



Game Design Dokument

SPACE INVADERS

MOHAMAD ALASKARI



INHALTSVERZEICHNIS

INSTALLATION UND SETUP	2
Voraussetzungen	2
Projekt Setup	2
Idee/Philosophie	3
FAQ.....	3
Worum geht es in dem Spiel?.....	3
Warum dieses spiel?.....	3
Wo spielt das Spiel? (Spiele-Welt).....	4
wie wird das Spiel gesteuert?	4
Was ist der Fokus?	4
was ist neu/anders?.....	4
Spielfiguren / Charaktere	5
Kriegsschiff Spieler.....	5
feindliche Raumschiffe – Gegner.....	6
Monster – Gegner.....	6
Bullet.....	6
Upgrade Bubbles.....	7
User Interface	7
System-Steuerung.....	7
Spieler-Steuerung	8
Spiele Logik-Elemente	9
Gegner AI-Monster	9
Gegner AI- Feindlichen Raumschiffe.....	10
Waffen	12
DIE FEINDLICHEN RAUMSCHIFFE	12
Spieler-Waffen	13
Grafik Elemente.....	16
Hintergründe.....	16
Sprites	17
Sound / Musik.....	19
Hintergrundmusik.....	19
Sound-Effekte	19
Level.....	20
LEVEL 1 - DIE INVASION BEGINNT:.....	20

LEVEL 2 - BULLET-UPGRADE VERFÜGBAR:	20
LEVEL 3 - STÄRKERE FEUERKRAFT:	20
LEVEL 4 - SHOWDOWN MIT DEM MONSTER:	20

INSTALLATION UND SETUP

VORAUSSETZUNGEN

Um das Spiel auf einem lokalen Computer zu starten oder weiterzuentwickeln, muss eine Umgebung eingerichtet werden, die Node.js und npm unterstützt. Darüber hinaus wird Vite.js für die Entwicklung des Spiels benötigt.

- Node.js und npm: Sie können Node.js und npm von der offiziellen Website herunterladen:
[HTTPS://NODEJS.ORG](https://nodejs.org)

Vite.js: Sie können Vite.js mit dem folgenden Befehl installieren:

```
npm install -g create-vite
```

PROJEKT SETUP

Auf GitHub finden Sie den vollständigen Quellcode und die Dateien des Spiels, sodass Sie es selbst ausprobieren oder sogar daran mitarbeiten können, falls Sie Interesse an der Weiterentwicklung haben.

Klicken Sie auf den Link, um direkt zu "Space Invaders v3" auf GitHub zu gelangen und die Weltraum-Action zu erleben. Viel Spaß beim Spielen und Entdecken! Hier ist der Link:

<https://github.com/MohamadAlaskari/spaceinvadersv3.git>

Sobald Node.js und Vite.js auf Ihrem Computer installiert sind, folgen Sie den unten stehenden Anweisungen, um "Space Invaders" einzurichten:

1. Klonen Sie das "Space Invaders" Projekt aus dem Repository auf Ihren lokalen Computer.
2. Navigieren Sie in das Projektverzeichnis.
3. Installieren Sie alle Projekt-Abhängigkeiten mit dem Befehl `npm install`.
4. Nach erfolgreicher Installation der Abhängigkeiten können Sie das Spiel mit dem Befehl `npm start` starten.

Das Spiel sollte jetzt in Ihrem Standard-Webbrowser laufen und auf localhost (meistens auf Port 5000 oder 3000) verfügbar sein.



IDEE/PHILOSOPHIE

"Space Invaders" ist ein fesselndes Spiel, das uns auf eine Reise in die Vergangenheit mit seinem nostalgischen Arcade-Stil mitnimmt. Es kombiniert Spaß und Herausforderung, indem die Spieler die Erde vor Angriffen außerirdischer Kreaturen verteidigen müssen.

Das Spiel entführt die Spieler in ein aufregendes Weltraumabenteuer, in dem sie die letzte Hoffnung der Menschheit sind, um der außerirdischen Invasion entgegenzutreten. Du wirst deine präzisen Zielfähigkeiten und schnelles Denken brauchen, um die außerirdischen Kreaturen zu zerstören und die Welt zu retten.

