

Compte rendu d'activité

Introduction :	2
Mes objectifs personnels dont j'ai été responsable:	3
Création du jeu partie mécanique en java :	3
Création d'aléatoire en python :	3
Création de bots qui jouent au jeu :	4
Création d'un input text :	4
Partie graphique :	4
Méthodes de Travail Mises en Œuvre :	6
Point de vue technique :	6
Point de vue communication :	6
Conclusion :	6
Les difficultés rencontrées et les solutions mises en œuvre :	7
Problèmes identifiés :	7
Solutions mises en œuvre :	7
Les apprentissages acquis ou travaillés :	8
Compétences en Communication :	8
Travail d'Équipe avec les Méthodes Agiles (Scrum) :	8
Utilisation Avancée de GitLab :	8
Gestion de Projet :	8
Collaboration avec des Clients :	9
Adaptabilité et Flexibilité :	9
Glossaire des sigles :	10

Introduction :

Dans le cadre de notre projet scolaire SAE dans le parcours IAMS1 (Intégration des applications et management des systèmes d'informations), notre équipe composé de cinq étudiants en BUT2 s'est engagée dans le développement d'un jeu de casino (plus précisément le jeu de la roulette), avec pour objectif premier de répondre de manière optimale aux besoins spécifiques de nos client. Ce projet a été orchestré en suivant les principes des méthodes agiles, avec un Product Owner à la tête de la vision du produit, un Scrum Master garantissant la bonne marche des opérations, et une équipe dédiée à la réalisation de l'ensemble des fonctionnalités attendues.

La méthodologie agile que nous avons adoptée a été structurée autour de quatre sprints, chacun découpé en itérations courtes de 3 semaines et dynamiques. Des réunions quotidiennes ont rythmé notre progression, favorisant la communication au sein de l'équipe et permettant d'ajuster rapidement notre travail en fonction des retours obtenus. L'utilisation de gitlab ainsi que ces fonctionnalités comme par exemple les issues ont été très important. Ces principes agiles ont été appliqués de manière rigoureuse tout au long du processus de développement, contribuant ainsi à la flexibilité et à l'adaptabilité nécessaires pour répondre efficacement aux évolutions des besoins du client.

Mes objectifs personnels dont j'ai été responsable:

Création du jeu partie mécanique en java :

J'ai pris en charge la conception et l'implémentation de la partie mécanique du jeu en utilisant Java. Avec un collègue, nous avons établi les bases du jeu, développant les mécanismes clés et assurant une programmation orientée objet modulaire et robuste. Cette contribution a joué un rôle central dans la création d'un code évolutif, répondant aux besoins du client, et a renforcé la dynamique collaborative au sein de l'équipe.

```

@Override
public void placerMise(Couleur couleur, double montant) {
    ResourceBundle resourceBundle = LangueManager.getResourceBundle();
    double miseMinimale = getMiseMinimaleProperty();
    double jetonMaximum = couleur.getJetonMaximum();
    double miseActuelle = joueur.getMiseTotalePourCouleur(couleur);
    double miseTotaleApresNouvelleMise = miseActuelle + montant;
    double miseRestante = jetonMaximum - miseActuelle;

    if (miseTotaleApresNouvelleMise > jetonMaximum) {
        if (miseActuelle > 0) {
            throw new IllegalArgumentException(
                resourceBundle.getString("betImpossibleAlreadyBet") + " " +
                formaterNombre(miseActuelle) + " " +
                resourceBundle.getString("tokensOn") + " " +
                couleur.getNomTraduit(resourceBundle) + ". " +
                resourceBundle.getString("canOnlyBet") + " " +
                formaterNombre(miseRestante) + " " +
                resourceBundle.getString("additionalTokens")
            );
        } else {
            throw new IllegalArgumentException(
                resourceBundle.getString("betExceedsMaximum") + " " +
                formaterNombre(jetonMaximum) + " " +
                resourceBundle.getString("tokensForColor") + " " +
                couleur.getNomTraduit(resourceBundle) + "."
            );
        }
    }
    if (montant < miseMinimale) {
        throw new IllegalArgumentException(
            resourceBundle.getString("betBelowMinimum") + " " +
            formaterNombre(miseMinimale) + "."
        );
    }
    if (montant > joueur.getSolde().get()) {
        throw new IllegalArgumentException(resourceBundle.getString("insufficientBalanceForBet"));
    }
    joueur.miser(couleur, montant);
}

