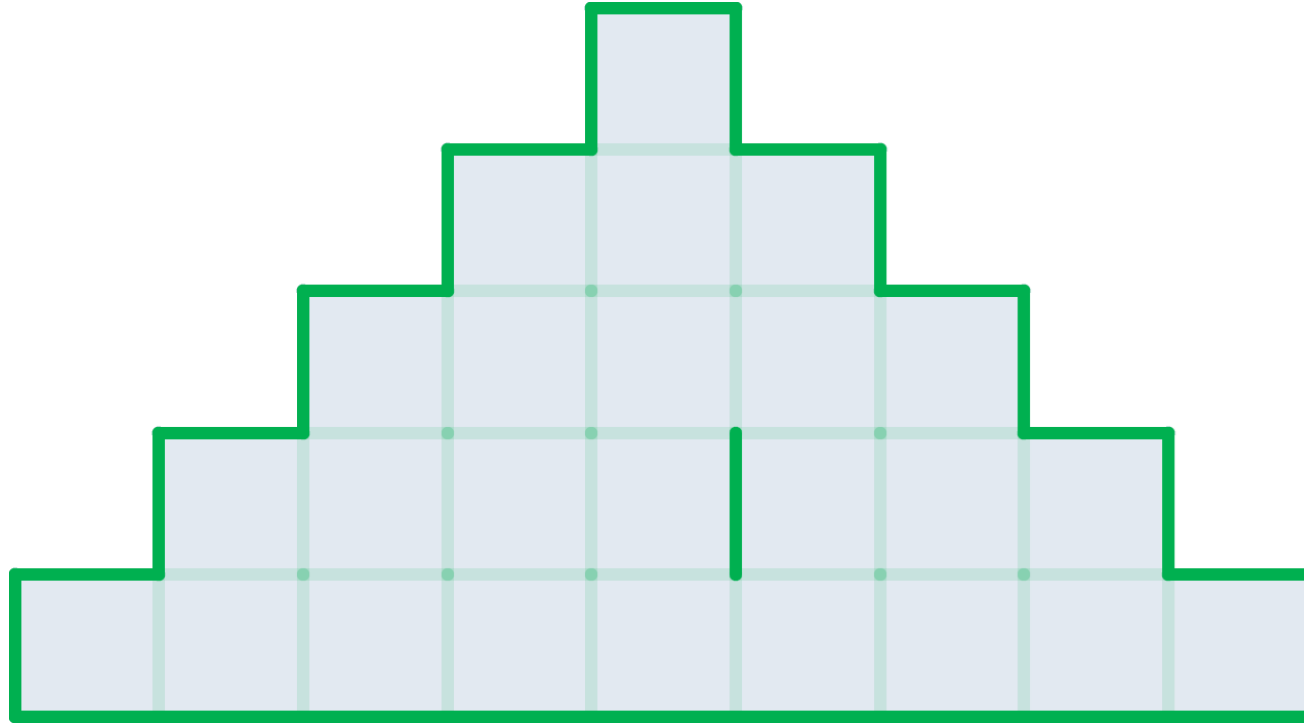


Luật chơi



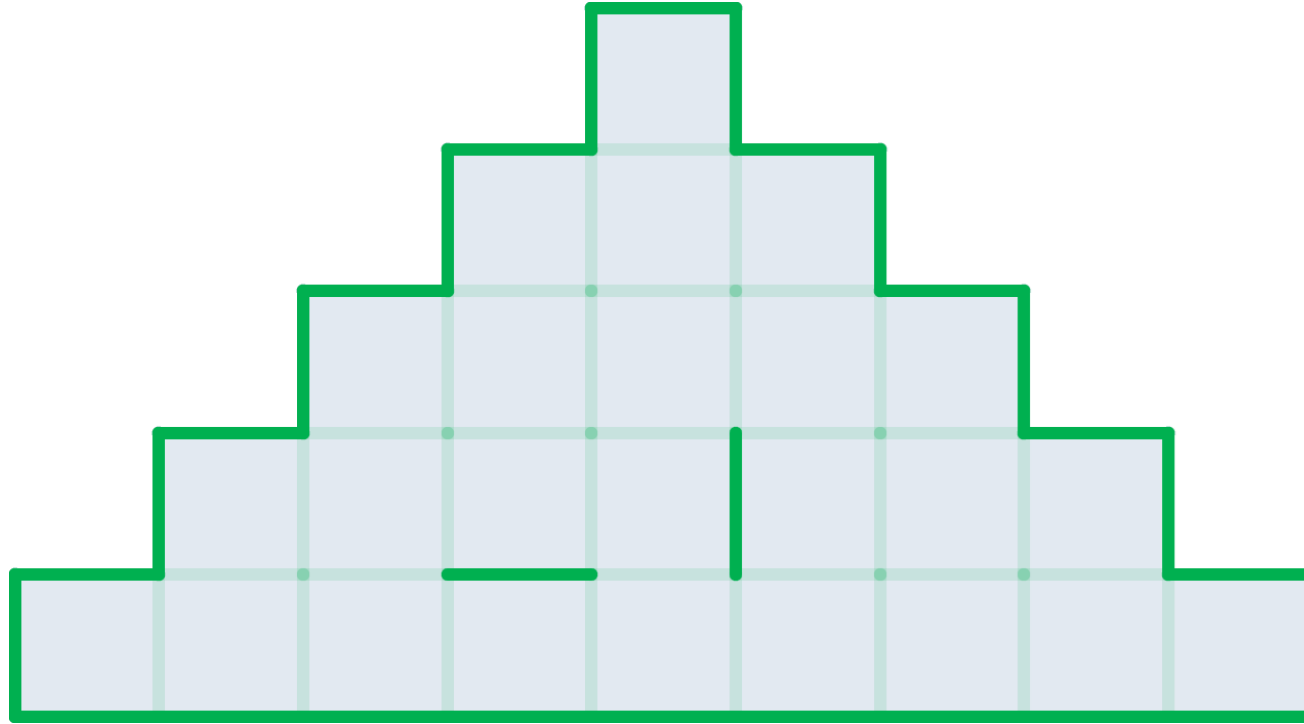
Lần lượt từng người chơi xây tường chiếm đất

Luật chơi



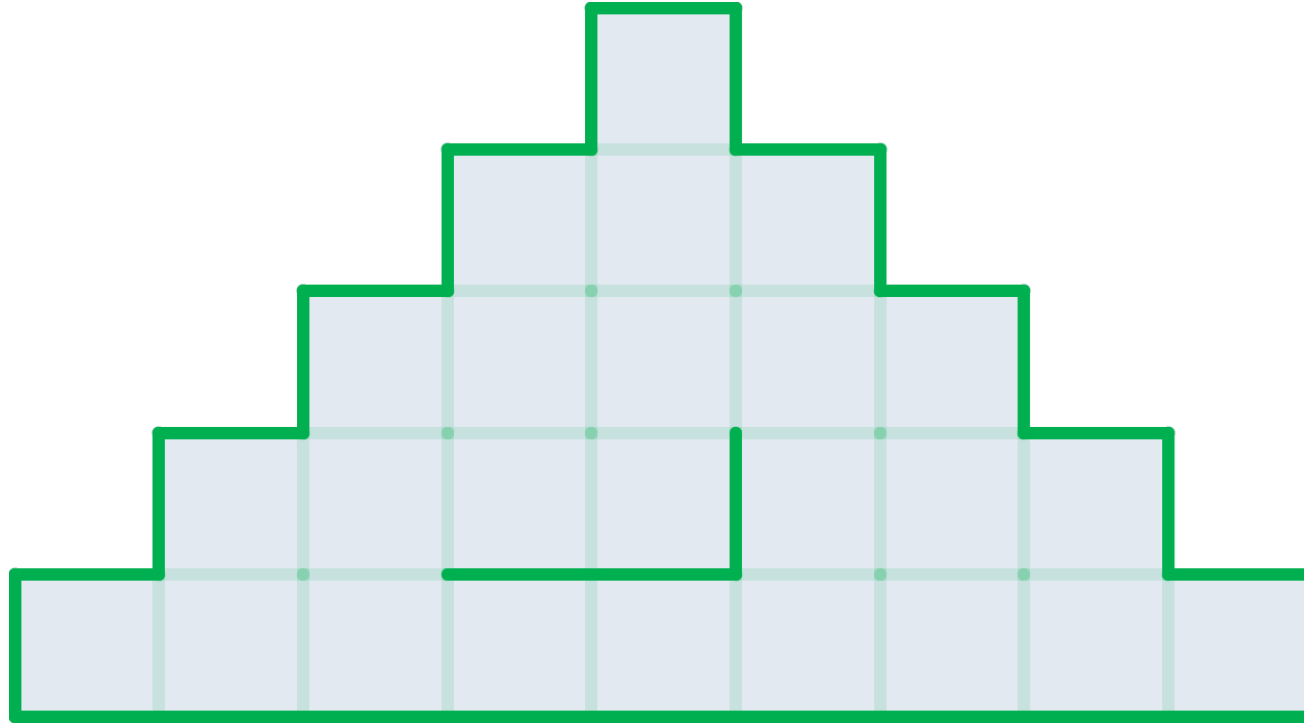
Lần lượt từng người chơi xây tường chiếm đất

Luật chơi



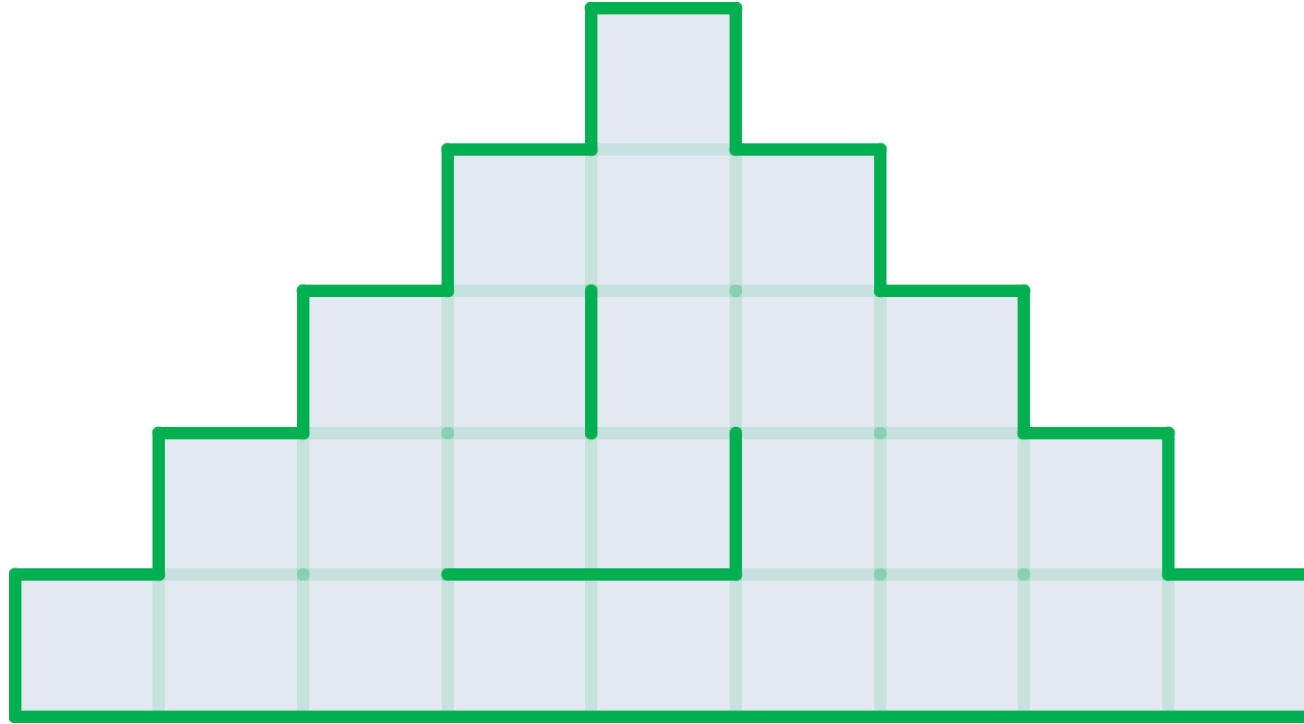
Lần lượt từng người chơi xây tường chiếm đất