Zusätzlich dazu kannst du dein Raumschiff aufrüsten und mit mächtigen Waffen ausstatten, um den schwierigen Herausforderungen zu begegnen.

"Space Invaders" basiert auf schneller Bewegung und sofortiger Reaktion und bietet ein aufregendes Spielerlebnis für Spieler jeden Alters. Du wirst Spannung und Nervenkitzel spüren, während du dich den immer stärker werdenden Wellen außerirdischer Kreaturen stellst und versuchst, im gefährlichen Weltraum zu überleben.

FAQ

WORUM GEHT ES IN DEM SPIEL?

Es geht um ein actiongeladenes Weltraumspiel, bei dem der Spieler ein Kriegsschiff steuert, um gegen verschiedene feindliche Raumschiffe zu kämpfen. Das Hauptziel des Spiels ist es, alle Levels zu bestehen und sich am Ende dem mächtigen Monster zu stellen.

Der Spieler sammelt Punkte, indem er die feindlichen Schiffe zerstört. Mit jeder zerstörten Einheit erhöht sich die Punktzahl des Spielers. Sobald eine bestimmte Punktzahl erreicht ist, steigt der Spieler in den nächsten Level auf. Jeder Level wird anspruchsvoller, mit schnelleren und aggressiveren Feinden.

Um die Herausforderungen zu bewältigen, hat der Spieler die Möglichkeit, das Kriegsschiff mit Upgrades auszustatten. Durch das Sammeln spezieller Upgrade-Bubbles kann das Schiff neue Waffen erhalten, schneller manövrieren. Diese Upgrades sind entscheidend, um den immer stärker werdenden Feinden entgegenzutreten und das Überleben des Spielers zu gewährleisten.

Das ultimative Ziel von "Space Invaders" ist es, alle Levels zu bestehen, indem man die verschiedenen feindlichen Raumschiffe besiegt und schließlich dem Monster entgegentritt. Das Monster ist ein besonders herausfordernder Endgegner, der über eine beeindruckende Angriffskraft und Bewegung verfügt. Der Spieler muss all seine Fähigkeiten und Upgrades nutzen, um das Monster zu besiegen und das Spiel erfolgreich abzuschließen.

WARUM DIESES SPIEL?

Das Spiel wurde ausgewählt, um die Freude an klassischen Arcade-Spielen wieder aufleben zu lassen und gleichzeitig neue Spielerlebnisse zu bieten. Es ist ein Tribut an die Nostalgie vergangener Zeiten, als Arcade-Spiele eine große Rolle im Gaming-Bereich spielten. Viele Spieler haben nostalgische Erinnerungen an das Original "Space Invaders", das mit einfachen, pixeligen Figuren und begrenzten Funktionen auf den frühen Spielkonsolen und Automaten gespielt wurde.

Das Ziel war es, diese klassische Erfahrung aufzugreifen und in eine zeitgemäße, hochwertige und herausfordernde Spielerfahrung zu verwandeln. Indem das Spiel mit modernen Technologien und hochwertigen Grafiken aufgewertet wurde, konnte eine fesselnde Atmosphäre geschaffen werden, die sowohl diejenigen, die das Originalspiel kannten, als auch neue Generationen von Spielern anspricht.

Für mich selbst war es auch eine Möglichkeit, ein Stück meiner eigenen Gaming-Vergangenheit wiederzuerwecken und zu verbessern. Das Spiel "Space Invaders" hatte einen besonderen Platz in meinem Herzen, und ich wollte diese Faszination für klassische Spiele mit anderen teilen. Durch die Neugestaltung des Spiels konnte ich meine Vision einer modernen Version von "Space Invaders" umsetzen, die die Essenz des Originals bewahrt, aber auch neue Elemente hinzufügt, um die Spieler zu begeistern und herauszufordern.

