Rapport de sécurité

Injection SQL

Pour une bonne sécurité pour éviter les injection SQL en important du code via le formulaire.

Une Foix l'entité généré nous devons ajouter un htlmlspecialchart decript les balise pour éviter tout injection d'html en base de données donc éviter tout injection de script

```php
public function setName(?string $name): static
{
    $this->name =htmlspecialchars( $name);

    return $this;
}
```

On hash le password pour éviter qu'il soit en claire

```php
$user->setPassword(
    $userPasswordHasher->hashPassword(
        $user,
        $form->get('plainPassword')->getData()
    )
);
```

On vérifie les informations donner dans le formulai s'ils sont valides

```php
if ($form->isSubmitted() && $form->isValid()) {
    $slug = $slugger->slug($game->getName())->lower();
    $game->setSlug($slug);
    $entityManager->persist($game);
    $entityManager->flush();
    return $this->redirectToRoute( route: '/game');
}
```

On ne cible pas l'id mai une empreinte qui est le slug pour éviter de pouvoir dans l'url de pointer n'importe qu'elle id

```
{% if is_granted('ROLE_ADMIN') %}
    <a href="{{ path('app_game_edit', {'slug': game.slug}) }}">Editer</a>
{% endif %}
```

On attribue des sécurité pour les route qu'on ne veux pas que les utilisateur qui sois pas connecté ou pas de privilège

```
  - { path: ^/game/admin, roles: ROLE_ADMIN }
  - { path:  ^/game, roles: ROLE_USER }
```