

7. Sprint

 Hase Script 👁️ 💬 2 SH
 Hase Features 👁️ 💬 5 <input checked="" type="checkbox"/> 4/4 SH
 Igel Script 💬 4 YC
 Schwein Script 👁️ 💬 6 SH T
 Mögliche Drop bei Allen Gegnern 👁️ 💬 2 SH
 Abgrund Script 💬 1 YC
 Fallen Raum design & Script 💬 11 <input checked="" type="checkbox"/> 8/11 T YC

Ziele aus dem Trello-Board für den Sprint 7

Erläuterung der Ziele:

Hase Script (Part 2)

Die dazugehörige Logik wird implementiert.

Diese Aufgabe wurde aus dem letzten Sprint übernommen

Hase Features

Der Hase unterscheidet sich grundlegend von den anderen Gegnern, daher wurden hier im nachhinein spezielle Features implementiert.

Igel Script (Part 2)

Die dazugehörige Logik wird implementiert.

Diese Aufgabe wurde aus dem letzten Sprint übernommen

Schwein Script (Part 2)

Die dazugehörige Logik wird implementiert.

Diese Aufgabe wurde aus dem letzten Sprint übernommen

Mögliche Drop bei Allen Gegnern

Alle unsere Gegner haben jetzt eine geringe Chance, eine Waffe zu dropfen.

Abgrund Script

Das Abgrundskript sorgt dafür, dass, wenn der Spieler in einen Abgrund fällt, zurück zum Spawnpunkt kommt und er verliert ein Leben.

Fallen Raum design & Script (Part 3)

Die erstellten und fertigen Waffen sollen in einem Raum sinnvoll angeordnet werden, gegebenenfalls ausgebessert werden und der erste Fallenraum wird verbessert und weiter ausgebaut.

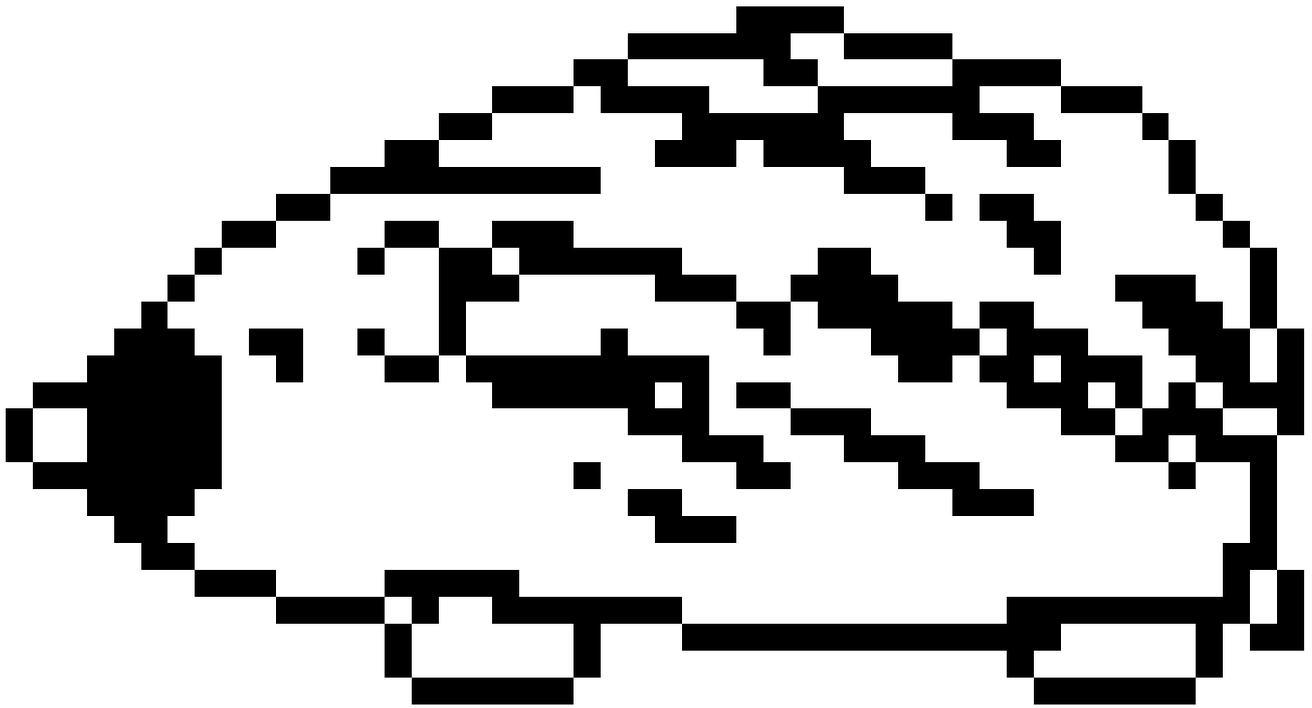
Diese Aufgabe wurde aus dem letzten Sprint übernommen