WO SPIELT DAS SPIEL? (SPIELE-WELT)

"Space Invaders" spielt in den unendlichen Weiten des Weltraums. Die Umgebung ist geprägt von endlosen Sternen, schwebenden Asteroiden und Meteoren, die die Spielwelt zum Leben erwecken. Die atemberaubende Kulisse zieht im Hintergrund vorbei und verleiht dem Spieler das Gefühl, tief in ein fesselndes Weltraumabenteuer einzutauchen.

Während des Spiels bewegt sich das Kriegsschiff des Spielers geschmeidig durch den Weltraum, während Sterne funkeln und Asteroiden vorbeiziehen. Die beeindruckende Spielewelt schafft eine einzigartige Atmosphäre und lässt den Spieler das Gefühl haben, Teil eines epischen Weltraumkampfes zu sein.

WIE WIRD DAS SPIEL GESTEUERT?

Die Spieler haben zwei Möglichkeiten, das Kriegsschiff zu steuern:

- 1. Tastatursteuerung**
- 2. Touchscreen-Steuerung**

Die feindlichen Raumschiffe und das Monster bewegen sich eigenständig und greifen den Spieler an.

WAS IST DER FOKUS?

Das Spiel legt großen Wert auf schnelle Reaktionen und sorgfältige Strategie. Spieler müssen die Muster der Feinde beobachten und ihre Aktionen entsprechend planen, um zu überleben und die maximale Punktzahl zu erreichen.

WAS IST NEU/ANDERS?

In dem Spiel gibt es mehrere neue und innovative Elemente, die das Spiel von der ursprünglichen Version abheben:

1. Upgrade-System: Das Punkte-Upgrade-System ermöglicht es den Spielern, durch das Sammeln von Upgrade-Bubbles ihr Kriegsschiff zu verbessern. Diese Upgrades bieten neue Fähigkeiten und stärkere Waffen, die es den Spielern ermöglichen, sich den Herausforderungen effektiver zu stellen.

2. Bewegung der Gegner: Die feindlichen Raumschiffe bewegen sich auf eine neue Art und Weise. Sie bewegen sich vertikal von oben nach unten und wechseln ihre x-Position zufällig. Diese Bewegungsmuster erfordern schnelle Reaktionen und strategisches Denken, um den Angriffen auszuweichen und die Feinde zu besiegen.

3. Vielfalt der Gegner: Das Spiel bietet eine Vielzahl von feindlichen Raumschiffen, die sich in ihrem Verhalten und ihren Angriffsmustern unterscheiden. Jeder Gegnertyp erfordert eine andere Herangehensweise und Strategie, um besiegt zu werden. Dies erhöht die Herausforderung und Vielfalt im Spiel.

4. Endgegner-Monster: Als Höhepunkt des Spiels tritt ein mächtiges Endgegner-Monster auf. Das Monster hat beeindruckende Angriffsfähigkeiten und erfordert besondere Strategien, um besiegt zu werden. Der Kampf gegen das Monster ist ein herausforderndes und spannendes Spielerlebnis.

SPIELFIGUREN / CHARAKTERE

Alle Grafiken für die Spielfiguren, einschließlich des Kriegsschiffs, der feindlichen Raumschiffe und des Monsters, wurden von Freepik bezogen.

Ich habe die Bilder von Freepik genommen und sie in Illustrator manipuliert, um sie an mein Spiel anzupassen.

Diese Grafiken von Freepik tragen zur visuellen Darstellung der Spielfiguren in "Space Invaders" bei und verleihen dem Spiel einen individuellen Stil.

KRIEGSSCHIFF SPIELER



FEINDLICHE RAUMSCHIFFE – GEGNER



MONSTER – GEGNER



BULLET

Feindliche raum schiff Bullet:



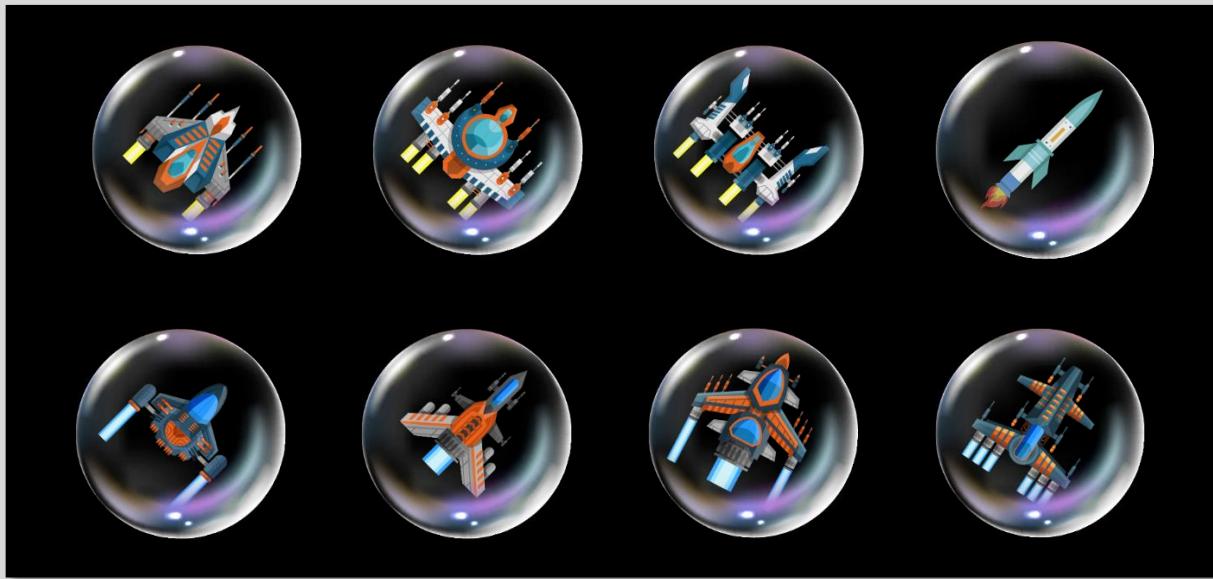
Monster Bullet:



Player Ballerets:



UPGRADE BUBBLES



USER INTERFACE

SYSTEM-STEUERUNG

Die Systemsteuerung in "Space Invaders" ermöglicht es, verschiedene Aspekte des Spiels anzupassen und zu steuern. Hier sind einige wichtige Einstellungen und Parameter, die für die Spielsteuerung verwendet werden:

- 1. MAX_LEVEL:** Dieser Wert definiert das maximale Level im Spiel. Es legt fest, wie viele Level der Spieler insgesamt durchlaufen kann.
- 2. LEVEL_SCORE:** Diese Konstante gibt an, wie viele Punkte der Spieler benötigt, um ein Level abzuschließen. Sobald der Spieler diese Punktzahl erreicht, steigt er in das nächste Level auf.
- 3. MAX_SCORE:** Dieser Wert berechnet sich aus der Multiplikation von LEVEL_SCORE und MAX_LEVEL. Es gibt die maximale Punktzahl im Spiel an, die der Spieler erreichen kann.
- 4. MONSTER_LIFE_TIME:** Diese Konstante definiert die Lebensdauer des Monsters im Spiel. Das Monster wird für eine bestimmte Zeitspanne sichtbar sein, bevor es wieder verschwindet.
- 5. MONSTER_SHOW:** Diese Einstellung gibt an, in welchem Level das Monster im Spiel erscheinen soll. Es wird im angegebenen Level sichtbar sein und stellt eine besondere Herausforderung für den Spieler dar.
- 6. NUR_BY_DEAD_WIN:** Diese Variable bestimmt, ob der Spieler das Spiel nur gewinnen kann, indem er das Monster besiegt. Wenn der Wert auf "false" gesetzt ist, kann der Spieler das Spiel auch gewinnen, indem er die maximale Punktzahl erreicht, und dazu den Monster besiegen.
- 7. SHOWROCKET_IN_SCORE_1 und SHOWROCKET_IN_SCORE_2:** Diese Variablen definieren die Punktzahlen, bei denen Raketen im Spiel angezeigt werden. Je nach Punktzahl werden Raketen für den Spieler verfügbar sein, um sie zu verwenden aber unter Bedingung das Monster schon sichtbar ist.

