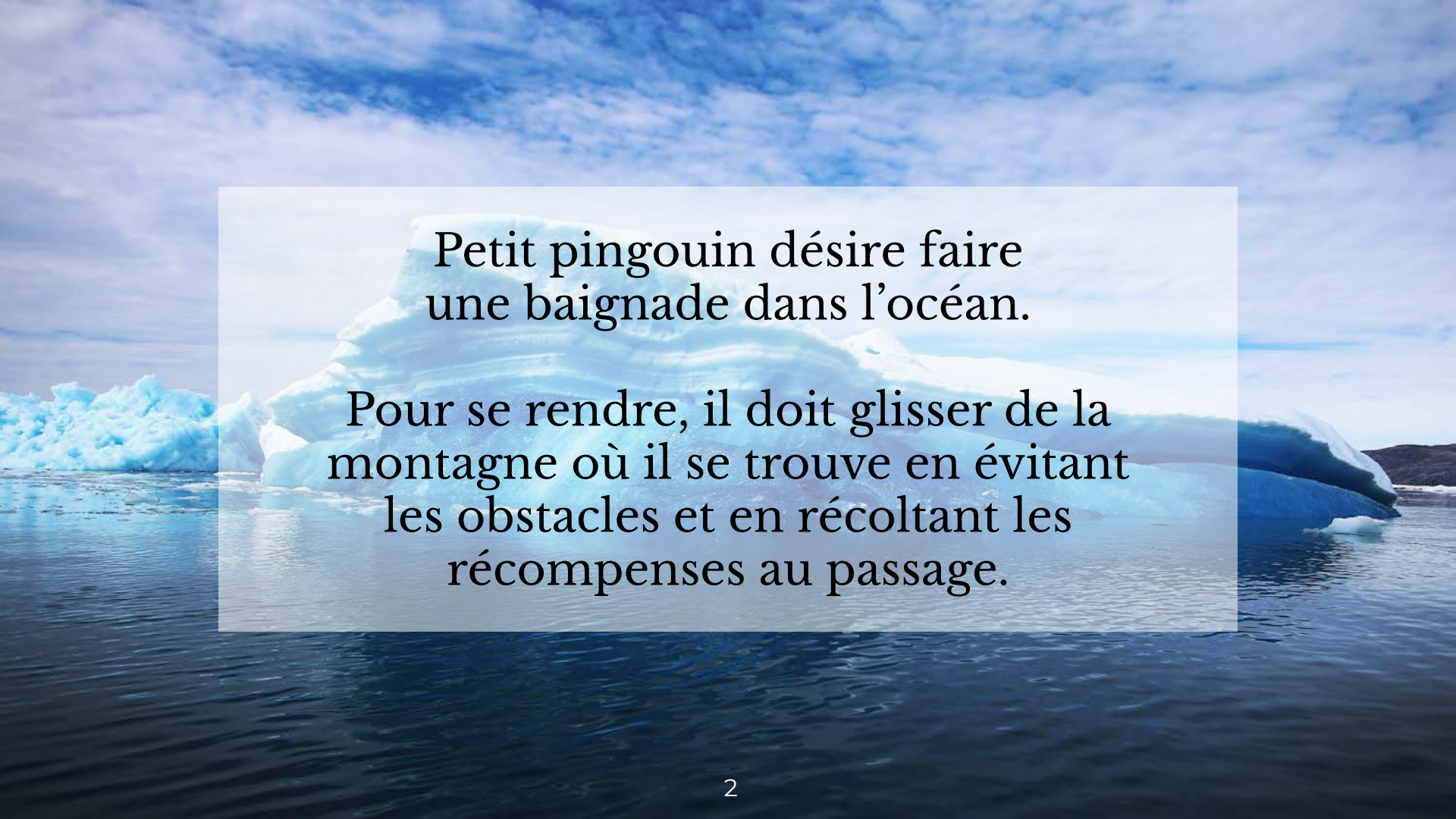
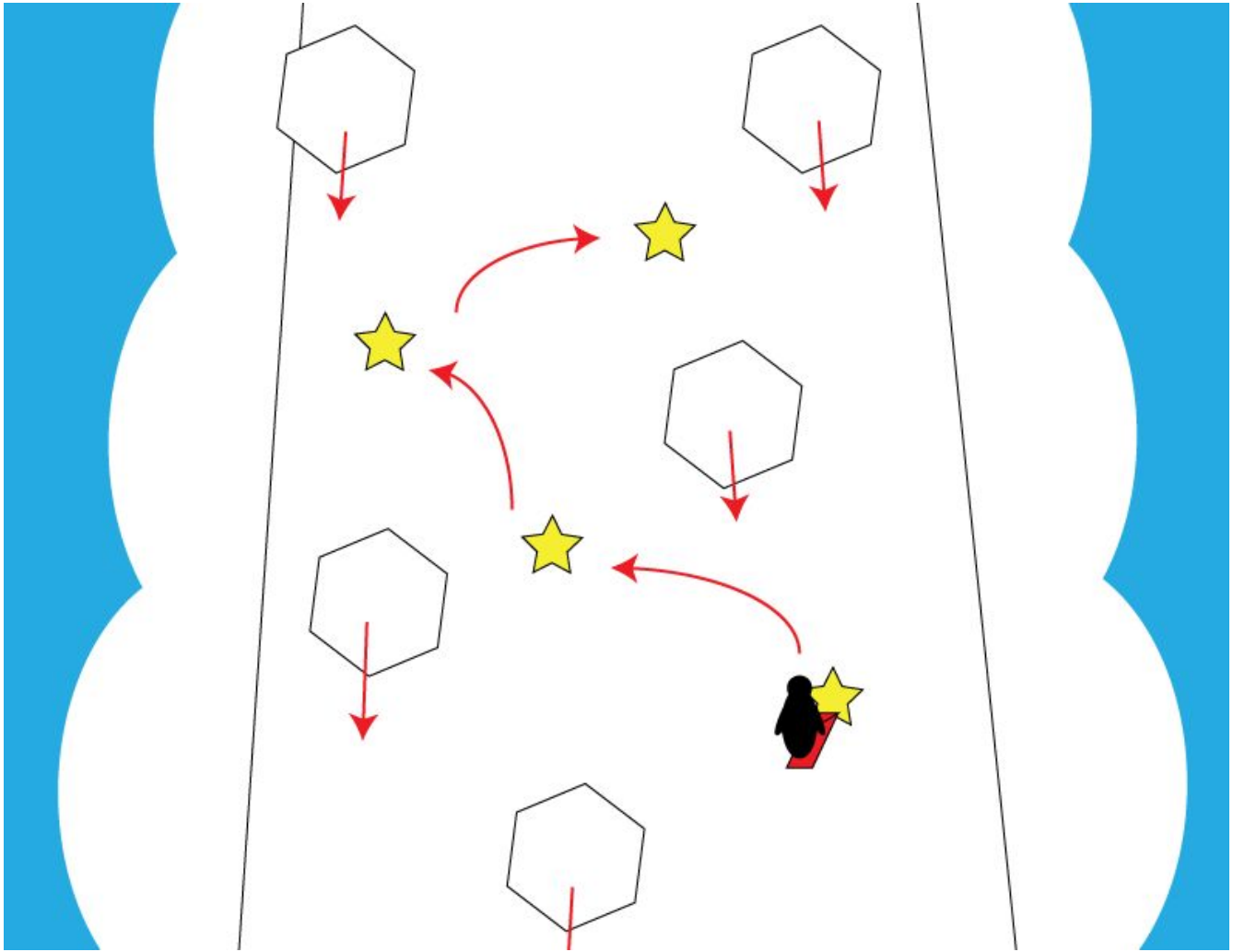
Présentation “Sliding Penguin”



