

Documentation du projet JRPG



1. Introduction

Le projet JRPG (Jeu de Rôle en console) est une application console créée en C#. Le but est de proposer un jeu de combat tour par tour entre deux équipes de personnages. Chaque joueur construit son équipe et sélectionne les actions de ses personnages pour vaincre l'équipe adverse.

2. Structure du projet

Le projet suit une architecture bien organisée avec une séparation claire entre les fichiers et les classes. Voici les principaux dossiers et fichiers :

- **Enums** : Contient les énumérations telles que ActionType, TargetType, TypeDamage, TypeOfArmor.
- **Models** : Contient les classes représentant les personnages et éléments du jeu, comme Character, Warrior, Mage, etc.
- **Services** : Contient des services et utilitaires pour le fonctionnement du jeu, tels que Menu, Utils, ClassConfigLoader.
- **Resources** : Contient un fichier de configuration classes.json pour les configurations des personnages.

3. Classes et Responsabilités

3.1 Classes principales

Character (Classe de base)

- **Responsabilité** : Représente un personnage générique avec des attributs et méthodes communes à toutes les classes de personnage.
- **Attributs** : Nom, points de vie, puissance d'attaque physique et magique, type d'armure, probabilités d'esquive/parade/résistance aux sorts, vitesse, etc.
- **Méthodes** :
 - Defend: Gère la défense contre une attaque.
 - ChoiceAction: Demande à l'utilisateur de choisir une action pour le personnage.
 - ToString: Retourne un résumé des statistiques du personnage.

Warrior (Guerrier)

- **Caractéristiques** :

Statistique	Valeur
Points de vie (HP) max	100
Attaque physique	50
Attaque magique	0
Type d'armure	Plates
Esquive	5%
Parade	25%
Résistance magique	10%
Vitesse	50

- **Compétences** :

Nom	Description	Coût en Mana	Cooldown	Effet
Heroic Strike	Attaque physique infligeant 100% de la puissance d'attaque physique.	0	1 tour	Dégâts physiques
Battle Cry	Augmente l'attaque physique de tous les alliés de 25.	0	2 tours	Buff
Whirlwind	Attaque tous les ennemis pour 33% de la puissance d'attaque physique.	0	2 tours	Dégâts physiques

Mage

- **Caractéristiques :**

Statistique	Valeur
Points de vie (HP) max	60
Attaque physique	0
Attaque magique	75
Type d'armure	Fabric
Esquive	5%
Parade	5%
Résistance magique	25%
Vitesse	75
Mana	100

- **Compétences :**

Nom	Description	Coût en Mana	Cooldown	Effet
Frost Bolt	Attaque magique infligeant 100% de la puissance d'attaque magique.	15	1 tour	Dégâts magiques
Frost Barrier	Réduit les dégâts des 2 prochaines attaques.	25	2 tours	Buff
Blizzard	Attaque tous les ennemis pour 50% de la puissance d'attaque magique.	25	2 tours	Dégâts magiques
Spell Return	Retourne la prochaine attaque magique reçue.	25	1 tour	Buff
Mana Burn	Réduit de moitié le mana d'un ennemi.	20	3 tours	Debuff

Paladin

- **Caractéristiques :**

Statistique	Valeur
Points de vie (HP) max	95
Attaque physique	40
Attaque magique	40
Type d'armure	Mesh
Esquive	5%
Parade	10%
Résistance magique	20%
Vitesse	75
Mana	60

- **Compétences :**

Nom	Description	Coût en Mana	Cooldown	Effet
Crusader Strike	Attaque physique infligeant 100% de la puissance d'attaque physique.	5	1 tour	Dégâts physiques
Judgement	Attaque magique infligeant 100% de la puissance d'attaque magique.	10	1 tour	Dégâts magiques
Bright Flash	Soigne un allié pour 125% de la puissance d'attaque magique.	25	1 tour	Soin

Thief (Voleur)

- **Caractéristiques :**

Statistique	Valeur
Points de vie (HP) max	80
Attaque physique	55
Attaque magique	0
Type d'armure	Leather
Esquive	15%
Parade	25%
Résistance magique	25%
Vitesse	100

- **Compétences :**

Nom	Description	Coût en Mana	Cooldown	Effet
Low Blow	Attaque physique infligeant 100% de la puissance d'attaque physique (150% si la cible est blessée).	0	1 tour	Dégâts physiques
Escape	Augmente les chances d'esquive et de résistance aux sorts de 20%.	0	1 tour	Buff

Priest (Prêtre)

- **Caractéristiques :**

Statistique	Valeur
Points de vie (HP) max	70
Attaque physique	0
Attaque magique	65
Type d'armure	Fabric
Esquive	10%
Parade	0%
Résistance magique	20%
Vitesse	70
Mana	100

- **Compétences :**

Nom	Description	Coût en Mana	Cooldown	Effet
Punishment	Attaque magique infligeant 75% de la puissance d'attaque magique.	15	1 tour	Dégâts magiques
Circle of Care	Soigne toute l'équipe pour 75% de la puissance d'attaque magique.	30	2 tours	Soin

3.2 Classes utilitaires

Menu

- **Responsabilité :** Gère l'interaction utilisateur, la navigation dans les menus, la création des équipes et le lancement des combats.

Utils

- **Responsabilité :** Fournit des méthodes utilitaires comme la sélection des options, la gestion des noms uniques, et la constitution des équipes.

ClassConfigLoader

- **Responsabilité :** Charge les configurations des personnages à partir du fichier classes.json.