"Best Of" the Sprint:

Der Fokus in diesen Sprint war die Verbesserung des bisher erstellten Contents. Dazu wurde jetzt der Fallenraum fertig gestellt.



Zudem haben wir die neuen Gegner aus dem letzten Sprint von Fehlern befreit, sodass jetzt das Schwein, der Hase und der Igel funktionieren, wie sie sollen.



Die Besonderheit des Hasen ist, dass er eine 100% Chance hat, eine Waffe zu dropen,
Das Problem dabei ist, dass er selten stehenbleibt, sehr schnell ist und sehr viel aushält.
Zur Krönung des Ganzen verschwindet er, sobald er lang genug im Raum verweilt hat.

Ein Beispiel eines Codeteils aus unserem Hasenskript, der die Dropchance und das Verschwinden nach einiger Zeit regelt.

```

// Update is called once per frame
void Update()
{
    deathTimer += Time.deltaTime;

    if (deathTimer >= death)
    {
        Destroy(gameObject);
    }

    enemyHealthBar.fillAmount = (float)health / (float)maxHealth;

    if (health <= 0)
    {
        Instantiate(deathExplosion, new Vector3(transform.position.x, transform.position.y - 0.5f, transform.position.z), Quaternion.identity);

        if (randomnumber <= 0.05)
        {
            Instantiate(droplist[0], new Vector3(transform.position.x, transform.position.y - 0.5f, transform.position.z), Quaternion.identity);
        }
        if (randomnumber > 0.05 && randomnumber <= 0.1)
        {
            Instantiate(droplist[1], new Vector3(transform.position.x, transform.position.y - 0.5f, transform.position.z), Quaternion.identity);
        }
        if (randomnumber > 0.1 && randomnumber <= 0.15)
        {
            Instantiate(droplist[2], new Vector3(transform.position.x, transform.position.y - 0.5f, transform.position.z), Quaternion.identity);
        }
        if (randomnumber > 0.15 && randomnumber <= 0.2)
        {
            Instantiate(droplist[3], new Vector3(transform.position.x, transform.position.y - 0.5f, transform.position.z), Quaternion.identity);
        }
        if (randomnumber > 0.2 && randomnumber <= 0.25)
        {
            Instantiate(droplist[4], new Vector3(transform.position.x, transform.position.y - 0.5f, transform.position.z), Quaternion.identity);
        }
        if (randomnumber > 0.25 && randomnumber <= 0.3)
        {
            Instantiate(droplist[5], new Vector3(transform.position.x, transform.position.y - 0.5f, transform.position.z), Quaternion.identity);
        }
        if (randomnumber > 0.3 && randomnumber <= 0.35)
        {
            Instantiate(droplist[6], new Vector3(transform.position.x, transform.position.y - 0.5f, transform.position.z), Quaternion.identity);
        }
        if (randomnumber > 0.35 && randomnumber <= 0.4)
        {
            Instantiate(droplist[7], new Vector3(transform.position.x, transform.position.y - 0.5f, transform.position.z), Quaternion.identity);
        }

        Destroy(gameObject);
    }
}

```