Diese Einstellungen und Parameter ermöglichen es den Entwicklern, das Spielerverhalten anzupassen und verschiedene Herausforderungen und Belohnungen für den Spieler zu definieren. Indem sie diese Werte ändern, können sie das Spiel anpassen und ein ausgewogenes Spielerlebnis schaffen.

```
1 export const MAX_LEVEL = 4;
2 export const LEVEL_SCORE = 15;
3 export const MAX_SCORE = LEVEL_SCORE * MAX_LEVEL;
4 export const MONSTER_LIFE_TIME = 10;
5 export const MONSTER_SHOW = MAX_LEVEL;
6
7 //nur bei monster tot gewinnen und unabhängig von maxscore
8 export const NUR_BY_DEAD_WIN = false;
9
10 // du kannst nur zwei rocket anzeigen und wenn Monszter ist sichtbar
11 export const SHOWROCKET_IN_SCORE_1 = MAX_SCORE - Math.floor(LEVEL_SCORE / 2);
12 export const SHOWROCKET_IN_SCORE_2 = (MAX_SCORE - LEVEL_SCORE) + 2;
13
```

SPIELER-STEUERUNG

Die Spieler haben zwei Möglichkeiten, das Kriegsschiff zu steuern:

3. Tastatursteuerung:

Bewegung:

- Mit den Pfeiltasten der Tastatur kann das Kriegsschiff horizontal und vertikal bewegt werden
- Wenn die linke Pfeiltaste gedrückt wird, bewegt sich das Schiff nach links.
- Wenn die rechte Pfeiltaste gedrückt wird, bewegt sich das Schiff nach rechts.
- Wenn die obere Pfeiltaste gedrückt wird, bewegt sich das Schiff nach oben.
- Wenn die untere Pfeiltaste gedrückt wird, bewegt sich das Schiff nach unten.
- Das Kriegsschiff stoppt seine Bewegung, wenn keine der Pfeiltasten gedrückt ist.

Abgefeuern:

- Durch Drücken der Leertaste werden Waffen abgefeuert.
- Durch Drücken der Taste "R" werden Raketen abfeuern. Wenn der Spieler genügend Raketen zur Verfügung hat, kann er diese Funktion nutzen.

4. Touchscreen-Steuerung:

Bewegung:

- Durch Berühren des Bildschirms und Ziehen des Fingers kann das Kriegsschiff in die gewünschte Richtung bewegt werden.
- Die Geschwindigkeit der Bewegung hängt von der Position des Fingers auf dem Bildschirm ab.
- Das Kriegsschiff stoppt seine Bewegung, wenn der Finger nicht mehr auf dem Bildschirm liegt.

Abfeuern:

- Durch Berühren des Bildschirms wird eine Waffe abgefeuert.

Die Spieler können die Steuerungsmethode wählen, die ihnen am besten liegt. Wenn sie das Spiel auf einem PC spielen, können sie die Tastatur verwenden, um das Kriegsschiff zu steuern. Wenn sie das Spiel auf einem mobilen Gerät wie einem Handy oder Tablet spielen, können sie das Kriegsschiff durch Berühren des Bildschirms steuern.

SPIELE LOGIK-ELEMENTE

GEGNER AI-MONSTER

Monster Bewegung: Das Monster nutzt eine Kombination aus horizontaler und vertikaler Bewegung, um den Spieler zu verfolgen und anzugreifen. Die Bewegung des Monsters basiert auf der Position des Spielers und der Bildschirmmitte.

Horizontale Bewegung: Das Monster bewegt sich horizontal in Richtung des Spielers. Wenn die X-Position des Spielers größer ist als die X-Position des Monsters, bewegt sich das Monster nach rechts. Ist die X-Position des Spielers kleiner, bewegt sich das Monster nach links. Die Geschwindigkeit der horizontalen Bewegung beträgt 120 Einheiten.