Luật chơi



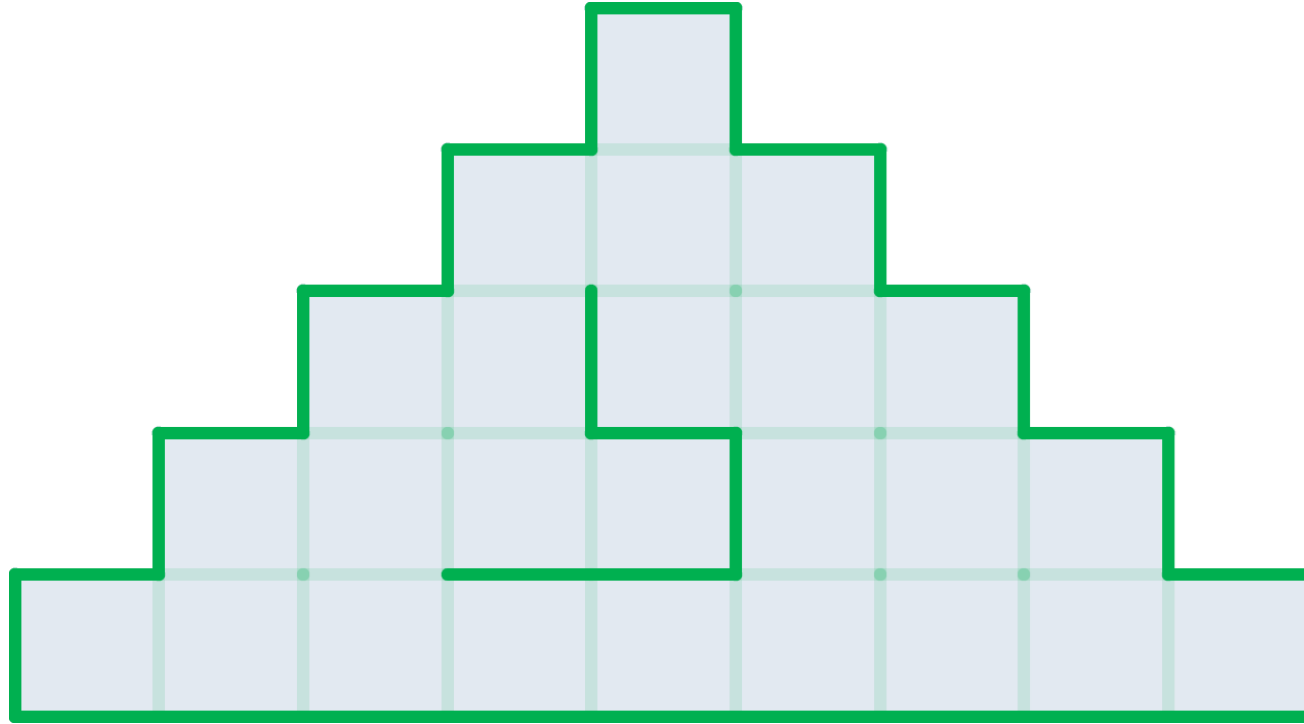
Lần lượt từng người chơi xây tường chiếm đất

Luật chơi



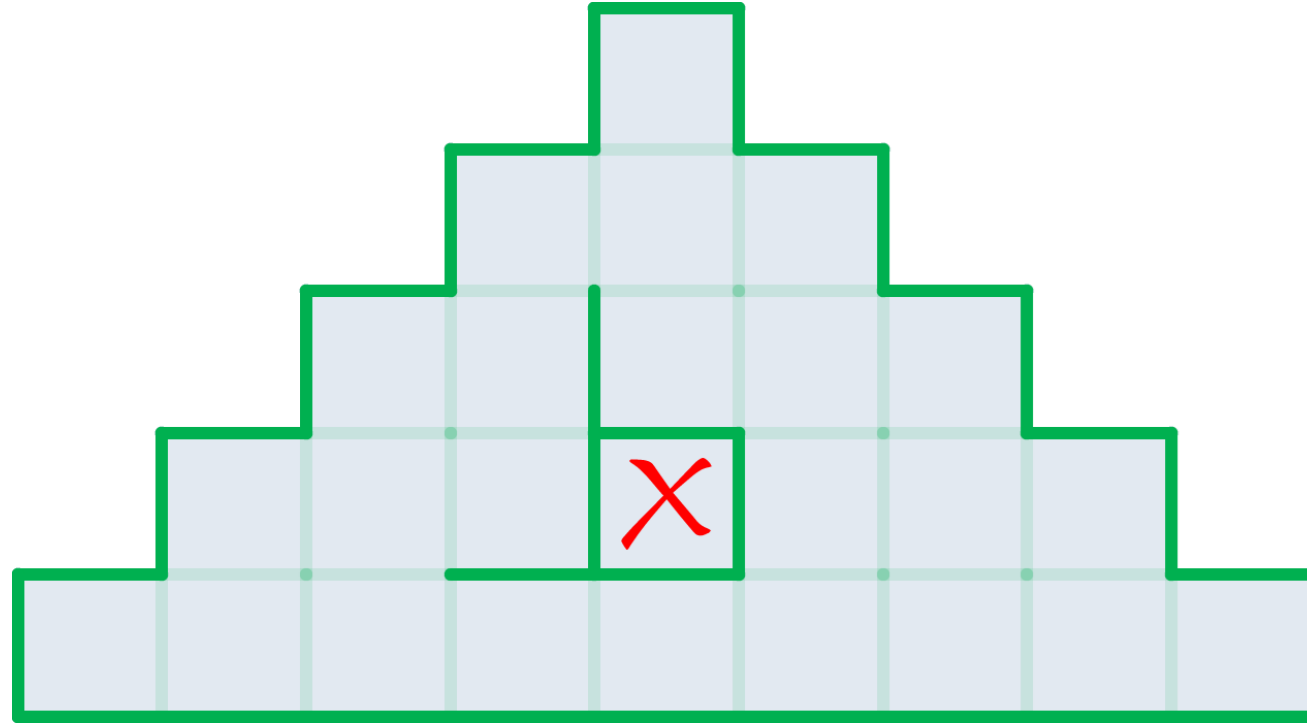
Lần lượt từng người chơi xây tường chiếm đất

Luật chơi



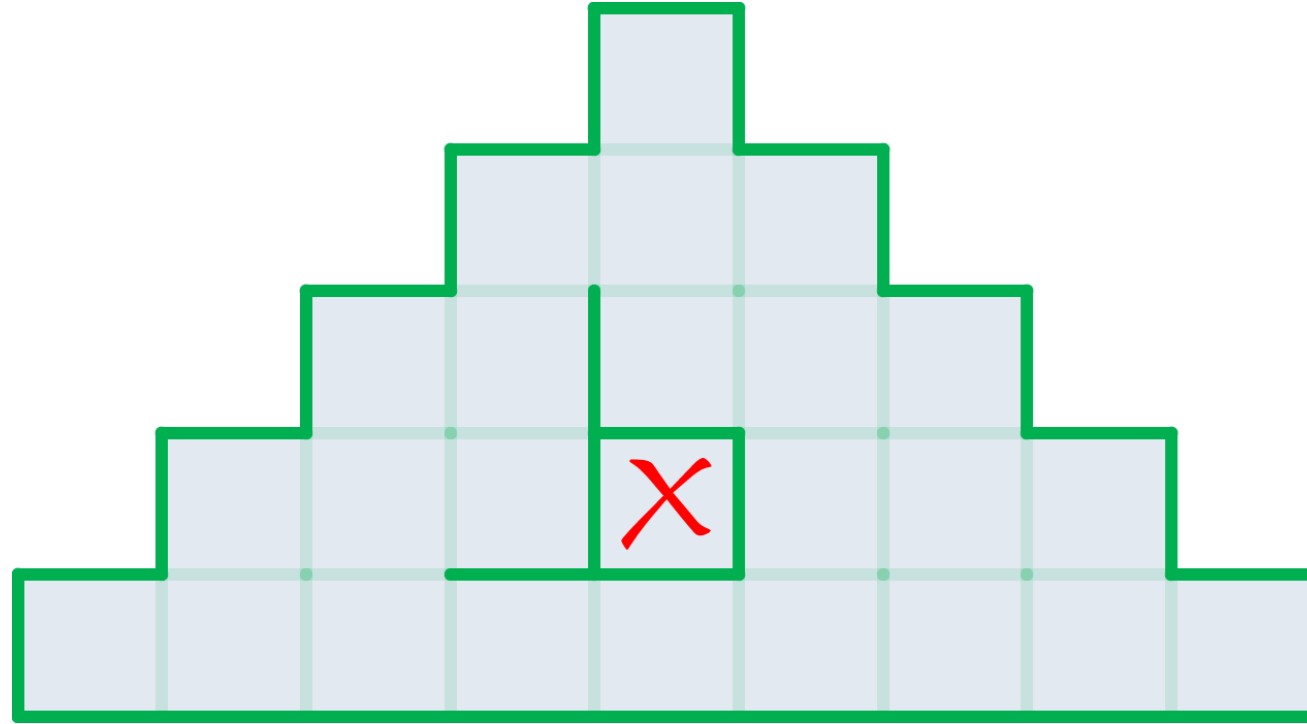
Lần lượt từng người chơi xây tường chiếm đất

Luật chơi



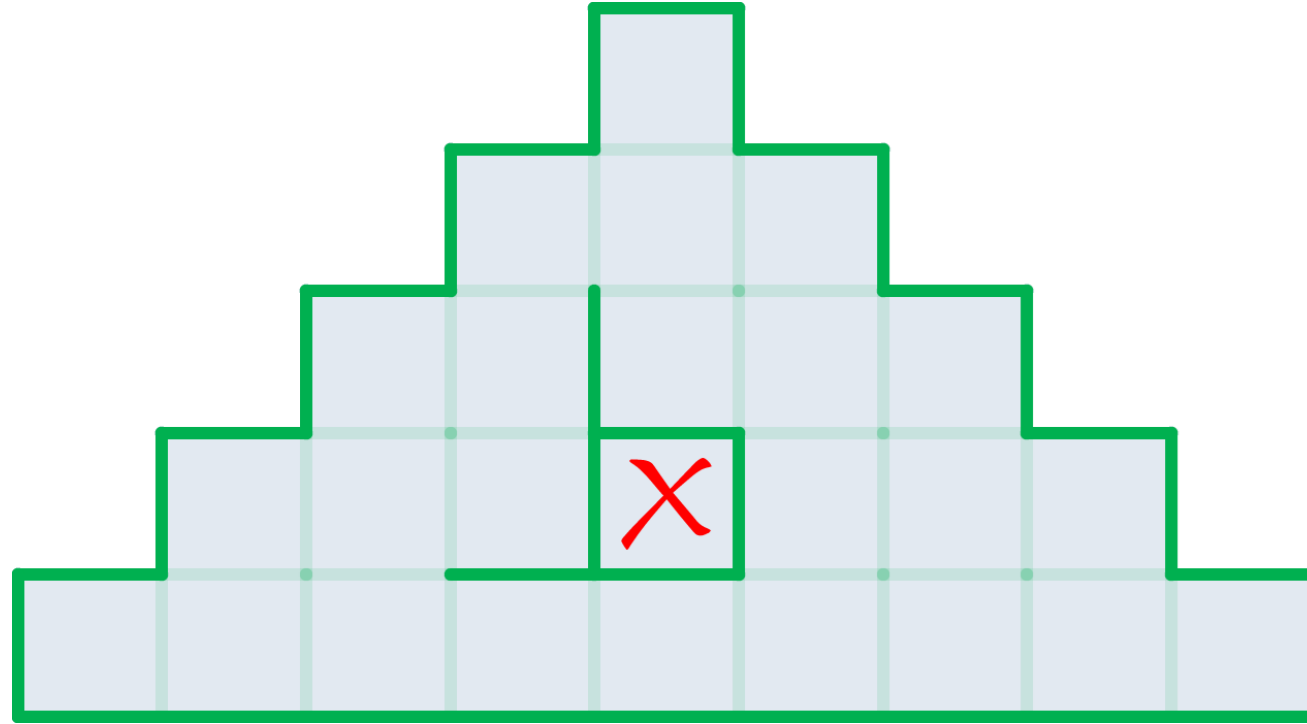
Lần lượt từng người chơi xây tường chiếm đất

