

# Hel : The pixelated horror

## Travail de diplôme 2015

Yannick R. Brodard

Centre de Formation Professionnelle Technique  
École d'informatique

Mercredi, 10 juin 2015

# Sommaire

Introduction

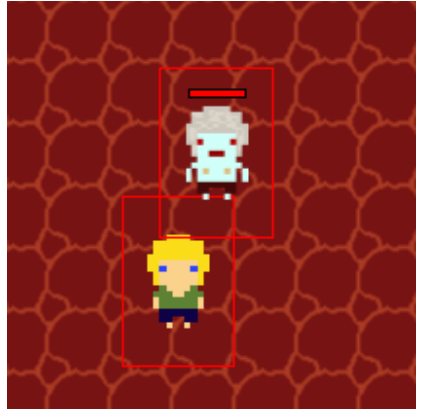
Fonctionnalités

Méthodes

Démonstration

Conclusion

- ▶ Jeu-vidéo 2D
- ▶ Aspects RPG
  - ▶ Système de compétences
  - ▶ Système d'objets
- ▶ Monogame



# Pourquoi ce sujet ?

- ▶ Les jeux ressemblants sont :
  - ▶ Trop faciles
  - ▶ ou trop complexes
- ▶ Trouver le juste milieu
- ▶ Défi à soi-même



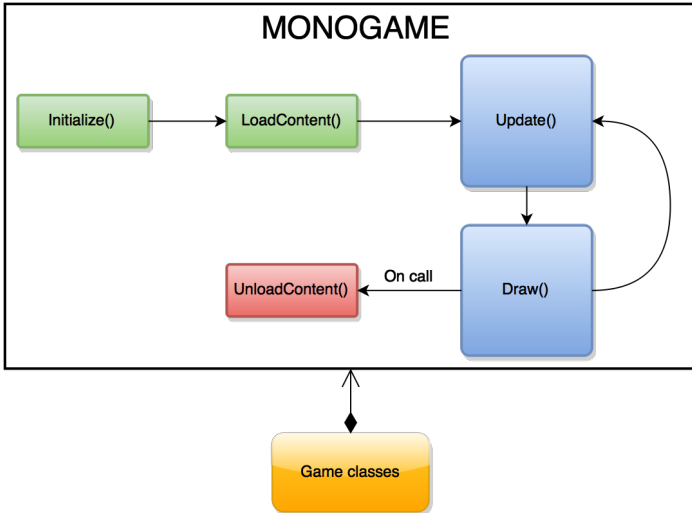
# Hel et son histoire

- ▶ Basé sur la mythologie nordique
- ▶ Hel, déesse de la mort



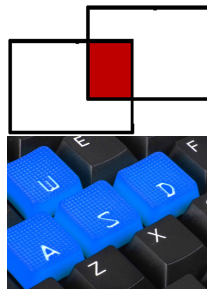
- ▶ Boucle de jeu
- ▶ Gestion des textures
- ▶ Gestion de l'affichage
- ▶ Utilisation du GPU





# Outils de développement

- ▶ Primitives 2D
- ▶ Collisions/intersections
- ▶ Gestion des entrées
- ▶ Sérialisation XML





```
1  <Map>
2    <Dimensions>
3      <Width>10</Width>
4      <Height>10</Height>
5    </Dimensions>
6    <Cells>
7      <Cell>
8        <X>0</X>
9        <Y>0</Y>
10       <IsWalkable>False</IsWalkable>
11       <Type>wall</Type>
12     </Cell>
13     <Cell>
14       <X>1</X>
15       <Y>0</Y>
16       <IsWalkable>False</IsWalkable>
17       <Type>wall</Type>
18     </Cell>
```

Figure: Fichier XML du village

Le jeu possède :

- ▶ Des règles de jeu
- ▶ Une carte générée aléatoirement
- ▶ Un personnage contrôlable
- ▶ Des ennemis indépendants

# Règles du jeu

- ▶ Vaincre des ennemis
- ▶ Gagner des points de compétences
- ▶ Trouver des objets
- ▶ Vaincre le boss "Hel"



Force



Vitalité



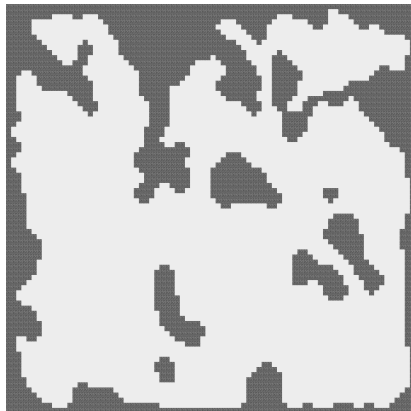
Agilité



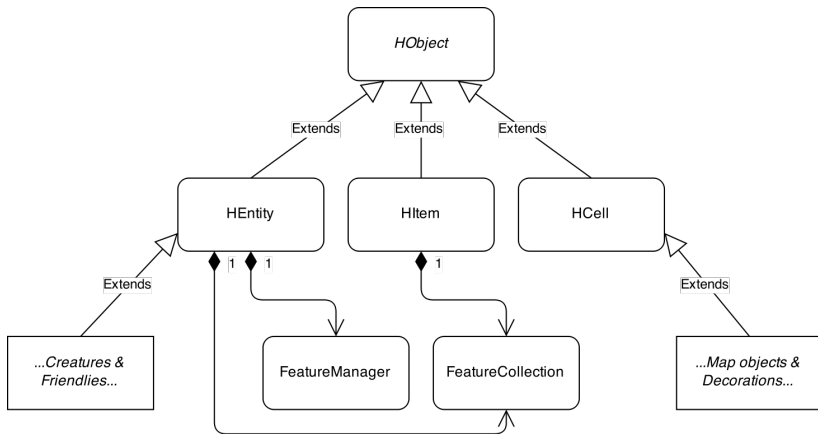
Magie

# Carte

- ▶ Génération aléatoire
- ▶ Utilise un algorithme d'automate cellulaire
  - ▶ Réagi selon les voisins



# Les éléments du jeu



&lt;&lt;abstract&gt;&gt;

***HEntity***

+ InitialFeatures: FeatureCollection  
+ ActualFeatures: FeatureCollection  
+ MaximizedFeatures: FeatureCollection  
+ FeatureCalculator: FeatureManager  
+ State: EntityState  
+ Direction: Vector2  
+ Bounds: FRectangle  
+ AttackBounds: FRectangle  
+ Texture: Texture2D  
+ LastAttackTime: double  
+ IsDead: boolean

+ <ctor> ()  
+ <ctor> (FPosition)  
+ <ctor> (FeatureCollection, FPosition)  
+ Update(GameTime) : void  
+ Draw(SpriteBatch) : void  
+ ApplyFluidMovement(Vector2, Vector2, FRectangle, float) : void  
+ IsCharacterSurfaceWalkable(Vector2, FRectangle) : void  
+ BasicMeleeAttack(HEntity, GameTime) : void  
+ CheckLife() : void

**PRESS START  
TO BEGIN**

## ► Conclusion

- Terminer le système d'objets et de points de compétences.
- Ajout de contenu est simple
- Nécessite beaucoup de temps de travail

## ► Perspectives

- Une version stable d'ici quelques mois
- Peut être suivi sur GitHub : [www.github.com/brodardy](http://www.github.com/brodardy)



# Questions ?