4. Système de Jeu

Constitution des équipes

- Chaque joueur sélectionne 3 personnages pour constituer son équipe.
- Les informations sur chaque personnage sont affichées pour aider à faire un choix stratégique.

Combat

1. **Sélection des actions** : Chaque joueur choisit les actions pour tous ses personnages.
2. **Ordre d'exécution** : Les actions sont effectuées par ordre de vitesse des personnages.
3. **Calcul des dégâts** : Les dégâts sont réduits en fonction de l'armure, de l'esquive, de la parade et de la résistance aux sorts.
4. **Mise à jour des cooldowns** : Les compétences utilisent un système de recharge.

Fin de partie

- La partie se termine lorsqu'une équipe n'a plus de personnages vivants.
- Le joueur ayant encore des personnages en vie est déclaré gagnant.

5. Instructions pour l'utilisateur

1. Lancer le programme via la console.
2. Sélectionner "Start a new game".
3. Créer une équipe en choisissant les classes et noms des personnages.
4. Suivre les instructions pour choisir les actions pendant les tours.
5. Observer les messages affichés pour suivre l'état de la partie.

6. Règles et Calculs Importants

- **Esquive** : Annule complètement les dégâts.
- **Parade** : Réduit les dégâts physiques de 50%.
- **Résistance aux sorts** : Annule les dégâts magiques.
- **Réduction par armure** : Varie en fonction du type d'armure et du type d'attaque.

7. Ajout d'un nouveau personnage

L'ajout d'un nouveau personnage dans le projet JRPG implique plusieurs étapes. Cela inclut la création de la classe du personnage, l'ajout de ses compétences spécifiques, la mise à jour du fichier de configuration `classes.json`, et l'intégration dans les mécanismes du jeu. Voici les étapes détaillées :

7.1 Créer une nouvelle classe

1. Créer un fichier pour la classe :

- Naviguez dans le dossier Models et ajoutez un nouveau fichier C# pour votre classe.
Par exemple : Necromancer.cs.

2. Définir la classe :

- La nouvelle classe doit hériter de la classe abstraite Character.
- Implémentez le constructeur en initialisant tous les attributs requis (points de vie, puissance d'attaque, armure, etc.).
- Ajoutez les compétences spécifiques à cette classe dans la méthode Skills.Add.

3. Exemple :

```
using Services;
using Enums;

public class Necromancer : Character
{
    public Necromancer(string name, int maxHitPoints, int physicalAttackPower,
        int magicAttackPower, TypeOfArmor armor, int dodgeChance, int paradeChance,
        int chanceSpellResistance, int speed, bool usesMana, int maxMana)
        : base(name, maxHitPoints, physicalAttackPower, magicAttackPower, armor,
        dodgeChance, paradeChance, chanceSpellResistance, speed, usesMana,
        maxMana)
    {
        Skills.Add(new Skill(
            "Raise Dead",
            "Summons a skeleton to fight for your team.",
            3,
            TargetType.Self,
            40,
            ActionType.Buff,
            0
        ));

        Skills.Add(new Skill(
            "Dark Blast",
            "Deals 120% of magical attack power to an enemy.",
            2,
            TargetType.Enemy,
            30,
            ActionType.Damage,
            (int)(magicAttackPower * 1.2),
            TypeDamage.Magic
        ));
    }

    public override void ChoiceAction()
    {
        // Implémentez le comportement spécifique pour le choix des actions
    }
}
```

```

    }

    protected override DefenseResult Defend(Character attacker, TypeDamage
    typeOfAttack, int attackPower)
    {
        // Implémentez les mécanismes de défense spécifiques
        return base.Defend(attacker, typeOfAttack, attackPower);
    }
}

```

7.2 Ajouter les caractéristiques dans classes.json

1. Ouvrez le fichier classes.json :

- Ce fichier est situé dans le dossier Resources.

2. Ajouter une nouvelle entrée pour le personnage :

- Ajoutez une section contenant toutes les caractéristiques par défaut de votre personnage.

3. Exemple :

```

{
  "Necromancer": {
    "MaxHitPoints": 80,
    "PhysicalAttackPower": 20,
    "MagicAttackPower": 100,
    "Armor": "Fabric",
    "DodgeChance": 10,
    "ParadeChance": 5,
    "ChanceSpellResistance": 25,
    "Speed": 60,
    "HasMana": true,
    "ManaPoints": 120
  }
}

```

7.3 Ajouter le personnage dans Utils.CreatePlayer

1. Mettre à jour la méthode CreatePlayer :

- Ouvrez la méthode CreatePlayer dans le fichier Utils.
- Ajoutez un nouveau cas pour le personnage en associant son nom avec sa classe.

2. Exemple :


```
case 6:
    return new Necromancer(chosenName, config.MaxHitPoints,
        config.PhysicalAttackPower, config.MagicAttackPower,
        Enum.Parse<TypeOfArmor>(config.Armor!), config.DodgeChance,
        config.ParadeChance,
        config.ChanceSpellResistance, config.Speed, config.HasMana,
        config.ManaPoints);
```

7.4 Tester le nouveau personnage

1. Lancer le jeu :
 - Compilez et exécutez le jeu pour vérifier que le nouveau personnage est disponible.
 2. Créer une équipe avec le nouveau personnage :
 - Assurez-vous que ses compétences fonctionnent comme prévu et que ses statistiques sont correctement appliquées.
 3. Valider les calculs :
 - Testez ses mécanismes de défense, d'attaque et l'utilisation des compétences.
-