Luật chơi



Người chiếm đất được tiếp tục xây tường

Luật chơi



Một bức tường không được xây quá 15 giây

Linux

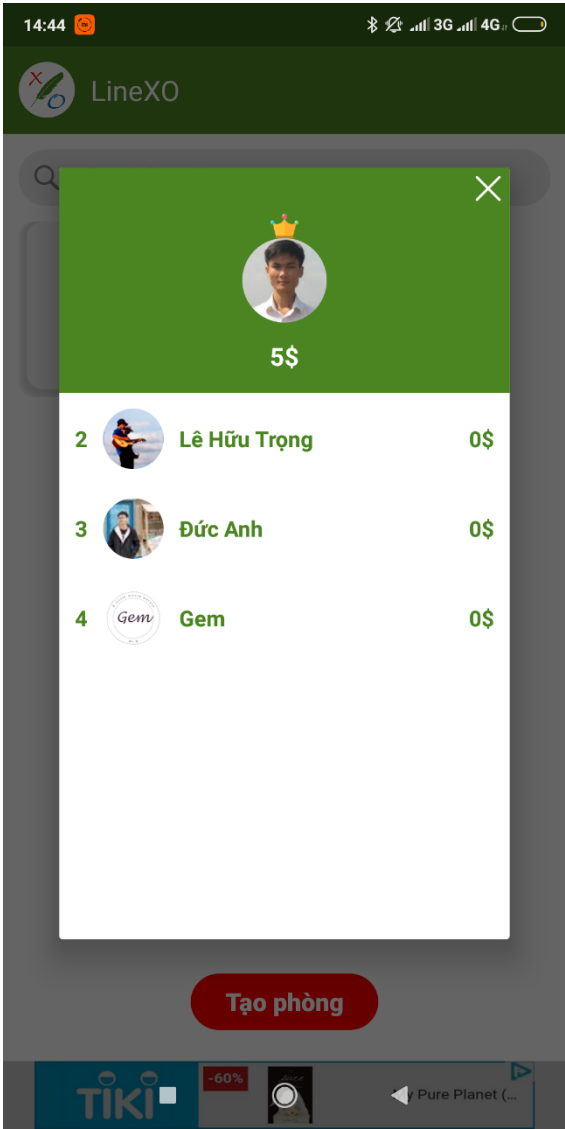
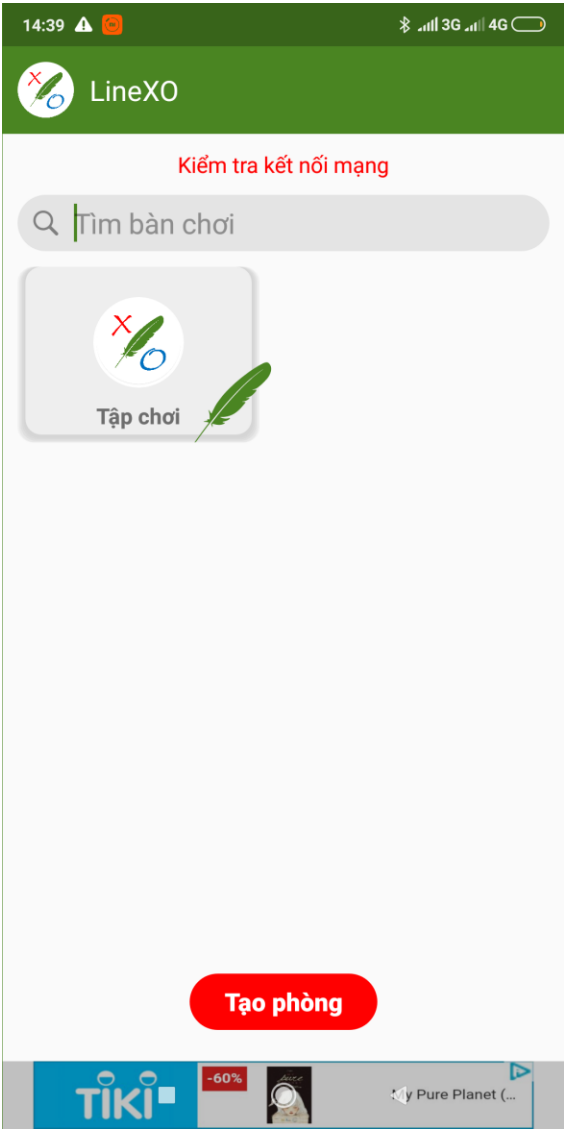
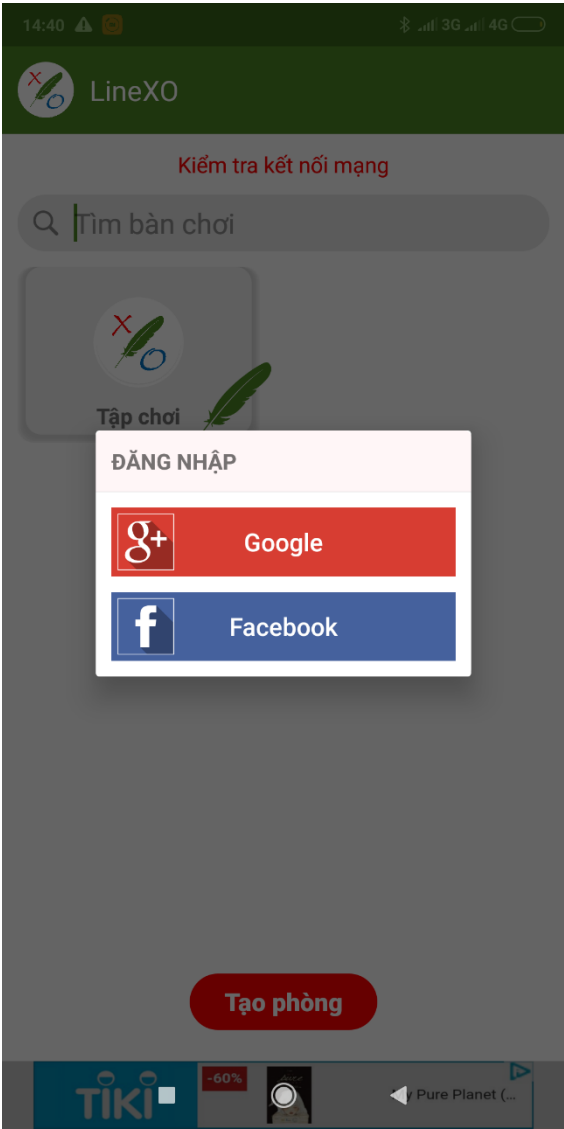
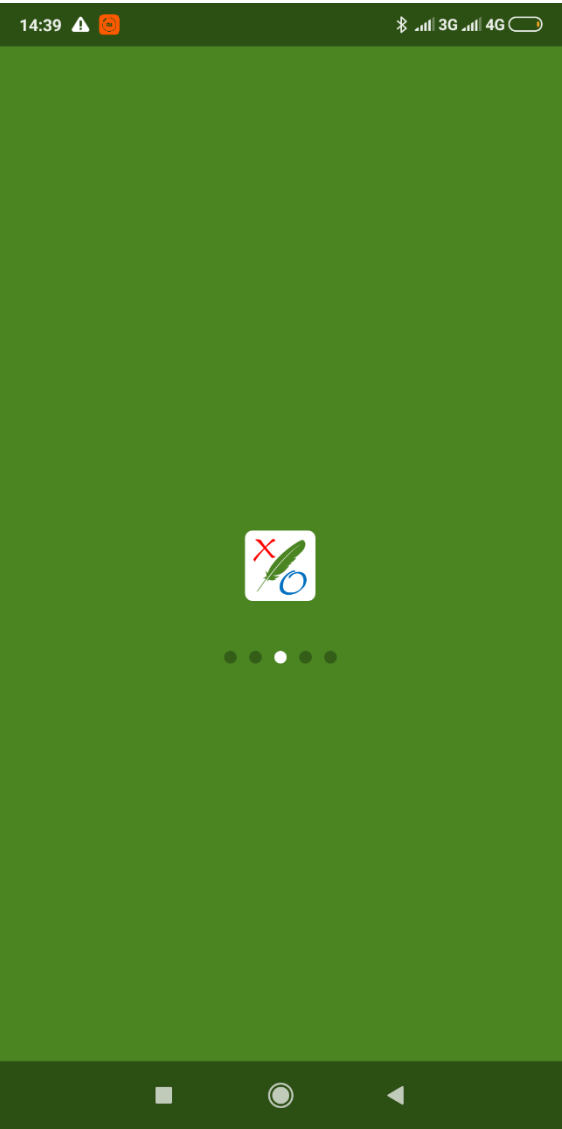


Mục tiêu

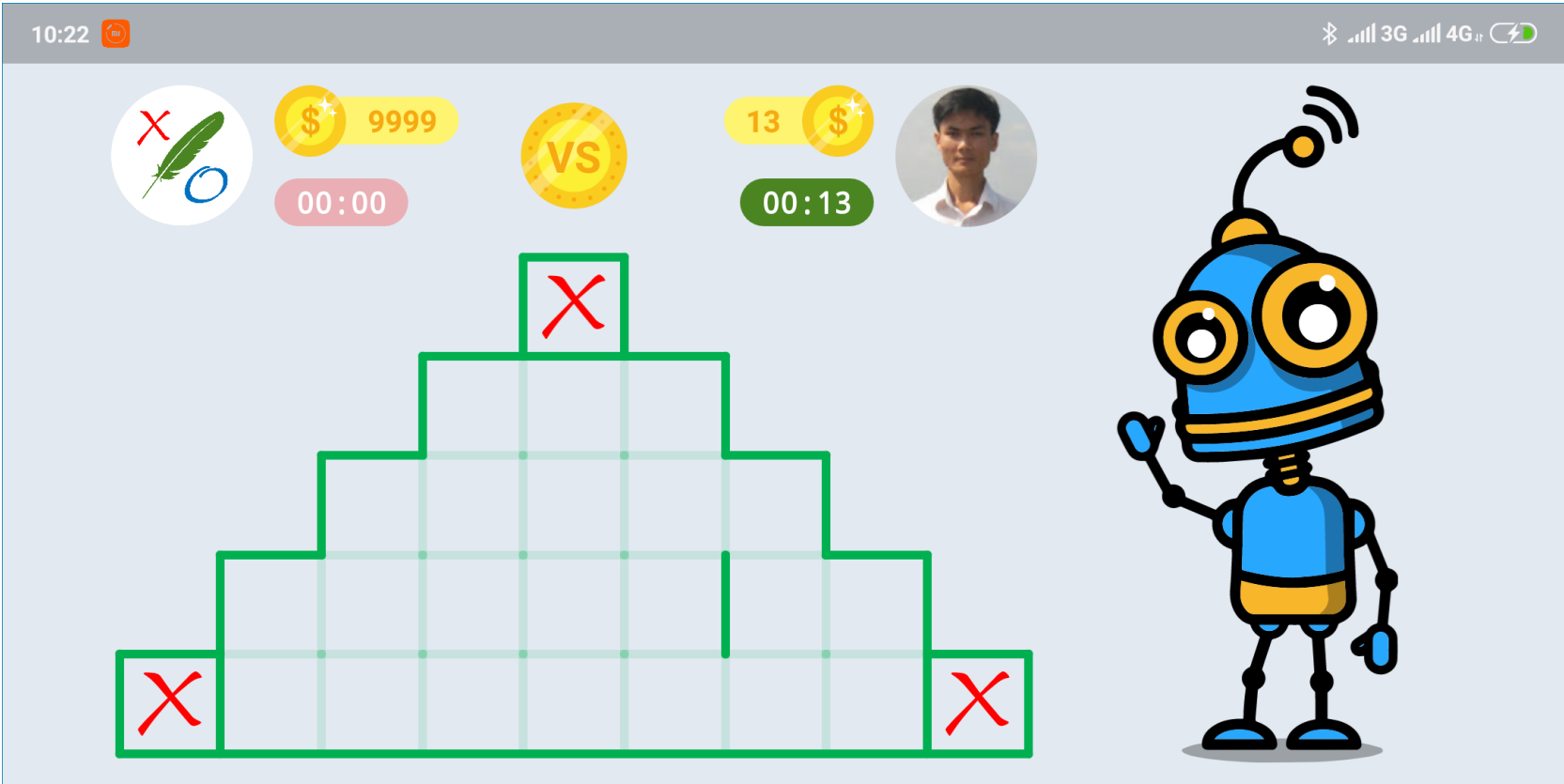
” Không phải xây dựng ứng dụng; xây dựng **nhóm** có thể **tạo** ra **ứng dụng** và ứng dụng **tốt** theo hướng phát triển ứng dụng **chuyên nghiệp**.