```

Création d'aléatoire en python :

J'ai conçu un générateur aléatoire en Python sans utiliser les classes ou fonctions standard comme `random`. En adoptant une approche manuelle, j'ai rassemblé de l'entropie à partir de diverses sources, via des APIs, puis appliqué des techniques de hachage personnalisées pour obtenir des données aléatoires.

Cette approche a été implémentée dans notre jeu en Java, où le script Python est exécuté pour fournir des données aléatoires. J'ai élaboré un document explicatif destiné au client, décrivant succinctement le code, la différence entre aléatoire et pseudo-aléatoire, ainsi que la complexité inhérente au problème.

Pour faciliter la maintenance future, j'ai inclus une documentation détaillée du code, destinée aux développeurs suivants. Cela couvre les choix de conception, les explications sur les

sources d'entropie, et les procédures de génération aléatoire. Cette initiative vise à assurer une compréhension claire du système pour les développeurs futurs tout en mettant en lumière la complexité de la génération aléatoire dans notre contexte de jeu de casino.

```
def collect_entropy():
    timestamp = int(time.time())
    url = "https://earthquake.usgs.gov/earthquakes/feed/v1.0/summary/all_day.geojson"
    response = requests.get(url)

    if response.status_code == 200:
        data = response.content

        earthquake_data = json.loads(data)

        extracted_data = {}
        if 'features' in earthquake_data and len(earthquake_data['features']) > 0:
            first_earthquake = earthquake_data['features'][0]
            if 'properties' in first_earthquake:
                properties_data = first_earthquake['properties']
                extracted_data['mag'] = properties_data.get('mag', 'N/A')
                extracted_data['time'] = properties_data.get('time', 'N/A')
                extracted_data['updated'] = properties_data.get('updated', 'N/A')
            if 'geometry' in first_earthquake:
                geometry_data = first_earthquake['geometry']
                if 'coordinates' in geometry_data:
                    extracted_data['coordinates'] = geometry_data['coordinates']

    return extracted_data
```

```
def custom_mix(data, key):
    data_to_mix = ''.join([str(byte) for byte in data])
    data_bytes = data_to_mix.encode()

    key_len = len(key)
    mixed_data = bytearray(len(data_bytes))

    for i in range(len(data_bytes)):
        char = data_bytes[i]
        key_char = key[i % key_len]
        key_value = int(key_char)
        mixed_byte = char ^ int(key_char)
        mixed_data[i] = mixed_byte

    return mixed_data
```

Création de bots qui jouent au jeu :

J'ai développé des bots avec des pseudos variés, capables d'envoyer des messages lorsqu'ils rejoignent, gagnent ou perdent. Ces bots participent activement au jeu, misant de manière aléatoire pour ajouter une dimension imprévisible et immersive. Ils jouent

```
public void afficherMiseBot() {
    String resultat = getPseudo() + " " + resourceBundle.getString("parier");
    int n = 0;
    for (Couleur c : getMise().keySet()) {
        double valeur = getMise().get(c);
        int nombreEntier = (int) valeur;
        resultat += " " + nombreEntier + " ";
        String couleurTraduite = c.getNomTraduit(resourceBundle);
        resultat += resourceBundle.getString("jetons") + " " + couleurTraduite;
        if (n != getMise().size() - 1) {
            resultat += ", ";
        }
        n++;
    }
    ChatService.ajouterMessage(resultat, false);
    ChatService.ajouterMessage("", false);
}

public void mettreAJourSolde(Couleur c){
    if (getMise().containsKey(c)){
        double somme = getMise().get(c)*c.getFacteurGain();
        setSolde(getSolde().get() + somme);
        possibilitéMessageWin(somme);
    }else{
        possibilitéMessageLoose();
    }
}
```

réellement au jeu, contribuant ainsi à la diversité des parties. Un classement affiche les performances des bots et du joueur pour une expérience interactive et compétitive.