Petit pingouin désire faire
une baignade dans l'océan.

Pour se rendre, il doit glisser de la
montagne où il se trouve en évitant
les obstacles et en récoltant les
récompenses au passage.



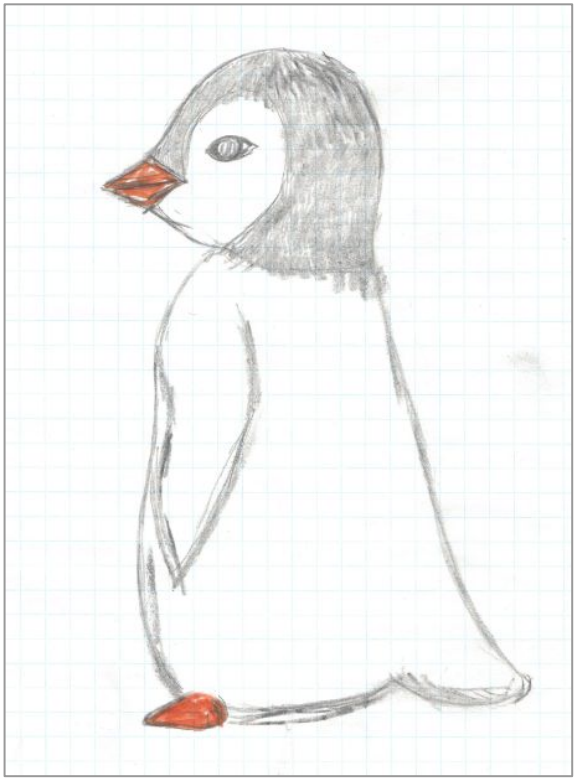
CONTENU DE LA PRÉSENTATION

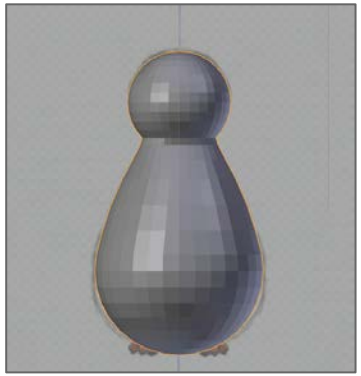
- Modèles
- Scène
- Canvas
- Obstacles
- Générateurs de particules
- Accélération