Vertikale Bewegung: Das Monster hat zwei Verhaltensweisen für die vertikale Bewegung:

- Wenn das Monster nach oben geht: Wenn das Monster gegen eine obere Kollisionsgrenze stößt, ändert es seine vertikale Bewegung und bewegt sich nach unten. Die Geschwindigkeit beträgt 130 Einheiten.
- Wenn das Monster die Mitte des Bildschirms überschreitet: Sobald das Monster die Mitte des Bildschirms erreicht oder überschreitet, ändert es seine vertikale Bewegung und bewegt sich nach oben. Die Geschwindigkeit beträgt -175 Einheiten.
- Wenn der Spieler sich unterhalb der Position des Monsters befindet: Wenn der Spieler sich unterhalb der aktuellen Position des Monsters befindet, bewegt sich das Monster nicht vertikal. Es behält seine Y-Geschwindigkeit von 0 Einheiten bei.

Durch die Kombination aus horizontaler und vertikaler Bewegung kann das Monster dem Spieler folgen und gleichzeitig eine dynamische Bewegung im Spiel bieten. Es erfordert vom Spieler strategisches Denken und Geschicklichkeit, um den Angriffen des Monsters auszuweichen und es letztendlich zu besiegen.

```
1  #moveMonster() {
2
3    const [width, height] = getWindowWidthAndHeight();
4    if (!this.#showMonstercheck) {
5      return;
6    }
7
8    var velocityX = 0;
9    var velocityY = 0;
10
11   // Das Monster bewegt sich horizontal, wie player
12   if (this.#player.x > this.#monster.x) {
13     velocityX = 120;
14   } else if (this.#player.x < this.#monster.x) {
15     velocityX = -120;
16   }
17
18   // Wenn der Spieler nach oben geht, folgt das Monster, solange es nicht die Mitte des Bildschirms überschreitet
19   if (this.#monster.body.blocked.up) {
20     this.#monster.setVelocityY(130);
21
22   } else if (this.#monster.y >= height / 3) {
23     // Wenn das Monster die Mitte des Bildschirms überschreitet, geht es nach unten
24     this.#monster.setVelocityY(-175);
25
26   } else if (this.#player.y <= this.#monster.y) {
27     // Das Monster bewegt sich nicht vertikal, wenn der Spieler nach unten geht
28     this.#monster.setVelocityY(0);
29   }
30 }
```

GEGNER AI- FEINDLICHEN RAUMSCHIFFE

Die feindlichen Raumschiffe in "Space Invaders" verwenden eine intelligente KI, um den Spieler herauszufordern und ihn zu besiegen. Die KI-Logik der feindlichen Raumschiffe basiert auf verschiedenen Funktionen und Ereignissen im Spiel.

Die Methode "createEnemy()" ist dafür verantwortlich, neue feindliche Raumschiffe zu erstellen. Diese Methode verwendet die Klasse "Enemy" und ruft deren "create()" Funktion auf. Dabei werden verschiedene Parameter übergeben, wie z.B. die Skalierung des feindlichen Raumschiffs, die Geschwindigkeit und der Typ des feindlichen Raumschiffs. Das feindliche Raumschiff wird an einer zufälligen Position am oberen Bildschirmrand erstellt und mit einer bestimmten vertikalen Geschwindigkeit versehen.

Die Methode "createEvents()" erstellt ein Ereignis, das regelmäßig neue feindliche Raumschiffe erstellt. Dabei wird die Häufigkeit der Erstellung basierend auf dem aktuellen Level des Spielers angepasst. Das Ereignis wird mit einem bestimmten Intervall gestartet und ruft die Methode "createEnemy()" auf, um ein neues feindliches Raumschiff zu erstellen. Dieses Ereignis wird in einer Schleife wiederholt, um kontinuierlich neue Herausforderungen für den Spieler zu generieren.

Die feindlichen Raumschiffe werden mit zunehmendem Level des Spielers anspruchsvoller. Die Skalierung und Geschwindigkeit der feindlichen Raumschiffe werden entsprechend angepasst, um den Schwierigkeitsgrad zu erhöhen. Dies sorgt für eine immer größer werdende Herausforderung für den Spieler, da die feindlichen Raumschiffe schneller und schwieriger zu besiegen sind.