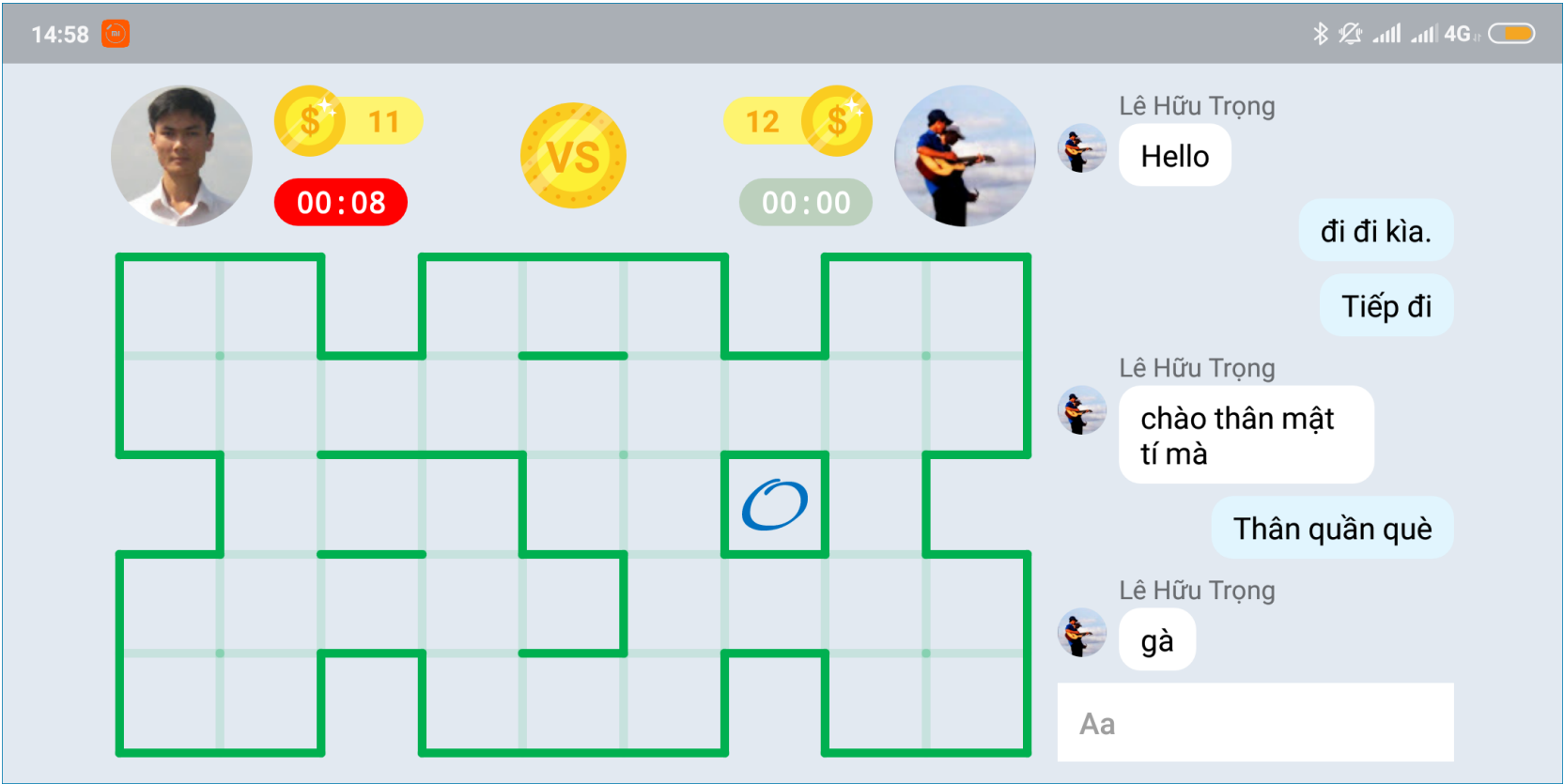
Giao diện



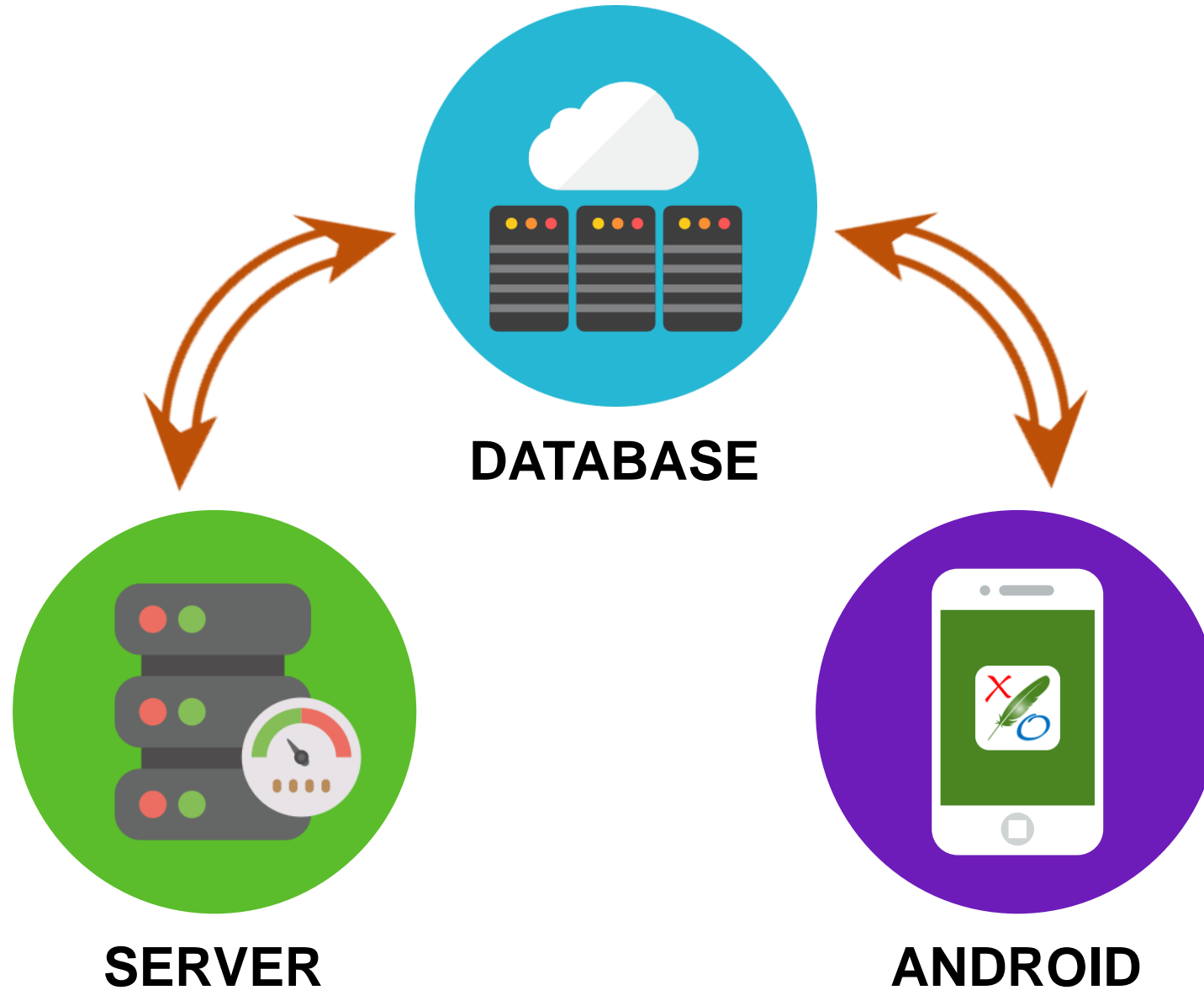
Giao diện



Giao diện



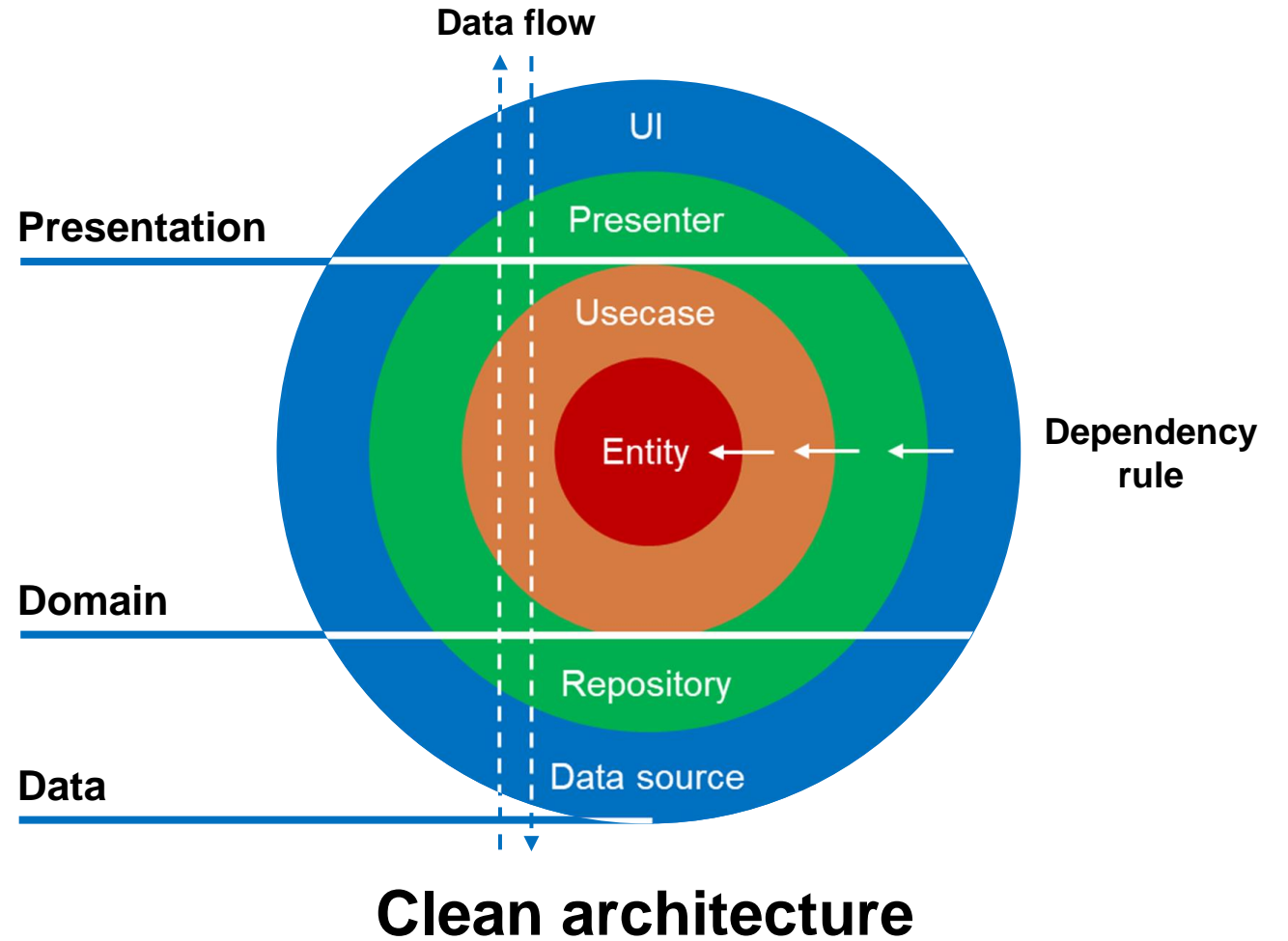
Kiến trúc hệ thống



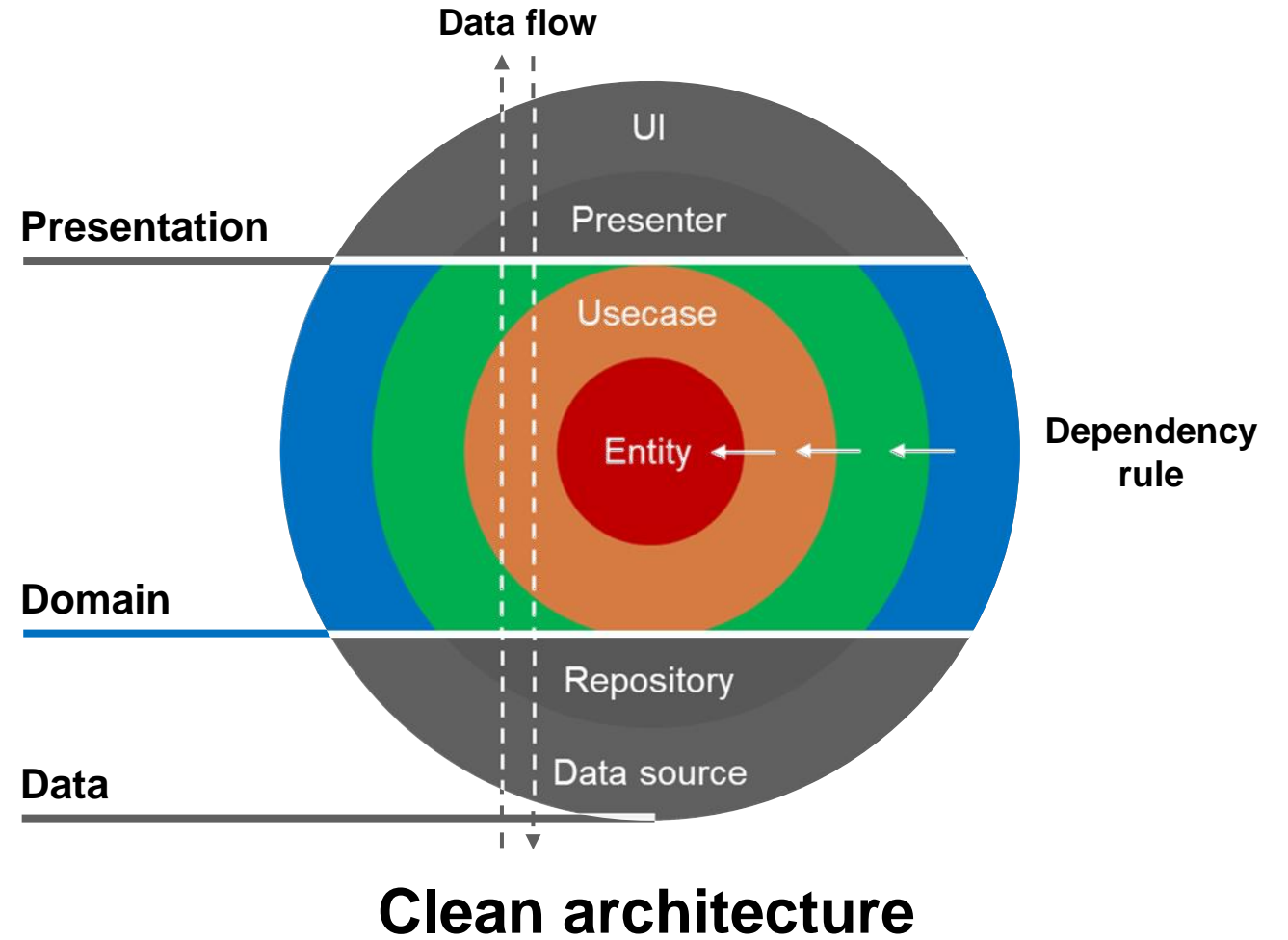
Kiến trúc ứng dụng



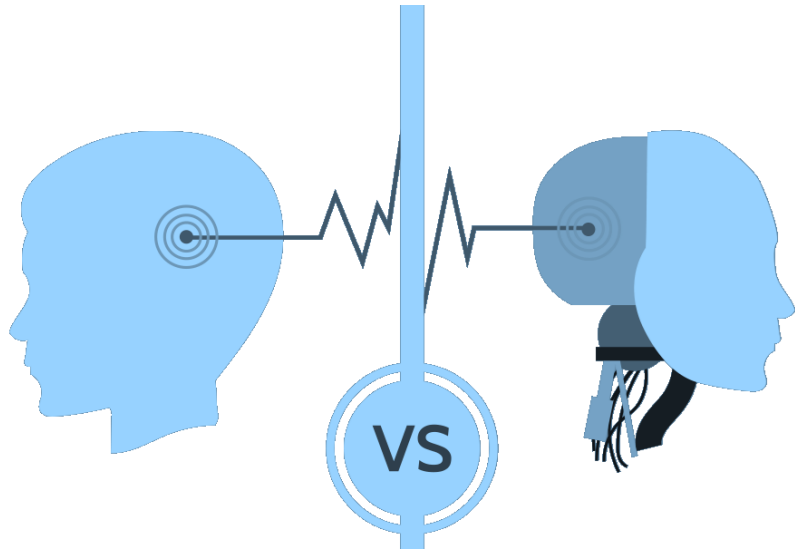
Uncle Bob



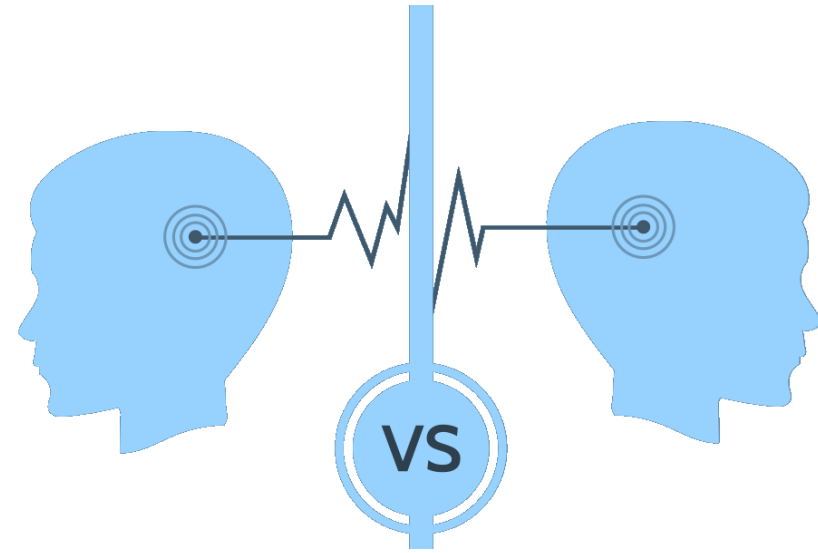
Domain



Domain



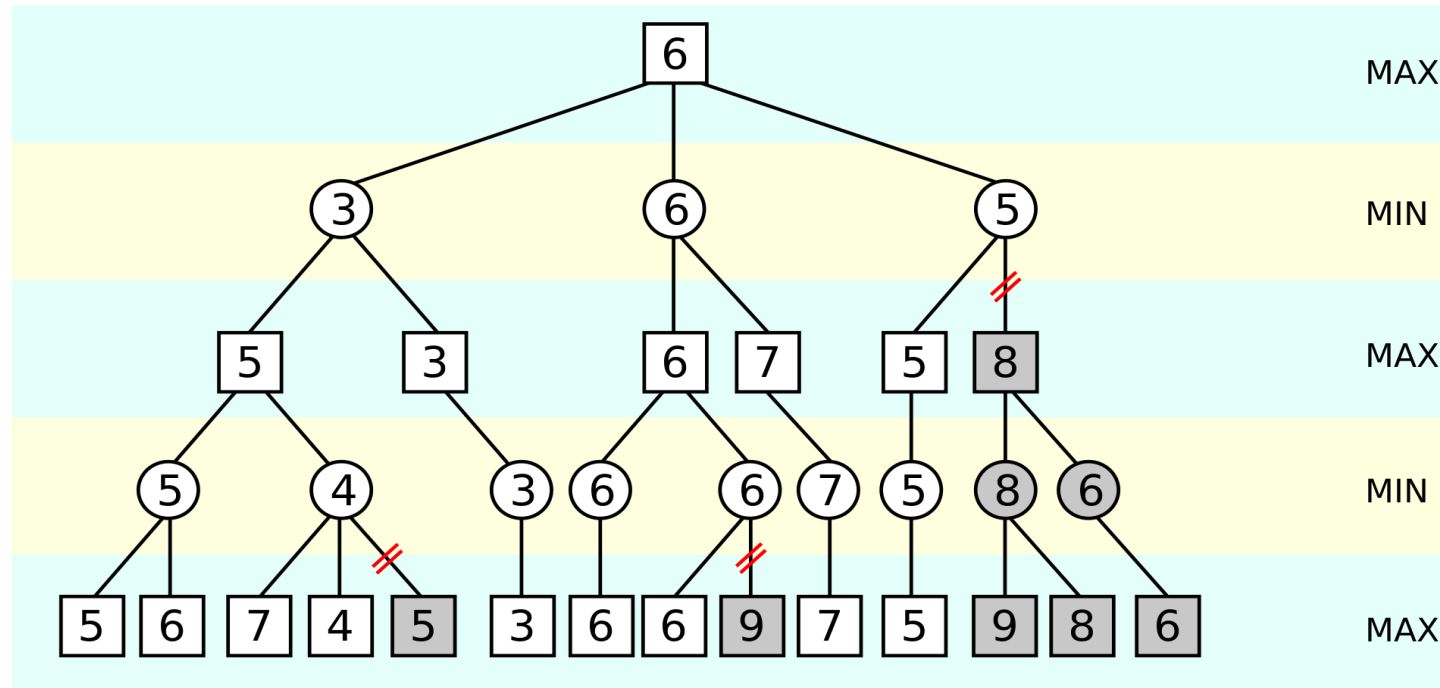
Chơi với máy



Chơi với người

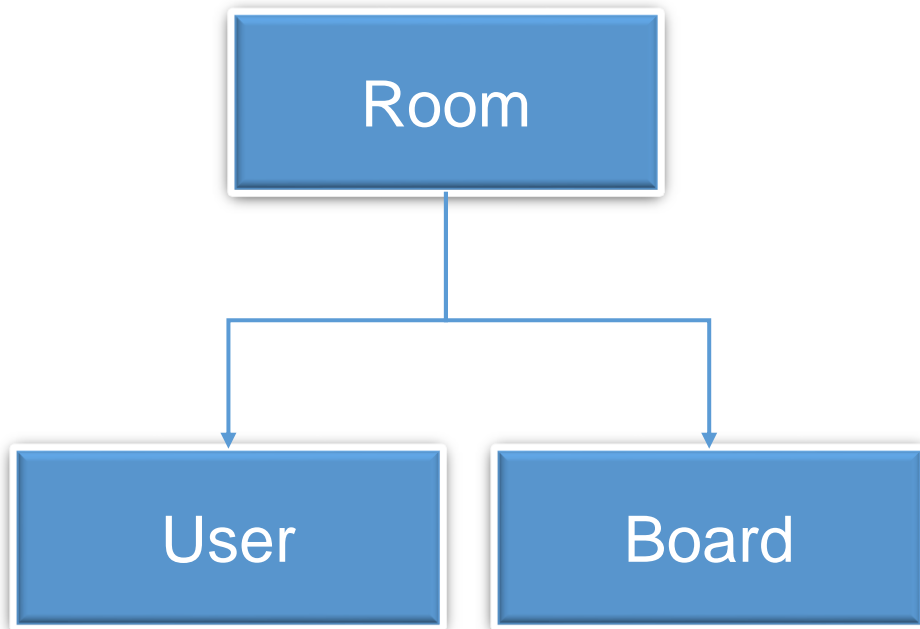