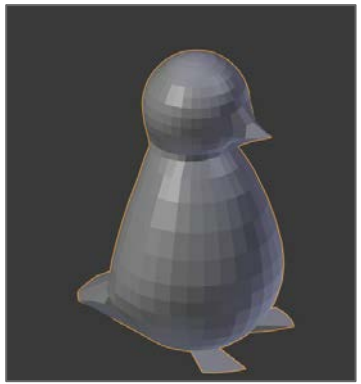
Modèles

Outil : Blender et SolidWorks

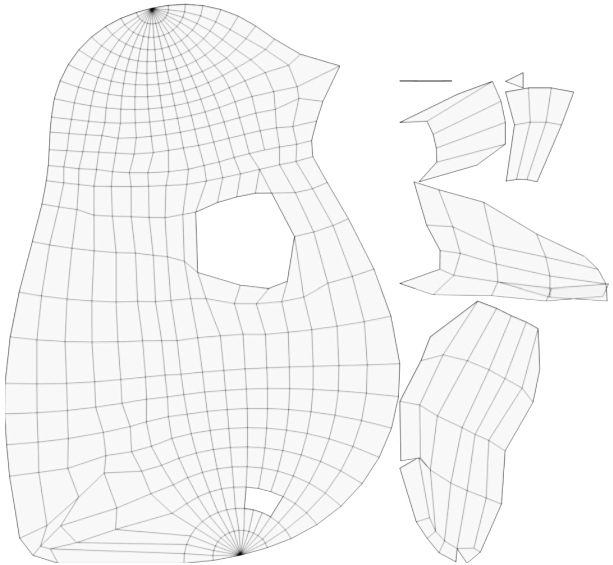
CRÉATION DU PERSONNAGE | Structure



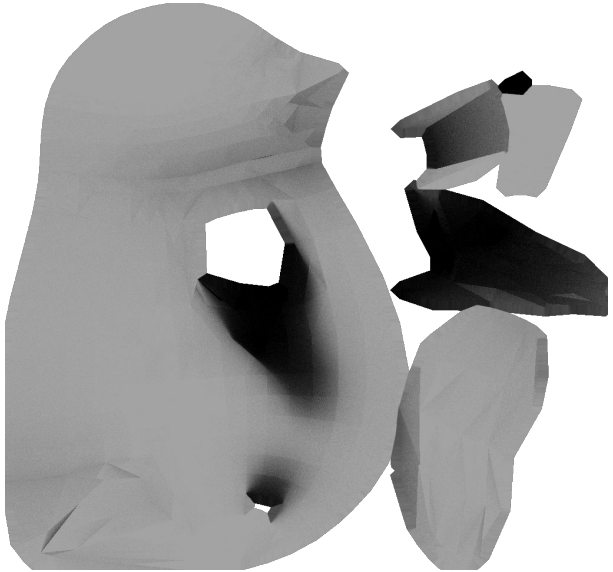




CRÉATION DU PERSONNAGE | Texture



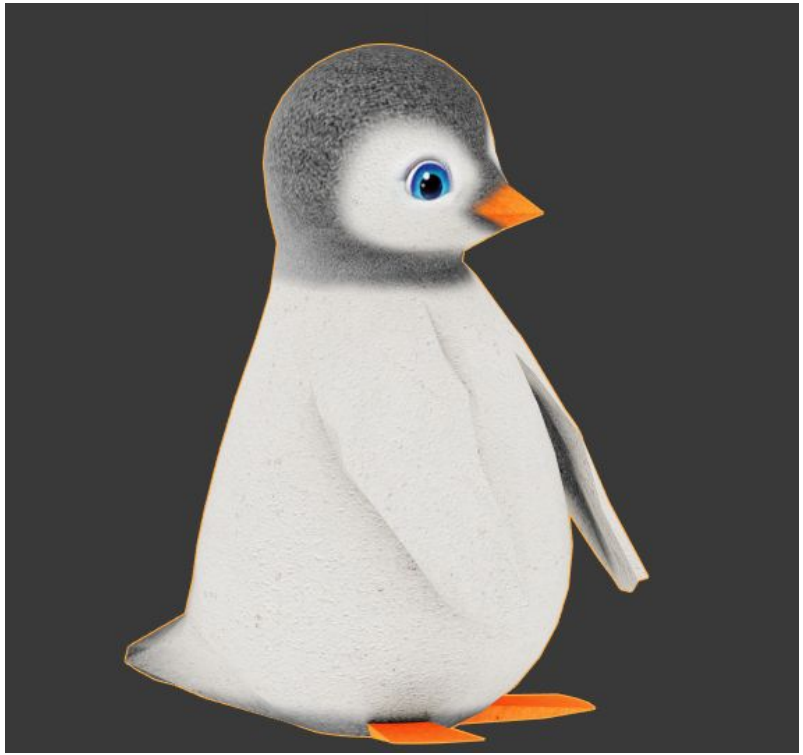
Découpage



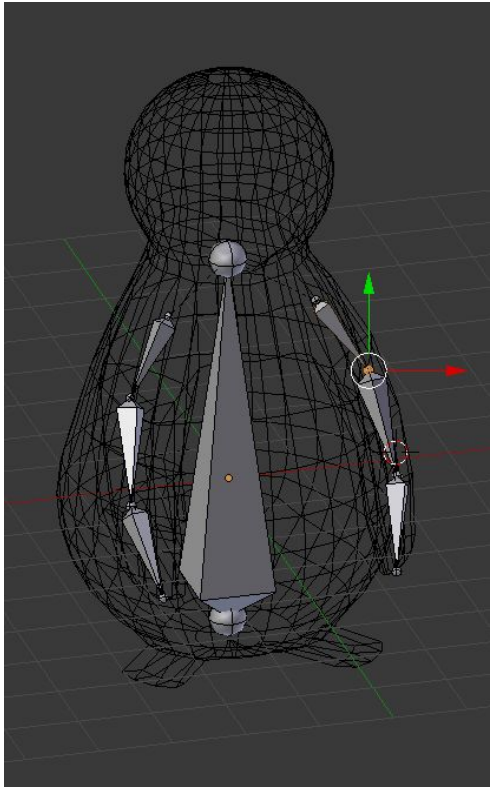