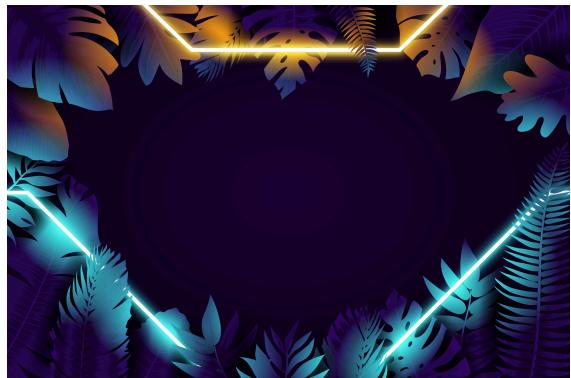
Création d'un input text :

J'ai intégré un champ de saisie (input text) permettant à l'utilisateur du jeu d'envoyer des messages dans le chat. Cette fonctionnalité offre une expérience interactive permettant à l'utilisateur de communiquer avec les bots. Les messages de l'utilisateur sont affichés en temps réel, facilitant la communication avec les bots et permettant à l'utilisateur de suivre les réponses. Cette interface de chat améliore l'aspect social du jeu en offrant une interaction dynamique avec l'environnement virtuel des bots.

Partie graphique :

J'ai personnalisé l'interface graphique en assemblant un fond d'écran avec des éléments PNG, répondant ainsi aux demandes spécifiques du client. Cette personnalisation vise à offrir une expérience visuelle unique et adaptée aux préférences du public cible.

Avant :