Chơi với máy

- Độ sâu tìm kiếm: 3
- Thời gian tối thiểu giữa hai nước đi: 0.5 giây



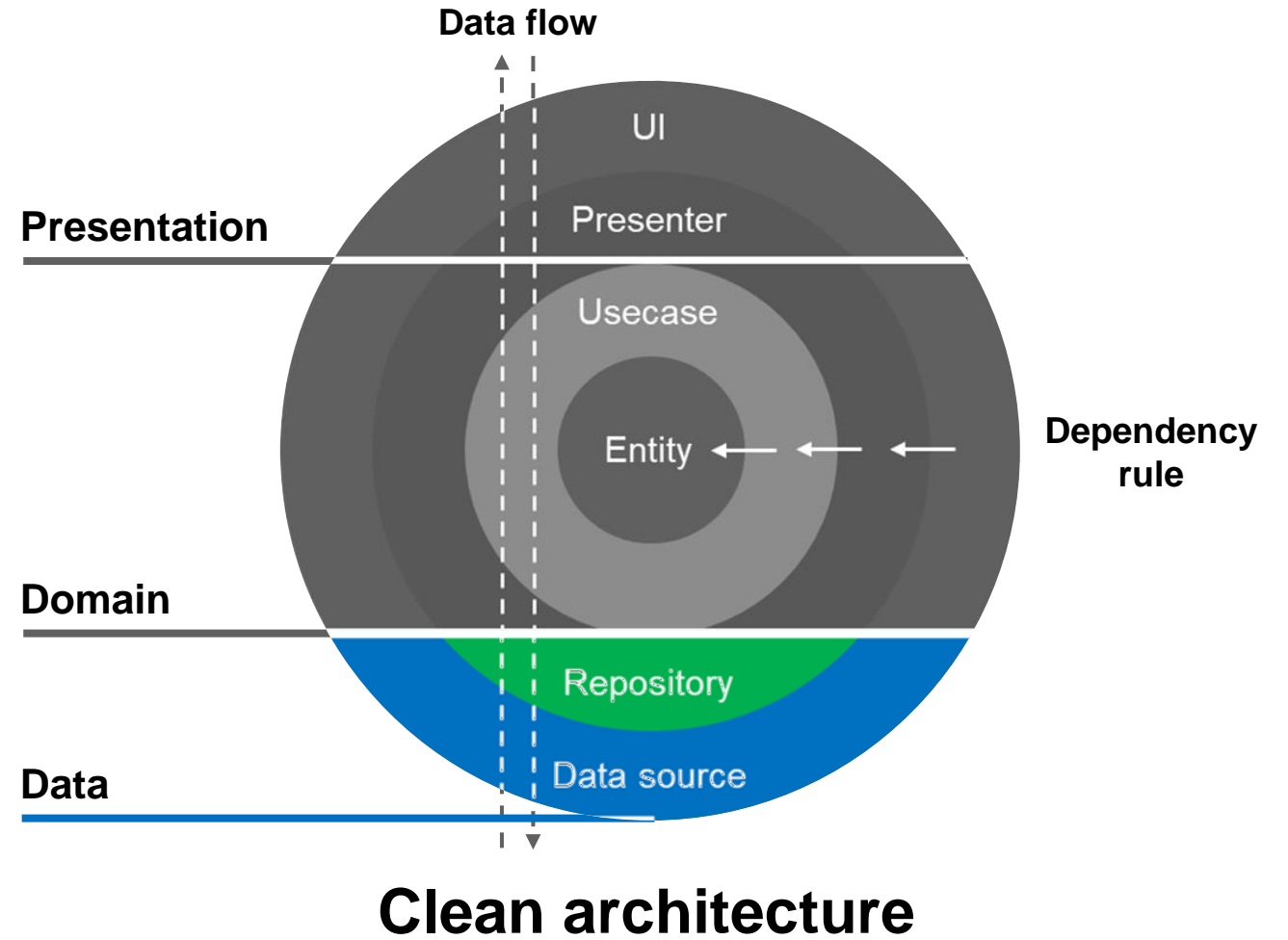
Alpha-Beta Search

Chơi với người



CREATE	• Phòng vừa được tạo
RANDOM	• Phòng được thêm một bàn cờ ngẫu nhiên
JOIN	• Người chơi thứ hai vào phòng
START	• Chọn ngẫu nhiên người đi trước
MOVE	• Có một nước đi mới
END	• Kết thúc ván chơi
LEAVE	• Người chơi thứ hai rời phòng
DESTROY	• Người tạo phòng rời phòng