Ombrage

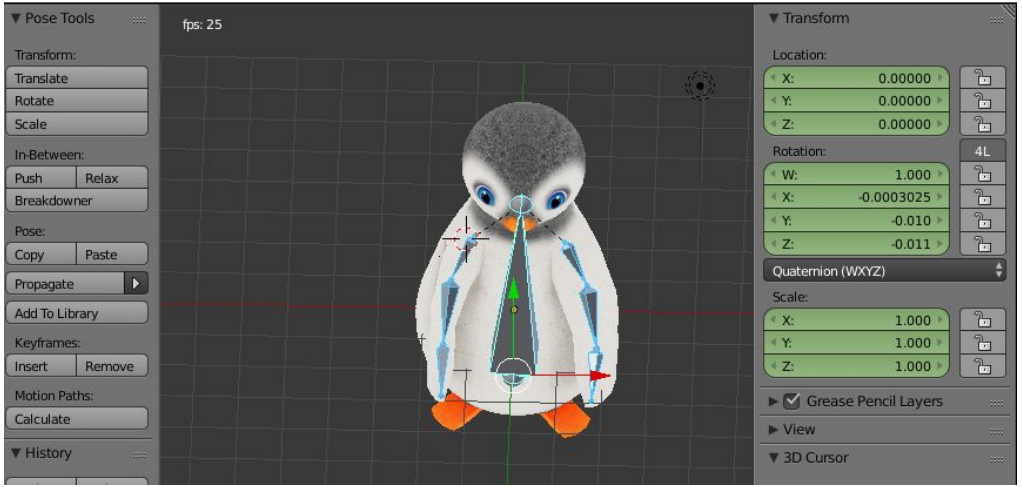
Texture finale

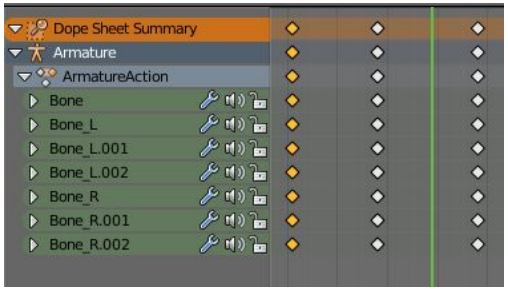
CRÉATION DU PERSONNAGE | Résultat

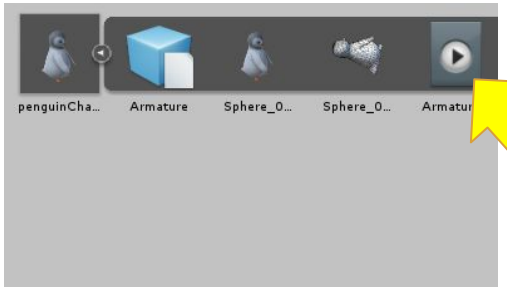


CRÉATION DU PERSONNAGE | Animation

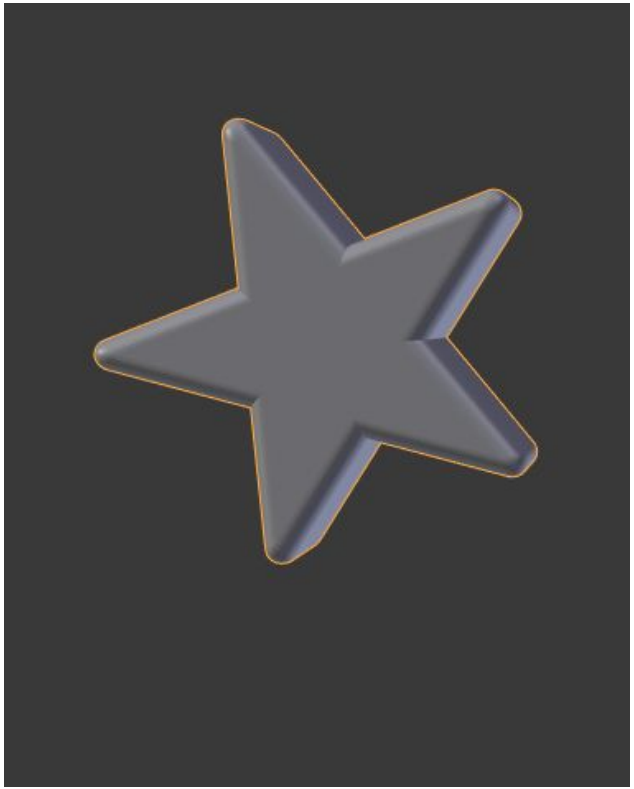


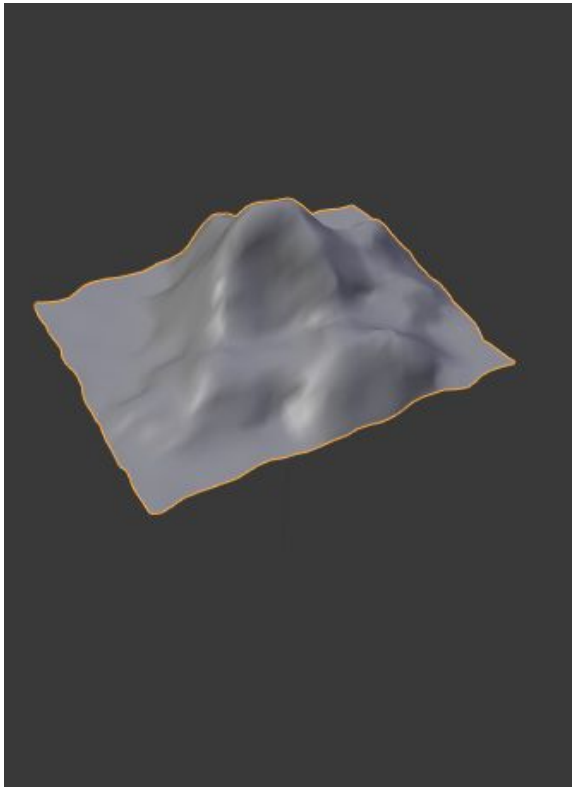






