

RdP engine – versão 0.5 Suporte a múltiplas instâncias

Exemplo: na rede Player, colisão com instância de Npc1 (enemy.pflow) gera “resposta” de retorno para a instância que gerou a colisão.

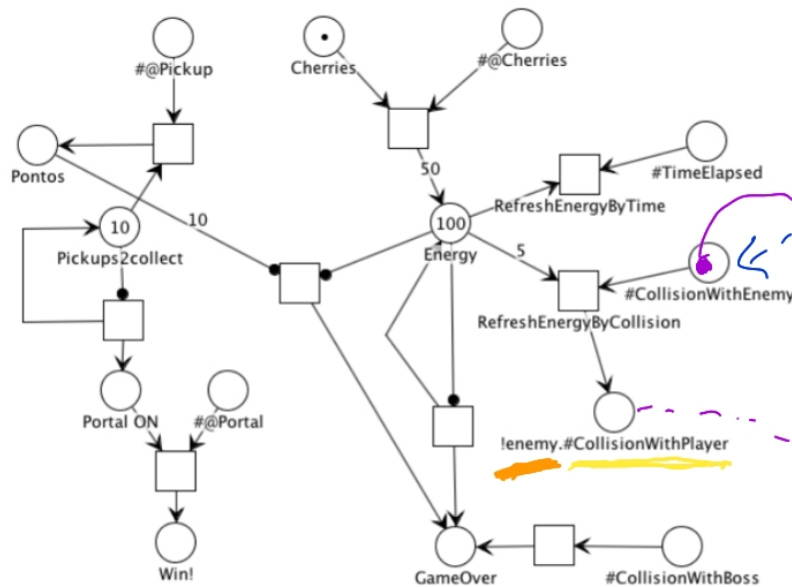
No PlayerController.cs :

```
countText.text = "Pontos:" + pontos.Tokens.ToString();
energyText.text = "Energy:" + energy.Tokens.ToString();
}
<# Event function
private void OnCollisionEnter2D(Collision2D other)
{
    if (other.gameObject.CompareTag("Enemy"))
    {
        PetriNet pn = other.gameObject.GetComponent<Npc1Controller>().npc;
        player.GetPlaceByLabel("#CollisionWithEnemy").AddTokens( quantity: 1, pn);
    }
}
/*
instead of:
if (other.gameObject.CompareTag("Enemy"))
    player.GetPlaceByLabel("#CollisionWithEnemy").Tokens = 1;
*/
*/
```

é a rede da
instância que
causou a
colisão
é propagada para
o lugar

Obs.: empregar método *AddTokens* ao invés de *Tokens = 1* (usando atribuição não é possível propagar identidade da rede externa)

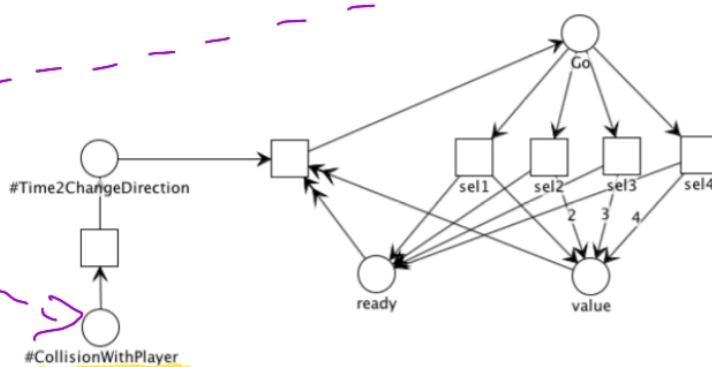
player.pflow



marca(token) adicionada
fica associada a
rede e instância
externa

Obs.: lugares externos são identificados com *!nomeRedeExterna.nomeLugar*

enemy.pflow



token é propagado para
rede e instância externa