Data



Các loại lưu trữ



CACHE



LOCAL



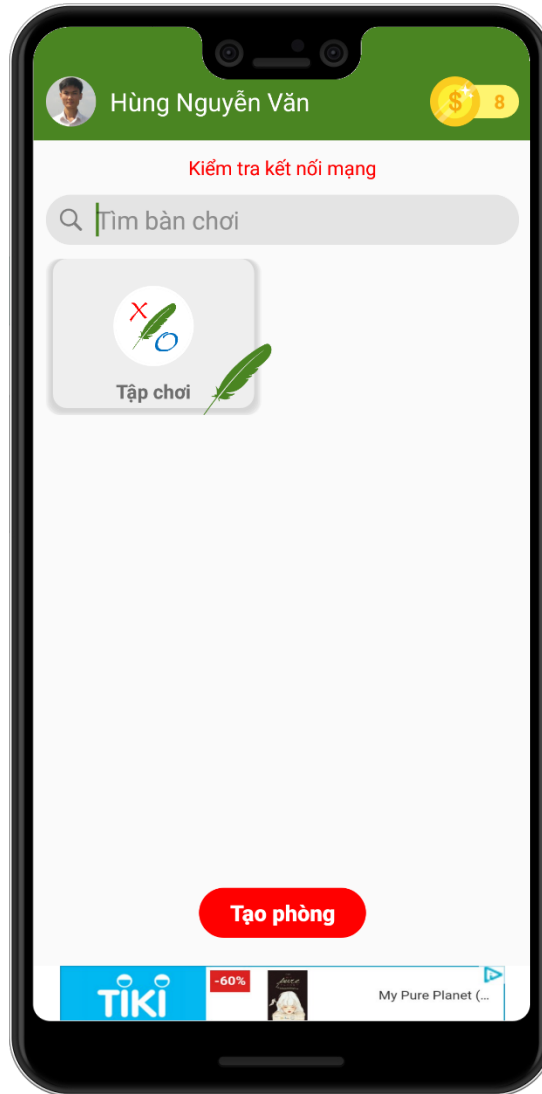
NETWORK

Cache



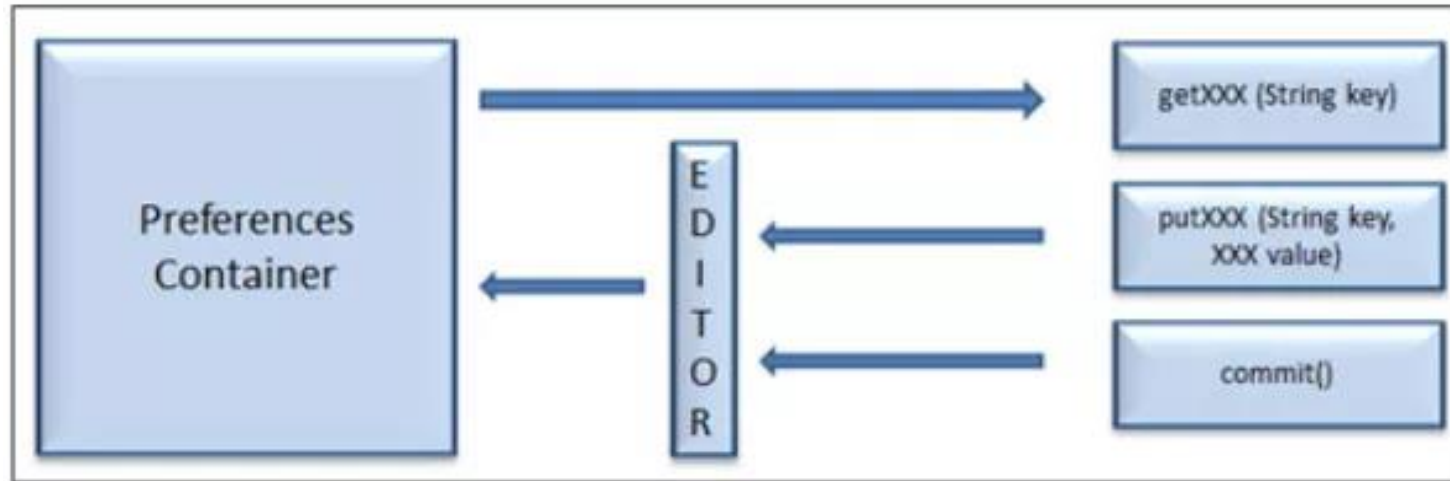
Cache giúp nhận biết **lần đầu sử dụng app** và hiển thị **hướng dẫn**

Cache



Cache lưu thông tin User để hiển thị khi **offline**

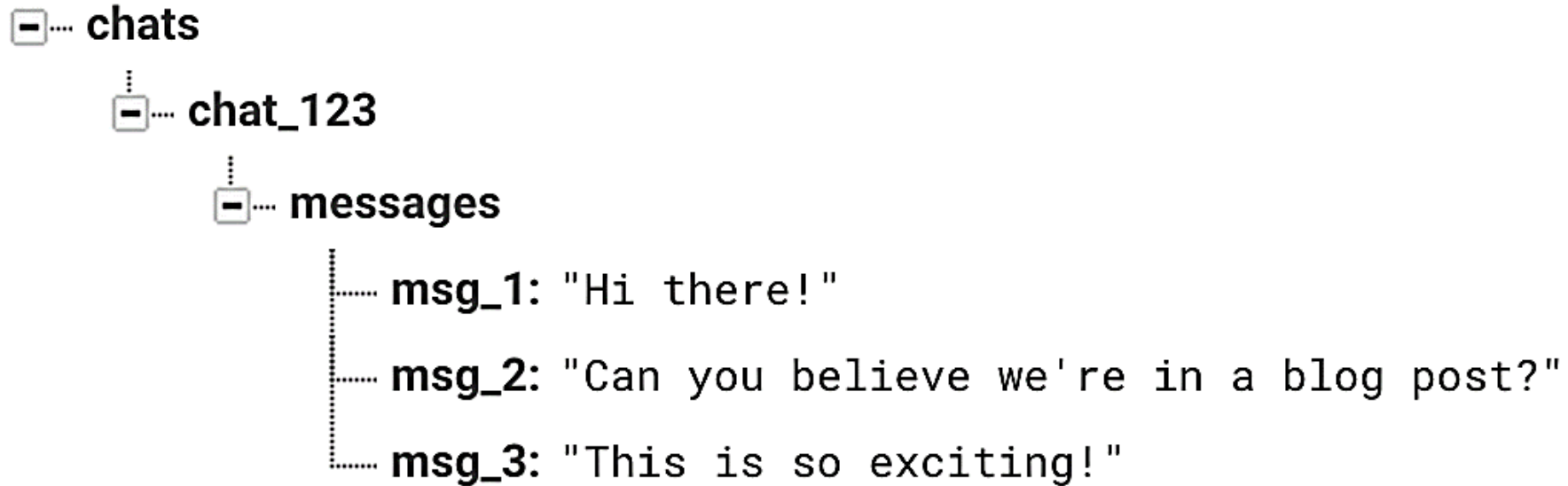
Cache



SharedPreferences

- Lưu trữ dữ liệu dạng <key, value>
- Thao tác qua các phương thức **get** từ Container và **put** thông qua Editor (putString, getLong, ...)

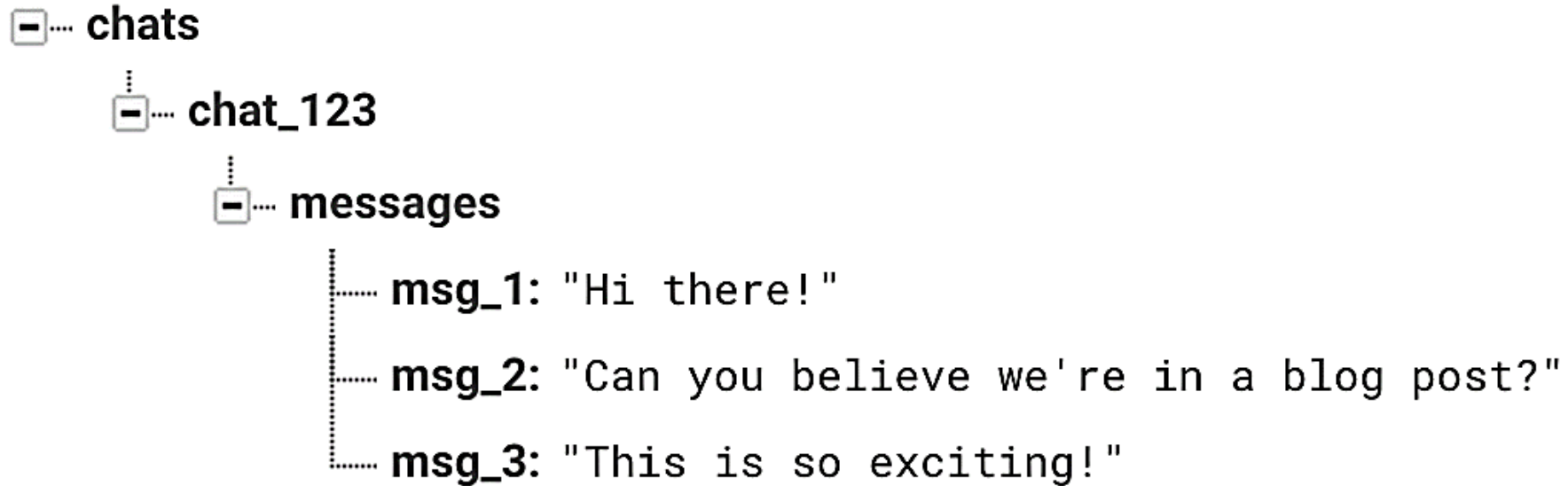
Network



Firestore

- Lưu trữ dữ liệu dạng <key, value>
- Đi đến một node với ***getReference()*** và ***child()***
- Lưu dữ liệu với ***setValue()***

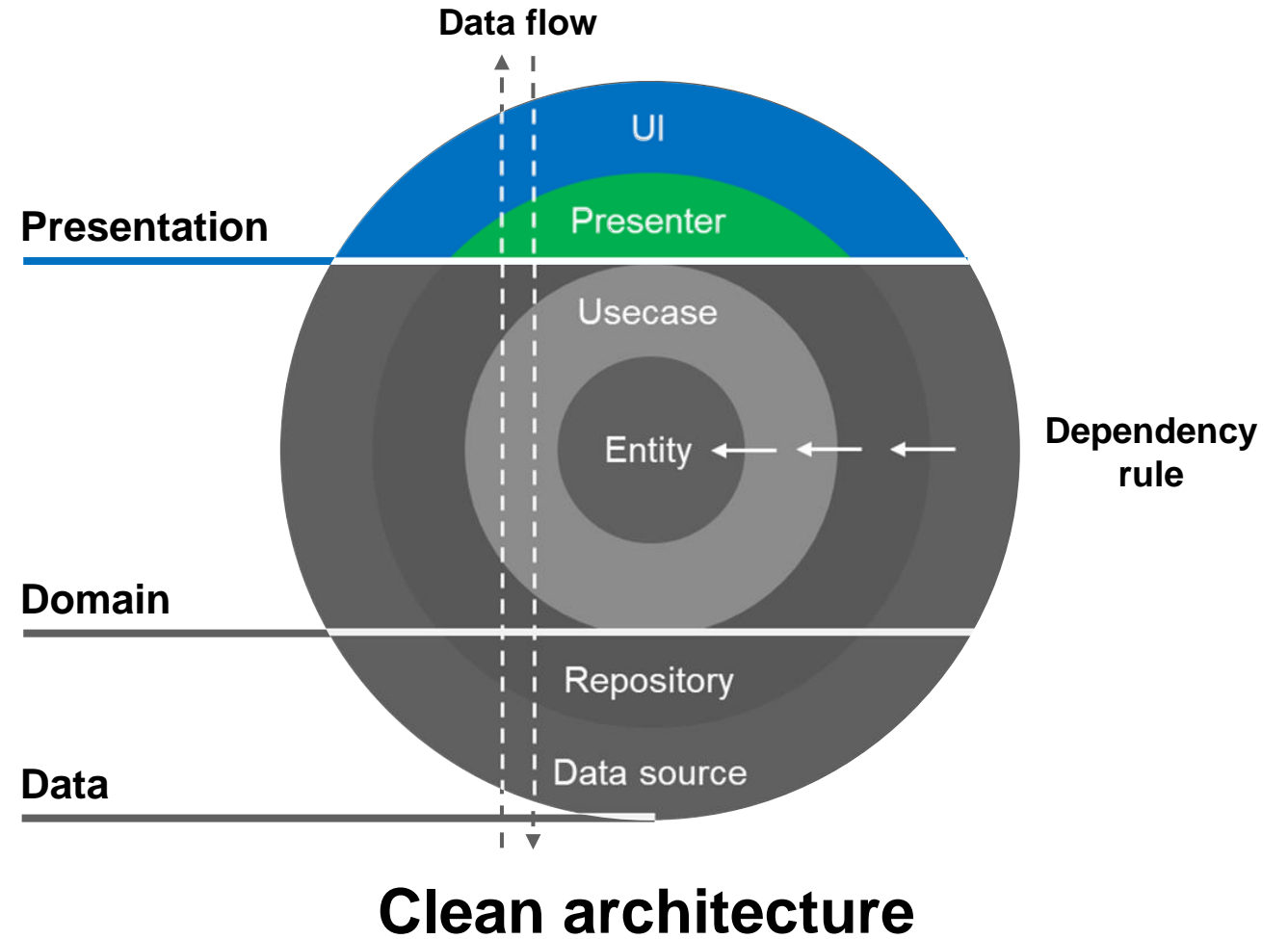
Network



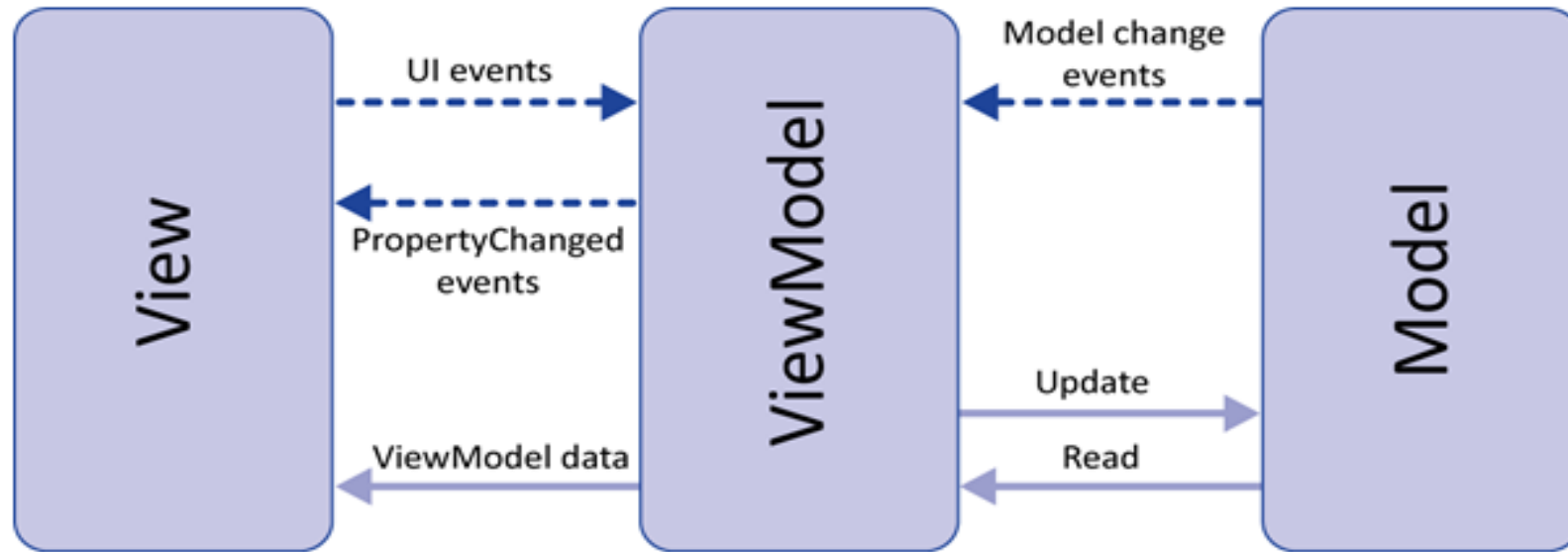
Firestore

- Lấy dữ liệu và lắng nghe sự thay đổi của dữ liệu theo ba cách:
 - ***addValueEventListener()***
 - ***addListenerForSingleValueEvent()***
 - ***addChildEventListener()***

Presentation



Mô hình MVVM



MVVM architecture

MVVM là kiến trúc giúp tăng cường phân tách mối liên quan giữa tầng giao diện người dùng với tầng xử lý business logic.