AUTRES MODÈLES

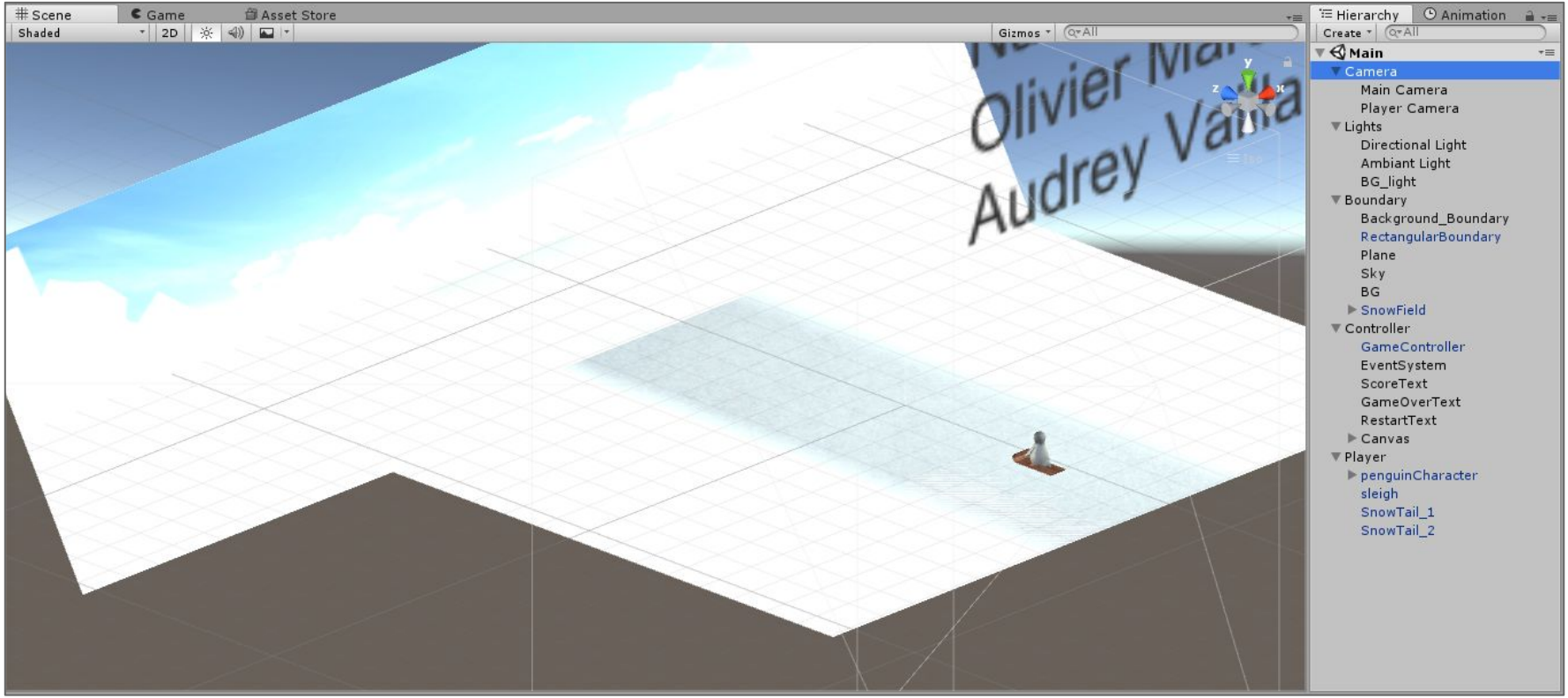




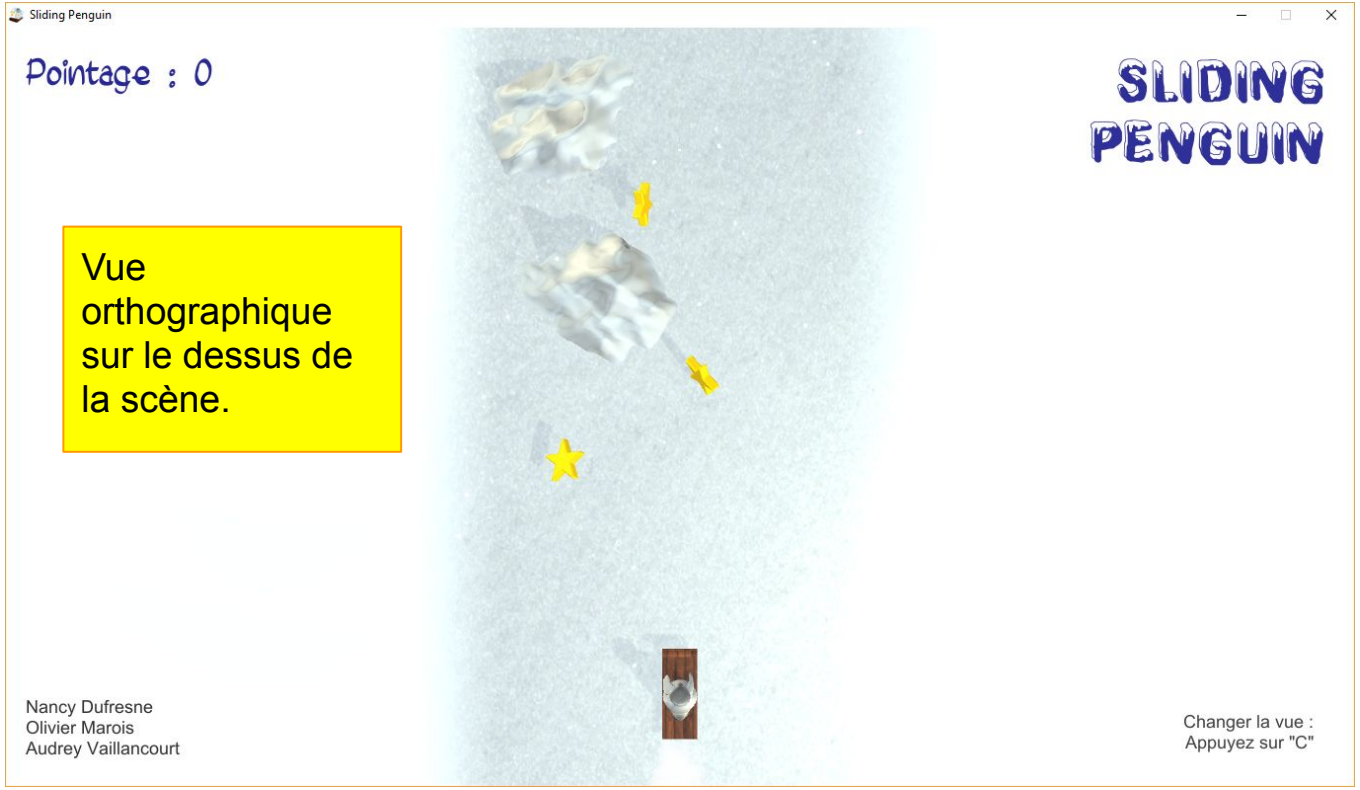
Scène

Outil : Unity 3D et Visual Studio (C#)

SCÈNE | Présentation



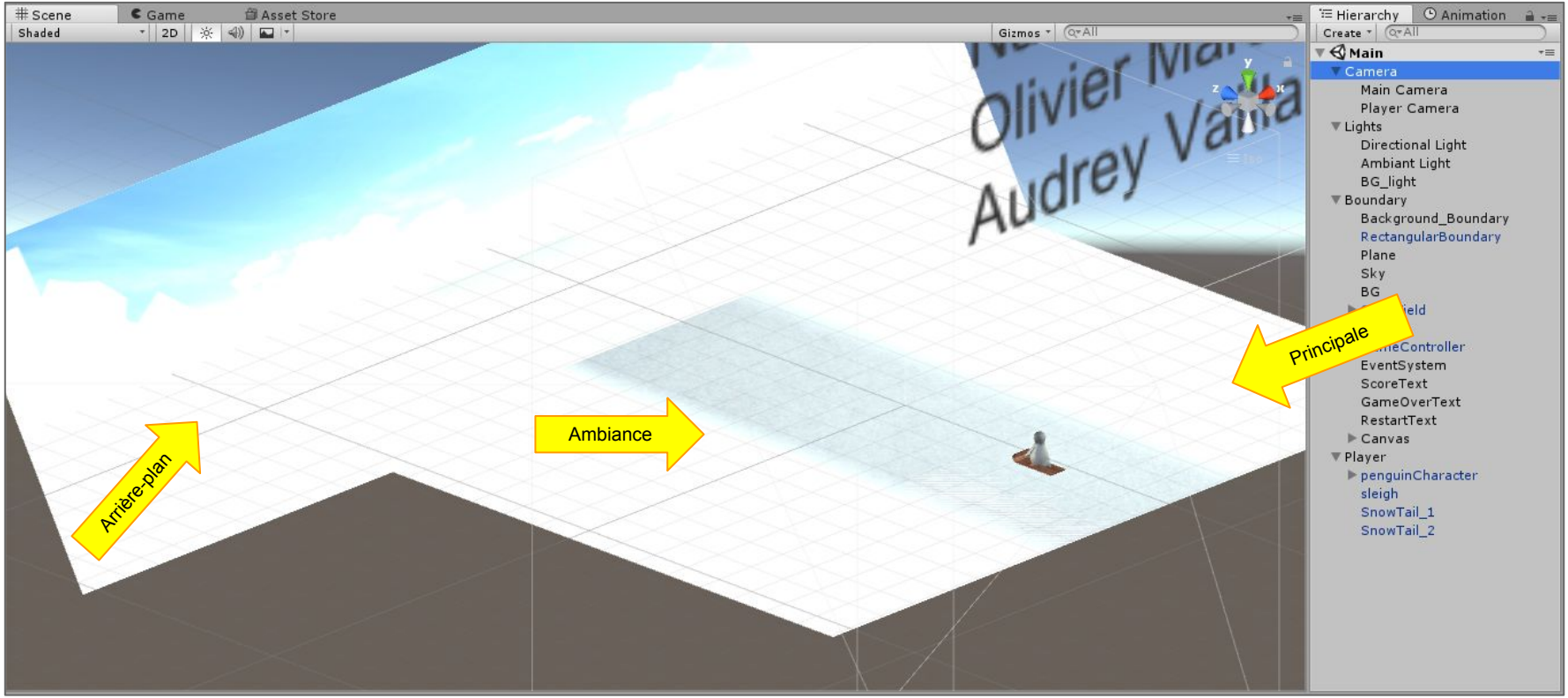
SCÈNE | Caméra principale



SCÈNE | Caméra du personnage

Vue en perspective qui suit le personnage.