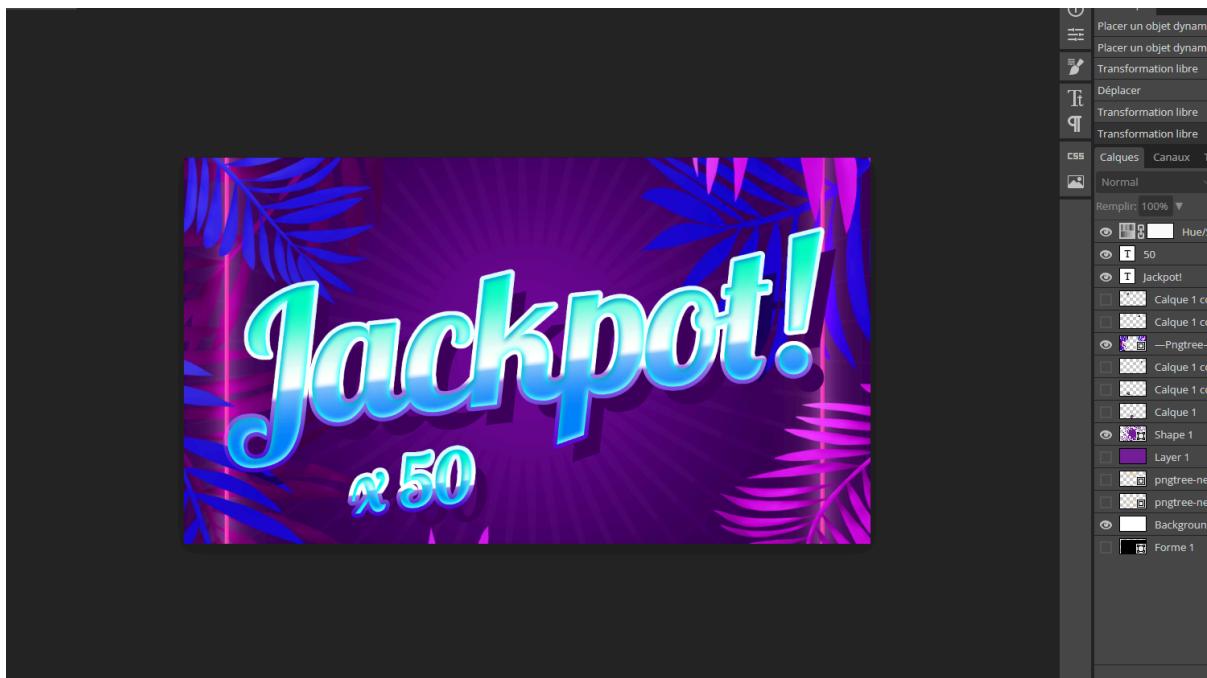


Après :

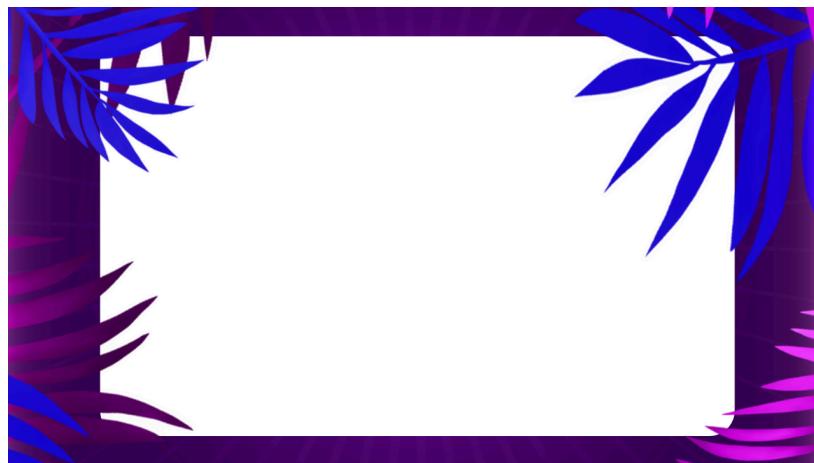


En outre, j'ai créé des visuels distincts pour les différentes victoires, ajoutant ainsi une dimension visuelle dynamique au jeu. Ces visuels sont conçus avec une direction artistique réutilisable, offrant une cohérence esthétique qui peut être étendue à d'autres éléments du jeu, comme les règles. Cette approche contribue à renforcer l'identité visuelle du jeu et à garantir une expérience utilisateur homogène.

- Visuel d'un jackpot :



- Réutilisation de la direction artistique pour faire les règles (en cours de réalisation) :



Méthodes de Travail Mises en Œuvre :

Point de vue technique :

Nous avons choisi d'appliquer la méthodologie Scrum, intégrant des daily meetings pour assurer un suivi quotidien du progrès et une communication fluide au sein de l'équipe.

En parallèle, pour la gestion de projet, nous avons utilisé GitLab, mettant en place des issues spécifiques assignées en fonction des compétences individuelles et attribuant des poids aux tâches pour planifier efficacement chaque sprint.

Point de vue communication :

En termes de communication, nous avons également utilisé Discord comme canal de discussion en temps réel. Ce choix s'est avéré particulièrement utile pour résoudre rapidement les problèmes rencontrés par les membres de l'équipe. Lorsqu'un membre faisait face à des difficultés, Discord nous permettait de réagir immédiatement, favorisant ainsi une collaboration étroite et une résolution rapide des obstacles.

Conclusion :

En somme, cette combinaison de méthodes techniques, de gestion de projet Scrum, et de communication via GitLab et Discord a créé un environnement de travail dynamique et collaboratif, assurant une progression régulière du projet tout en favorisant une communication transparente et efficace au sein de l'équipe.