Data Binding

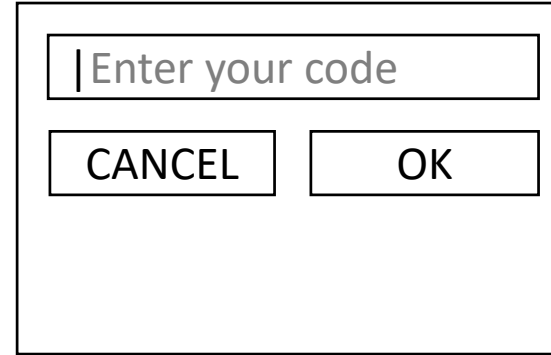
Tìm kiếm View có độ phức tạp lớn khi **không** dùng Data binding

layout.xml

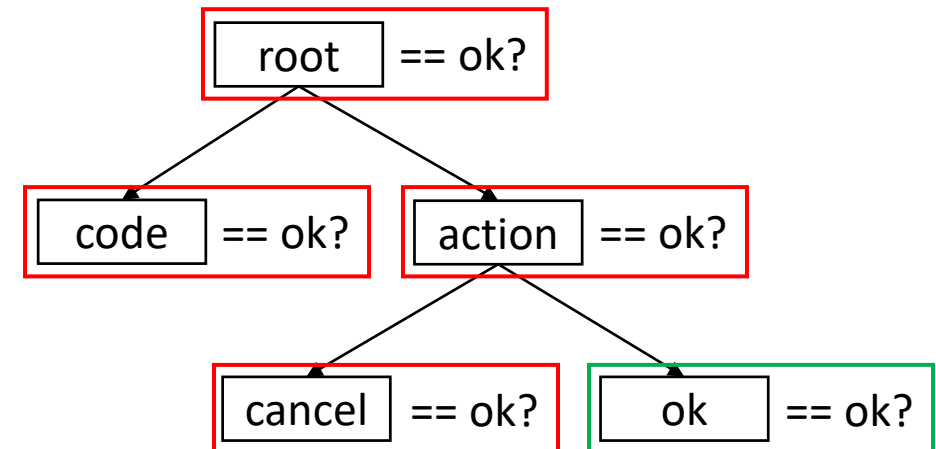
```
<RelativeLayout
    android:id="@+id/root">

    <TextView
        android:id="@+id/code"/>

    <LinearLayout
        android:id="@+id/action">
        <Button
            android:id="@+id/cancel"/>
        <Button
            android:id="@+id/ok"/>
    </LinearLayout>
</RelativeLayout>
```



findViewById(R.id.ok)



Data Binding

Tìm kiếm View có độ phức tạp **O(1)** khi dùng Data binding

layout.xml

```
<layout>
  <RelativeLayout
    android:id="@+id/root">

    <TextView
      android:id="@+id/code"/>

    <LinearLayout
      android:id="@+id/action">
      <Button
        android:id="@+id/cancel"/>
      <Button
        android:id="@+id/ok"/>
    </LinearLayout>
  </RelativeLayout>
</layout>
```



LayoutBinding.java

```
public abstract class LayoutBinding {
    @NonNull
    public final RelativeLayout root;

    @NonNull
    public final TextView code;

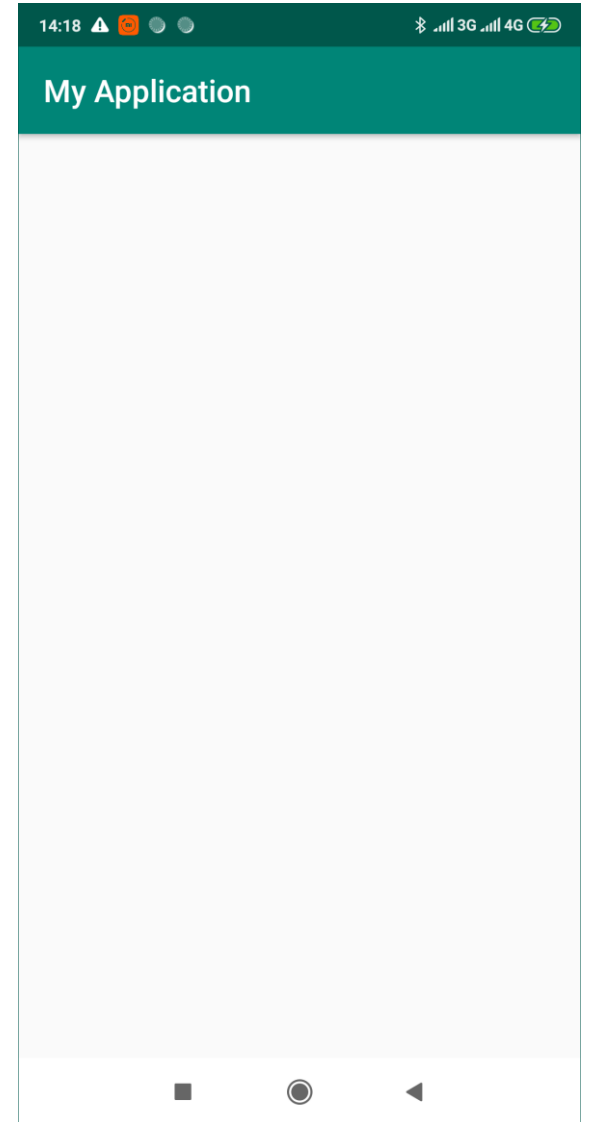
    @NonNull
    public final LinearLayout action;

    @NonNull
    public final Button cancel;

    @NonNull
    public final Button ok;
}
```

Custom layout, custom draw

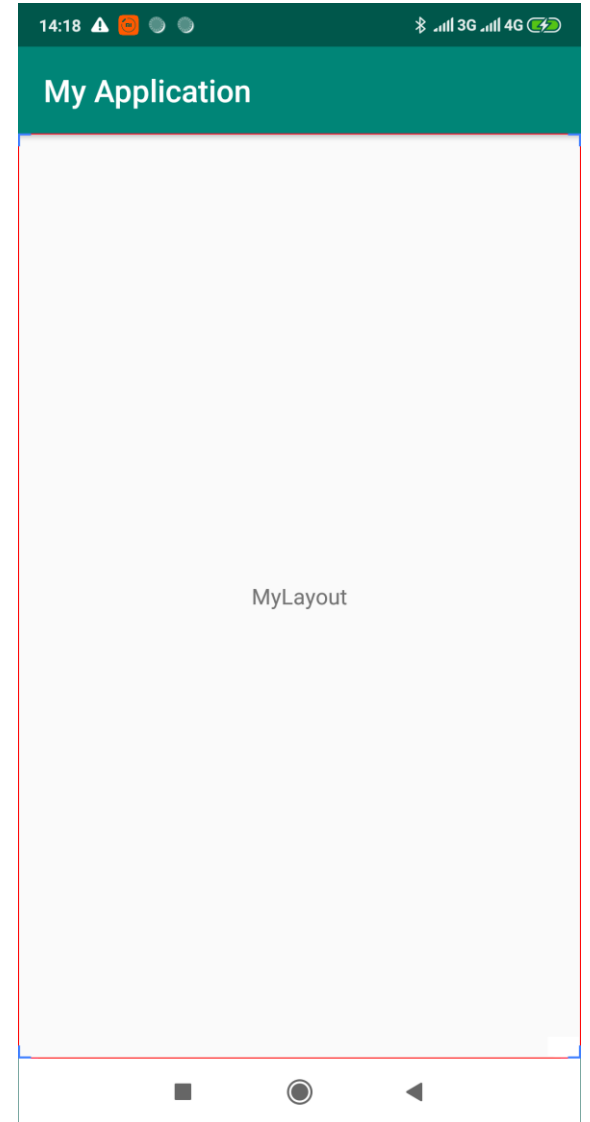
```
public class MyLayout extends FrameLayout {
```



Custom layout, custom draw

```
public class MyLayout extends FrameLayout {
```

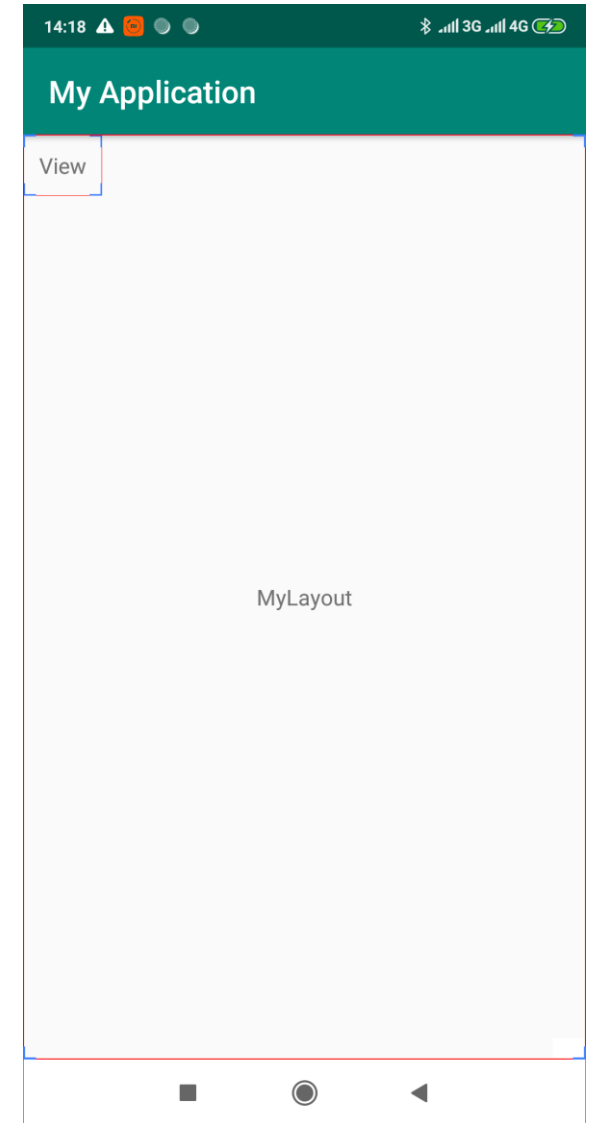
}



Custom layout, custom draw

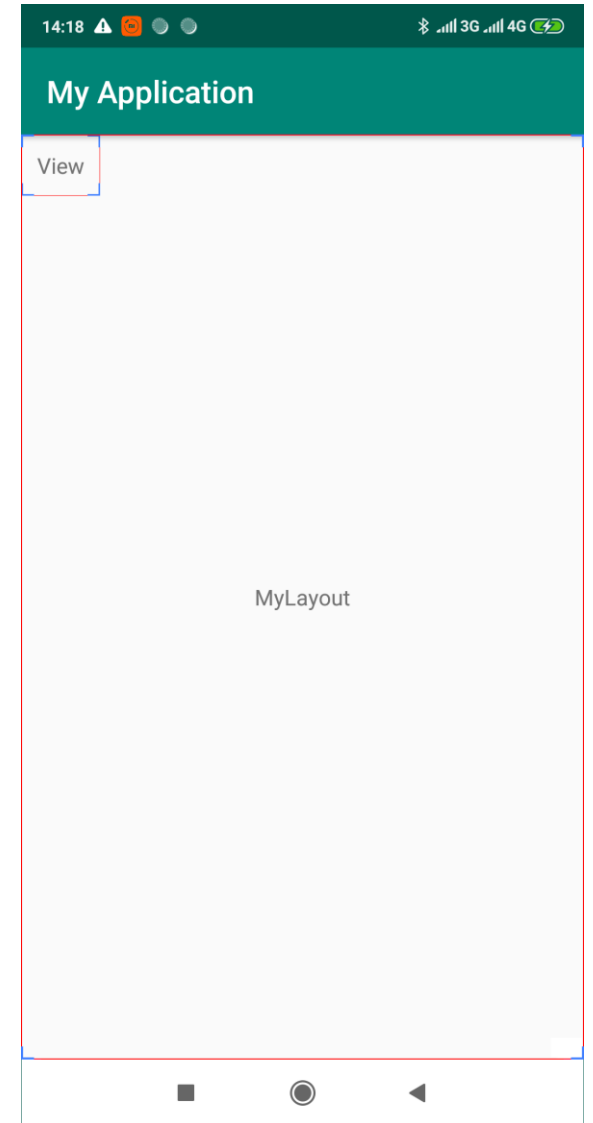
```
public class MyLayout extends FrameLayout {
```