SCÈNE | Éclairage

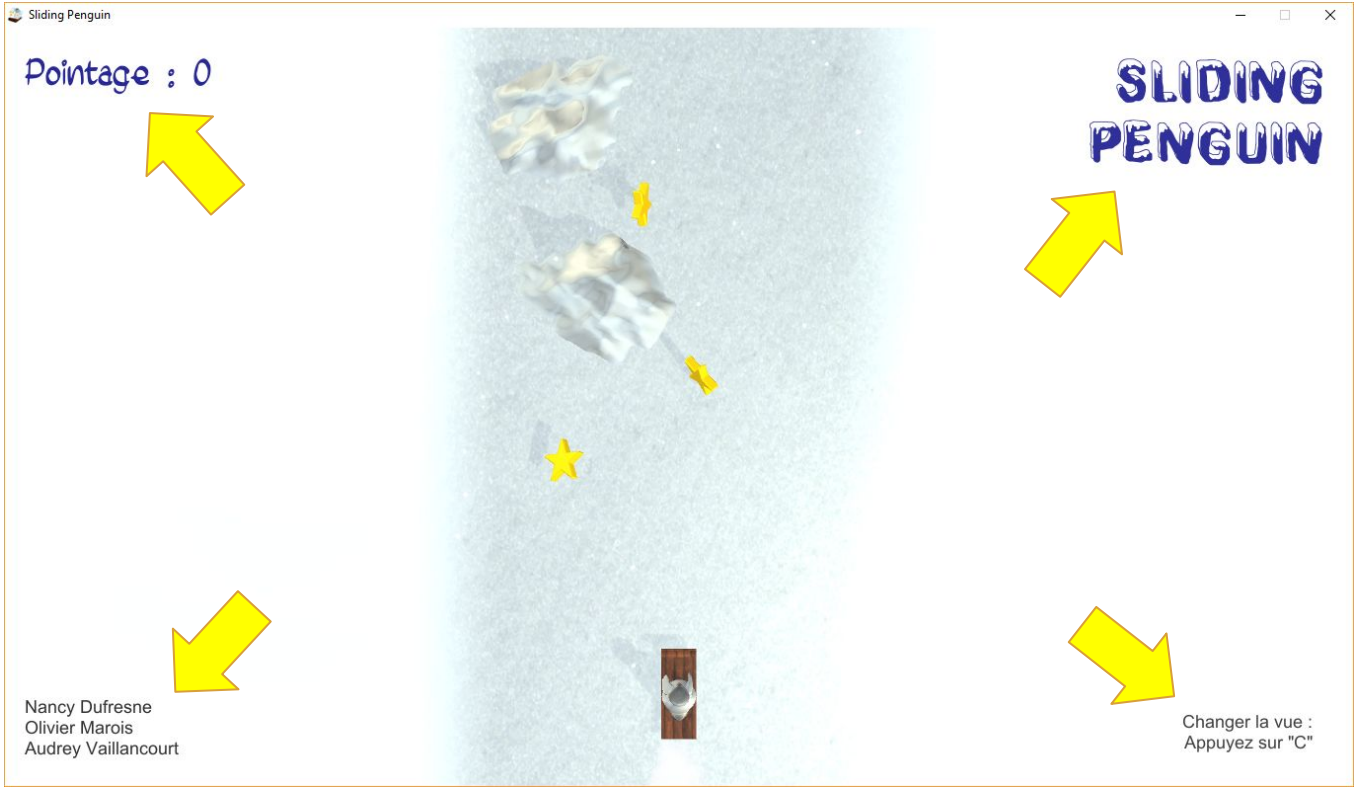


Canvas

Outil : Unity 3D

CANVAS | Affichage

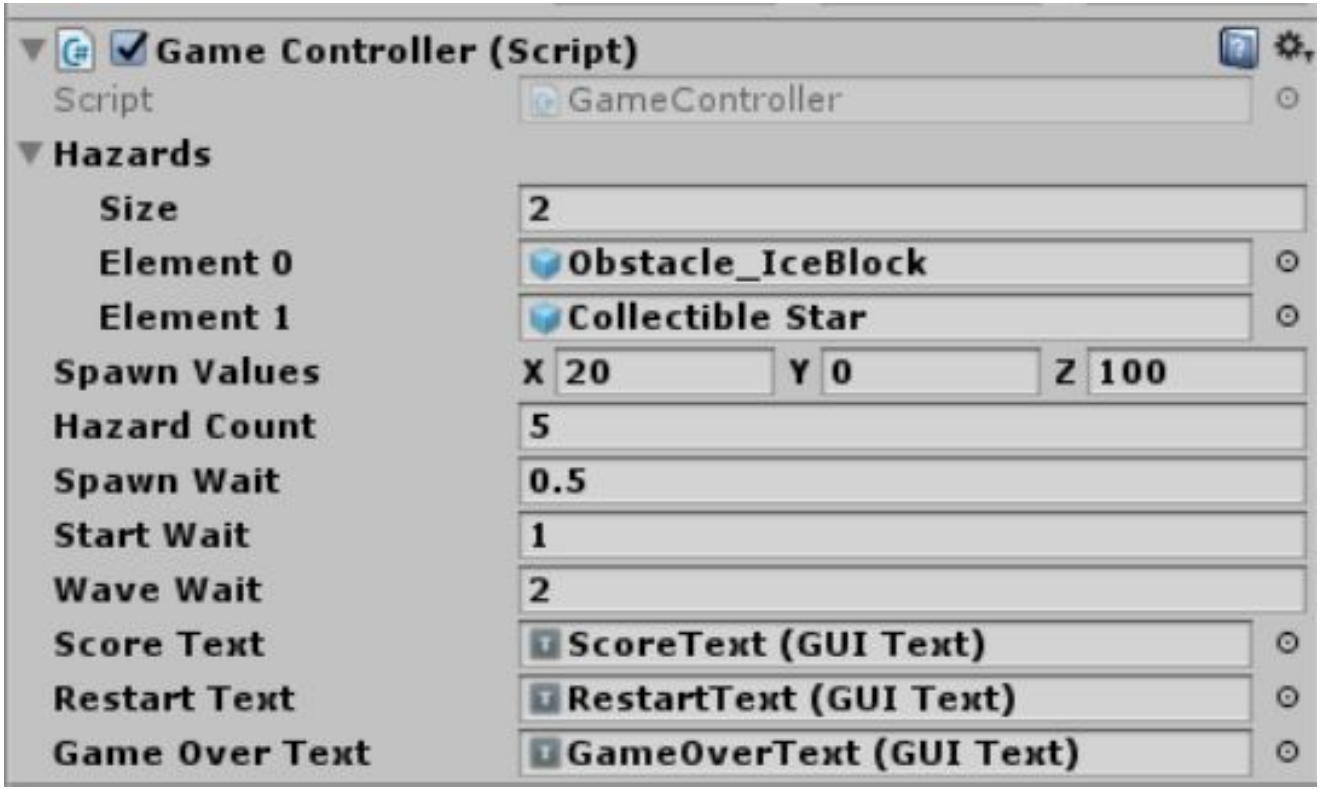
Texte accroché au 4 coins et polices spéciales disponibles dans les Assets.