Les difficultés rencontrées et les solutions mises en œuvre :

Problèmes identifiés :

Répétition de tâches : Au début du projet, des difficultés ont été rencontrées avec des personnes recodant par-dessus d'autres, ce qui entraînait parfois la duplication de fonctions.

Communication : Un problème de communication initiale a été observé, impactant la coordination et la cohérence du travail d'équipe.

Travail régulier : Un autre défi identifié était la difficulté à maintenir une cadence de travail régulière. Pour aborder ce problème, une approche proactive a été adoptée en utilisant les graphiques de Git pour visualiser la progression du travail au fil du temps.

Solutions mises en œuvre :

Répétition de tâches : Pour résoudre ce problème, des actions ont été prises, notamment lors du deuxième sprint. Des daily meetings ont été introduits, ce qui a amélioré la communication et permis à l'équipe de s'aligner plus efficacement. Cela a également contribué à relancer la productivité générale.

Communication : Les daily meetings ont été particulièrement utiles pour améliorer la communication. Les discussions quotidiennes ont permis de résoudre les malentendus rapidement et de garantir que chaque membre de l'équipe était informé des progrès et des défis.

Utilisation de graphiques Git : Les graphiques Git ont été intégrés dans la gestion du projet pour suivre la progression de manière visuelle. Cela a permis à l'équipe de mieux comprendre les tendances de productivité et d'identifier les périodes de pic ou de baisse d'activité. En ayant une vision claire de l'évolution du travail, il est devenu plus facile de planifier et de maintenir une cadence de travail plus constante.

Les apprentissages acquis ou travaillés :

Compétences en Communication :

- Compréhension de l'importance de la communication dans la réussite d'un projet d'équipe.
- Capacité à transmettre efficacement des informations à différents membres de l'équipe.

- Utilisation de la communication pour résoudre des problèmes et prendre des décisions collectives.

Travail d'Équipe avec les Méthodes Agiles (Scrum) :

- Maîtrise des concepts agiles, en particulier de la méthodologie Scrum.
- Expérience dans des rôles spécifiques tels que Product Owner, Scrum Master, et autres.
- Connaissance de l'importance des sprints, des réunions de planification, des revues de sprint, etc.

Utilisation Avancée de GitLab :

- Compétence approfondie dans l'utilisation de GitLab pour la gestion de versions.
- Capacité à collaborer efficacement avec d'autres membres de l'équipe à travers des branches, des merges, etc.
- Expérience dans le suivi des changements, la résolution des conflits, et la gestion des pipelines CI/CD.

Gestion de Projet :

- Compréhension des aspects clés de la gestion de projet, y compris la planification et l'organisation.
- Application des principes agiles pour s'adapter aux changements et améliorer continuellement le processus de travail.
- Réalisation de livrables à chaque fin de sprint, démontrant la progression du projet.

Collaboration avec des Clients :

- Expérience dans l'interaction avec des clients pour comprendre leurs besoins et attentes.
- Capacité à intégrer les commentaires des clients dans le processus de développement.
- Aptitude à ajuster les priorités en fonction des exigences client.

Adaptabilité et Flexibilité :

- Développement d'une capacité à s'adapter rapidement aux changements de priorités.
- Compréhension de l'importance de la flexibilité dans un environnement de travail dynamique.

Glossaire des sigles :

IAMSI : Intégration des Applications et Management des Systèmes d'Informations

API : Interface de Programmation Applicative

Git : Système de Contrôle de Version

Discord : Plateforme de Communication en Temps Réel

CI/CD : Intégration Continue / Livraison Continue

PNG : Format d'Image Portable (Portable Network Graphics)

Scrum : Méthode de Gestion de Projet Agile

Product Owner : Propriétaire du Produit

Scrum Master : Maître Scrum

GitLab : Plateforme de Gestion de Projet basée sur Git

Entropie : Mesure du Désordre ou de l'Incertitude dans un Système