}



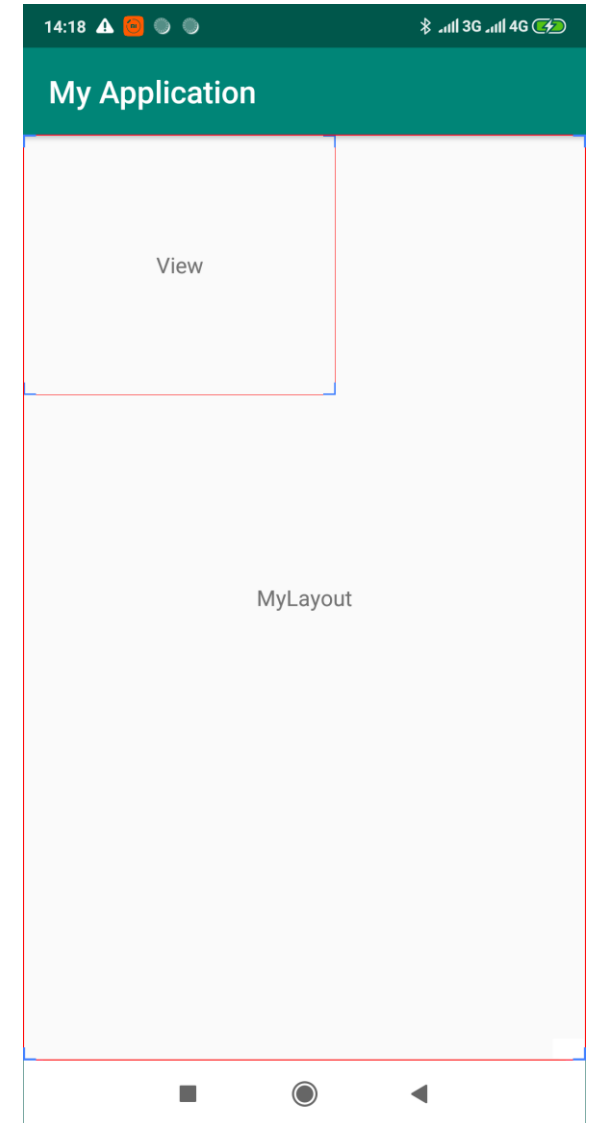
Custom layout, custom draw

```
public class MyLayout extends FrameLayout {  
    @Override  
    protected void onMeasure(int widthSpec,  
                             int heightSpec) {  
  
    }  
    @Override  
    protected void onLayout(boolean changed,  
                            int l, int t, int r, int b) {  
  
    }  
}
```



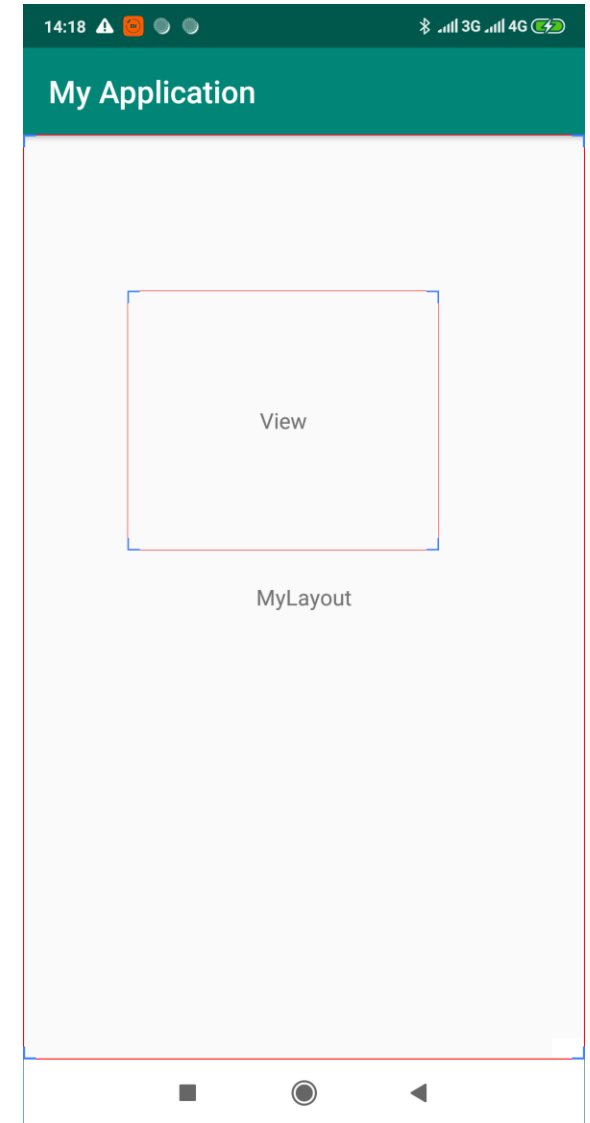
Custom layout, custom draw

```
public class MyLayout extends FrameLayout {  
    @Override  
    protected void onMeasure(int widthSpec,  
                             int heightSpec) {  
        view.measure(  
            600 | MeasureSpec.EXACTLY,  
            500 | MeasureSpec.EXACTLY  
        );  
    }  
    @Override  
    protected void onLayout(boolean changed,  
                             int l, int t, int r, int b) {  
    }  
}
```



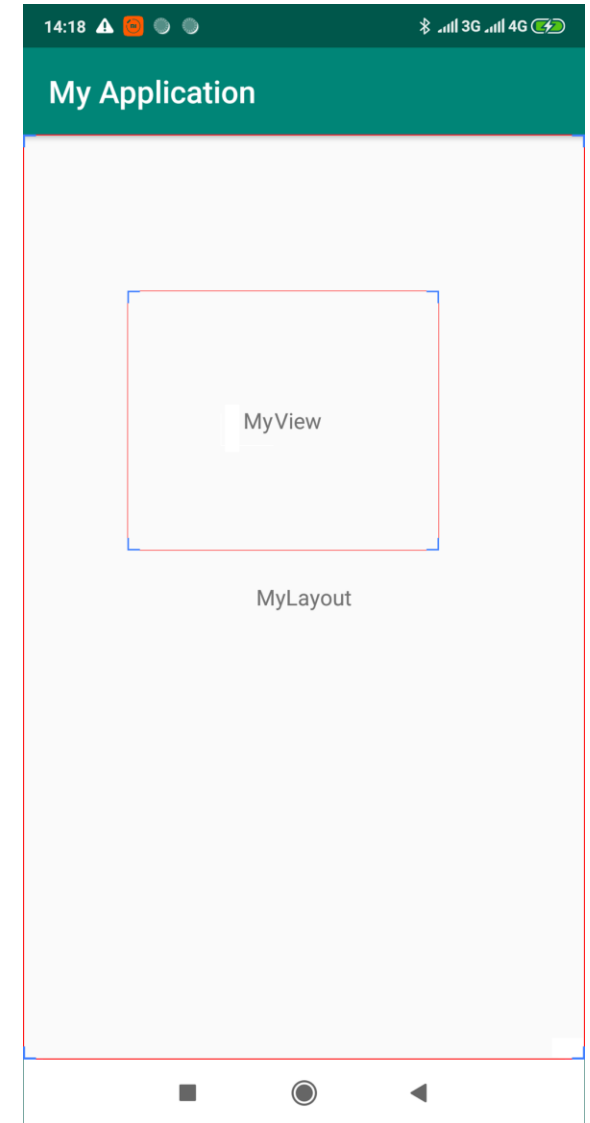
Custom layout, custom draw

```
public class MyLayout extends FrameLayout {  
    @Override  
    protected void onMeasure(int widthSpec,  
                             int heightSpec) {  
        view.measure(  
            600 | MeasureSpec.EXACTLY,  
            500 | MeasureSpec.EXACTLY  
        );  
    }  
    @Override  
    protected void onLayout(boolean changed,  
                             int l, int t, int r, int b) {  
        view.layout(200, 300, 200 + 600, 300 + 500);  
    }  
}
```



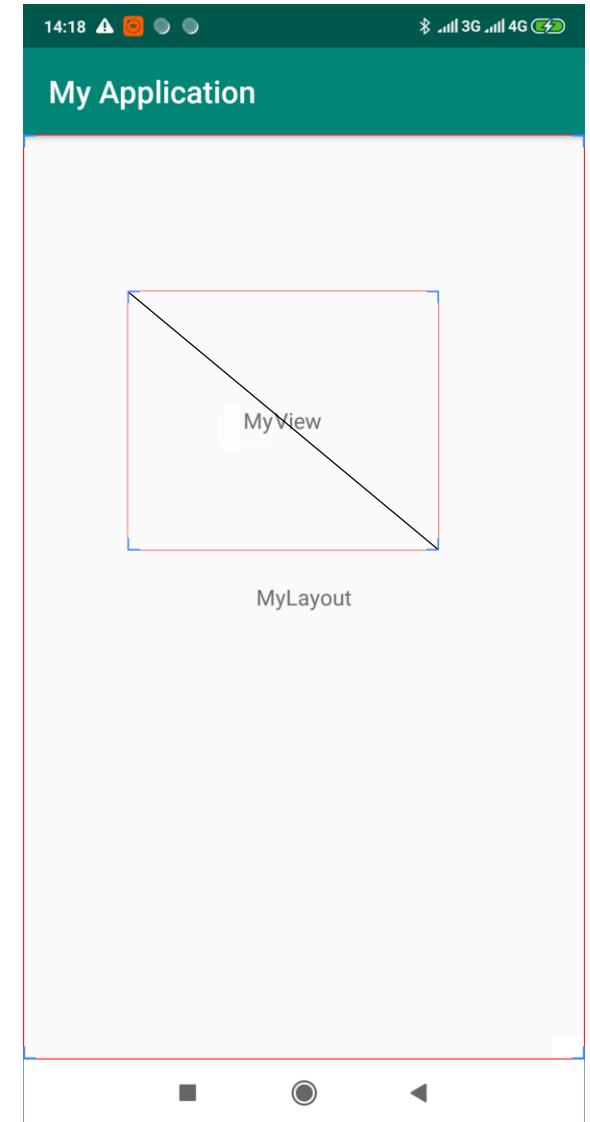
Custom layout, custom draw

```
public class MyView extends View {  
    @Override  
    protected void onDraw(Canvas canvas) {  
  
    }  
}
```

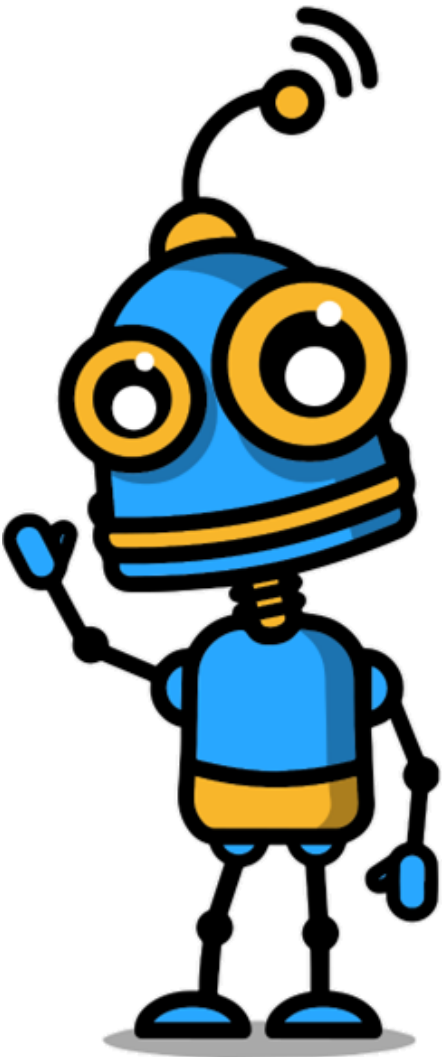


Custom layout, custom draw

```
public class MyView extends View {  
    @Override  
    protected void onDraw(Canvas canvas) {  
        canvas.drawLine(0, 0, 600, 500, new Paint())  
    }  
}
```



Drawable



A dark, atmospheric promotional image for the Gears of War franchise. It features several characters in heavy, futuristic armor. In the foreground, a character (likely Marcus Fenix) is shown in profile, holding a large, complex weapon. Other characters are visible in the background, some in action poses. The scene is set in a desolate, war-torn environment with a cloudy sky and distant structures. The overall tone is gritty and cinematic.

Tương lai

Tương lai



A dark, atmospheric background image from the Gears of War franchise. It features several characters in heavy, detailed armor. In the foreground, a character (likely Dom Monaghan) is shown from the waist up, holding a large, futuristic weapon. Other characters are visible in the background, some in combat stances. The scene is set in a dark, industrial environment with smoke and debris.

Chúc các bạn chơi game vui vẻ!