Obstacles

Outil : Unity 3D et Visual Studio (C#)

SCÈNE | Obstacles



SCÈNE | Obstacles

1. ***Au démarrage, débuter l'apparition d'obstacles :***

```
void Start ()
{

    // Sélection de la caméra
    CameraTop.enabled = true;
    CameraPlayer.enabled = false;

    gameOver = false;
    restart = false;
    restartText.text = "";
    gameOverText.text = "";
    score = 0;
    UpdateScore ();
    StartCoroutine(SpawnWaves ());

    initiateObjectMover();

}
```

SCÈNE | Obstacles

2. Obtenir un objet à faire apparaître au hasard

```
GameObject hazard = hazards[UnityEngine.Random.Range(0, hazards.Length)];
```

3. Déterminer la vitesse de l'obstacle, d'après une accélération grandissante

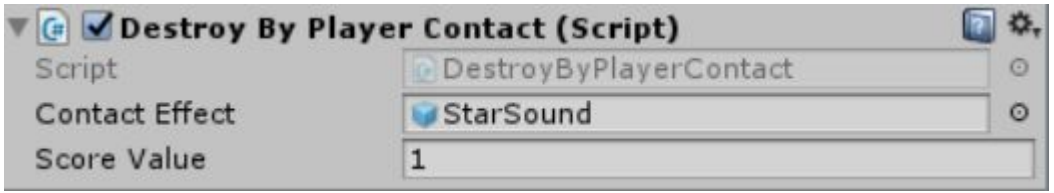
```
instancedHazard.GetComponent<Rigidbody>().velocity = transform.forward * -speed;
```

4. Faire apparaître l'obstacle avec une rotation aléatoire

```
Transform child = instancedHazard.transform.GetChild(0);
child.transform.Rotate(new Vector3(0.0f, 0.0f, UnityEngine.Random.Range(0, 359)));
```

SCÈNE | Obstacles - Effets sonores

Ajout de l'effet au contact directement à l'obstacle



Dans la méthode, "onTriggerEnter":

```
Destroy(gameObject);
gameController.AddScore(scoreValue);
if (contactEffect != null)
    Instantiate(contactEffect, transform.position, transform.rotation);
```

Générateurs de particules

Outil : Unity 3D

GÉNÉRATEURS DE PARTICULES | Paramètres

Duration	5.00
Looping	☐
Prewarm	☐
Start Delay	0
Start Lifetime	3
Start Speed	5.5
3D Start Size	☐
Start Size	15
3D Start Rotation	☐
Start Rotation	0
Randomize Rotation	0
Start Color	
Gravity Modifier	0
Simulation Space	Local
Simulation Speed	1
Scaling Mode	Local
Play On Awake*	☑
Max Particles	1000
Auto Random Seed	☑

✓ Emission

Rate over Time	0
Rate over Distance	0

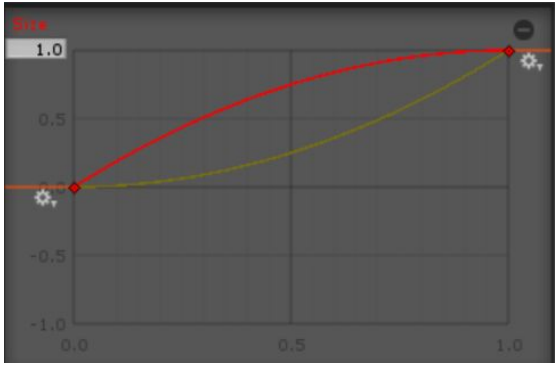
Bursts	Time	Min	Max
	0.10	30	30
	0.50	50	50

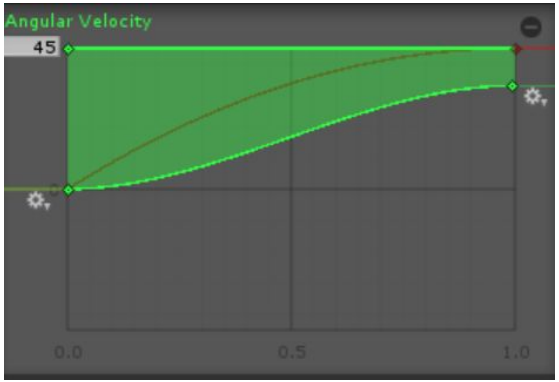
✓ Shape

Shape	Sphere
Radius	1
Emit from Shell	☐
Align To Direction	☐
Randomize Direction	0
Spherize Direction	0

✓ Color over Lifetime

Color	

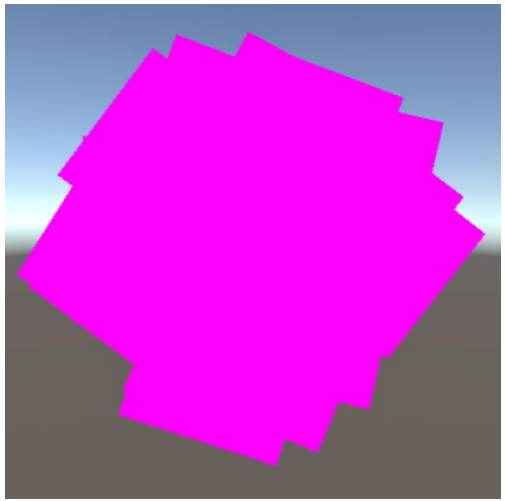




GÉNÉRATEURS DE PARTICULES | Texture

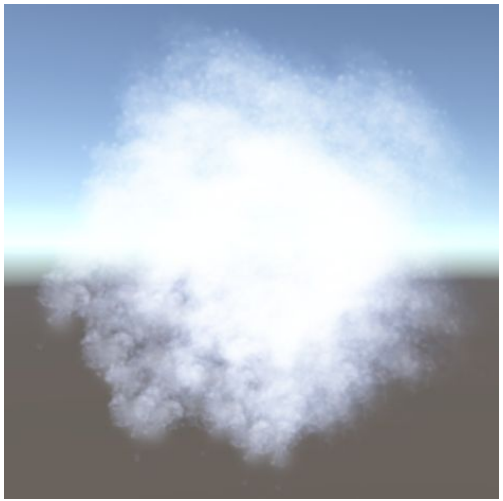
Texture

+



Système

=



Résultat

GÉNÉRATEURS DE PARTICULES | Résultats

Accélération

Outil : Unity 3D et Visual Studio (C#)

ACCÉLÉRATION | Calcul

$$m\frac{d\vec{v}(\vec{r},t)}{dt} = \vec{F}_f(\vec{v},\vec{r},t) + \vec{F}_g(\vec{v},\vec{r},t)$$

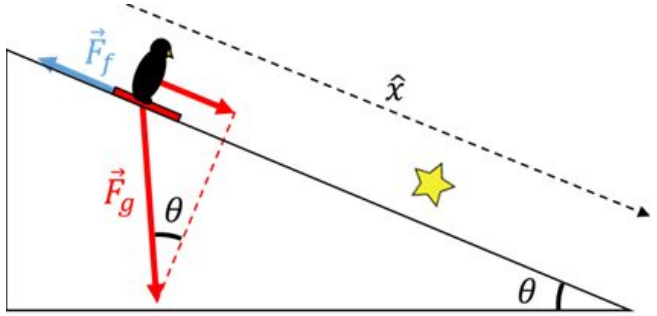
$$\Rightarrow m\frac{dv_x(\vec{r},t)}{dt} = -kv_x(\vec{r},t) + mg\sin\theta$$

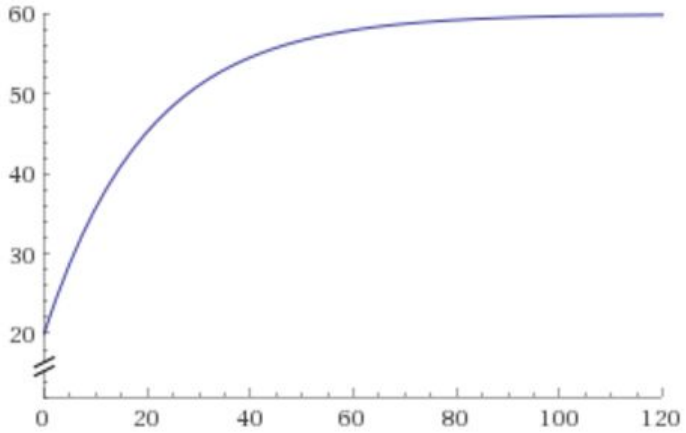
$$\Leftrightarrow \frac{dv_x(\vec{r},t)}{-\frac{k}{m}v_x(\vec{r},t) + g\sin\theta} = dt$$

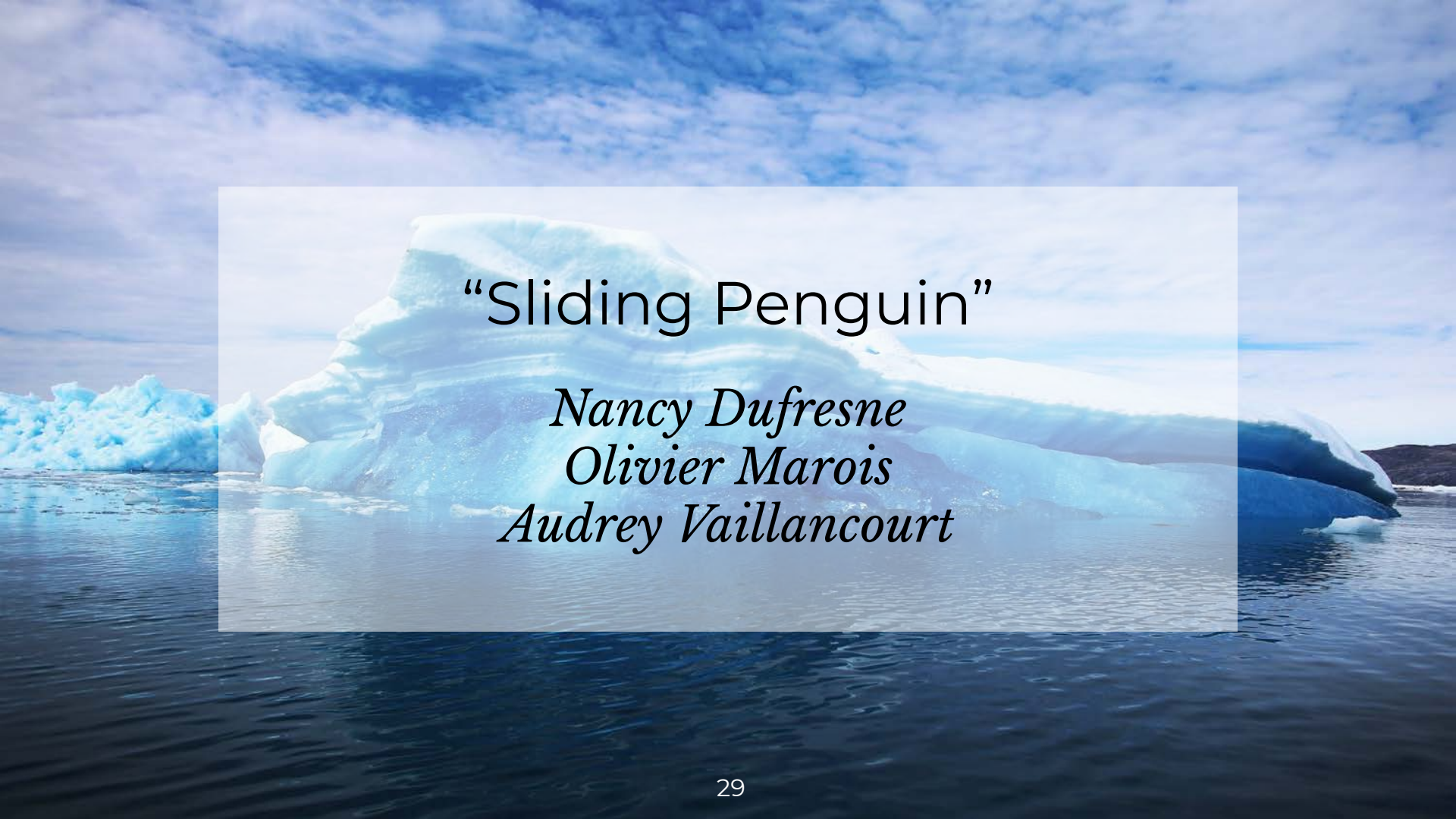
$$\Leftrightarrow -\frac{m}{k}\ln\left(-\frac{k}{m}v_x(\vec{r},t) + g\sin\theta\right) = t + C$$

$$\Leftrightarrow v_x(\vec{r},t) = \frac{m}{k}g\sin\theta - Ce^{-\frac{k}{m}t}$$

$$\Rightarrow v_x = v_f + (v_0 - v_f)e^{-\frac{k}{m}t}$$






“Sliding Penguin”
Nancy Dufresne
Olivier Marois
Audrey Vaillancourt