Die intelligente KI der feindlichen Raumschiffe stellt sicher, dass das Spiel eine abwechslungsreiche und spannende Erfahrung bietet. Die feindlichen Raumschiffe bewegen sich auf intelligente Weise und greifen den Spieler an, während sie versuchen, den Angriffen des Spielers auszuweichen. Der Spieler muss seine Fähigkeiten und Strategien einsetzen, um die feindlichen Raumschiffe zu besiegen und das Spiel erfolgreich abzuschließen.

```
1 #createEnemy() {
2     Enemy.create(
3         this.#enemies,
4         0.2 / (this.#level / 4),
5         50 * (this.#level / 2 + 1),
6         `enemy-${this.#level}`
7     );
8 }
```

```
1 export class Enemy {
2     static create (enemies, scale, velocity, enemyType) {
3         const enemy = enemies.create(Phaser.Math.Between(50, getWindowWidthAndHeight()[0] - 50), -20, enemyType);
4         enemy.setScale(scale);
5         enemy.setVelocityY(velocity);
6         return enemy
7     }
8 }
```

```
1 #createEvents() {
2     this.#events.createEnemyEvent = this.time.addEvent({
3         delay: 2000 / this.#level,
4         callback: this.#createEnemy,
5         callbackScope: this,
6         loop: true
7     });
8     ...
9 }
```

WAFFEN

DIE FEINDLICHEN RAUMSCHIFFE

in dem Spiel verfügen über intelligente Waffen, die den Spieler herausfordern und angreifen. Hier ist die Erklärung, wie die feindlichen Raumschiffe ihre Waffen abfeuern:

Die Methode `shootBulletFromEnemy()` ist verantwortlich für das Schießen der feindlichen Raumschiff-Bullets. Zuerst wird ein zufälliges feindliches Raumschiff ausgewählt, das aktiv ist und sich noch innerhalb des Bildschirms befindet. Falls kein passendes feindliches Raumschiff gefunden wird, wird die Methode abgebrochen.

Danach wird eine feindliche Bullet erstellt und an der Position des zufälligen feindlichen Raumschiffs platziert. Die Größe der Bullet wird auf 0.3 skaliert, um die Darstellung anzupassen.

Die Richtung der Bullet wird berechnet, indem ein Vektor zwischen dem Spieler und dem zufälligen feindlichen Raumschiff erstellt und normalisiert wird. Dadurch wird die Richtung der Bullet ermittelt. Die Bullet erhält dann eine Geschwindigkeit basierend auf dieser Richtung, um in Richtung des Spielers zu fliegen.

Ein Schuss-Soundeffekt wird abgespielt.

MONSTER-WAFFEN

Das Monster in "Space Invaders" verfügt ebenfalls über mächtige Waffen, die dem Spieler zusätzliche Herausforderungen bieten. Hier ist die Erklärung, wie das Monster seine Waffen abfeuert:

Die Methode `shootBulletFromEnemy()` wird auch für das Monster verwendet, um seine Bullets abzufeuern. Zunächst wird eine Bullet an der Position des Monsters erstellt und auf die Größe von 0.4 skaliert. Ein weiterer Schuss-Soundeffekt wird abgespielt.

Ähnlich wie bei den feindlichen Raumschiffen wird die Richtung der Bullet berechnet, indem ein Vektor zwischen dem Spieler und dem Monster erstellt und normalisiert wird. Dadurch erhält die Bullet ihre Flugrichtung. Die Bullet erhält eine höhere Geschwindigkeit als die der feindlichen Raumschiffe, um die erhöhte Schwierigkeit und Gefahr des Monsters widerzuspiegeln.

Die feindlichen Raumschiffe und das Monster feuern ihre Bullets in regelmäßigen Abständen ab, um den Spieler kontinuierlich unter Druck zu setzen und ein herausforderndes Spielerlebnis zu bieten. Die Spieler müssen geschickt ausweichen und zurückschießen, um die Bedrohung zu bewältigen und das Weltraumabenteuer zu überstehen.

```
1  #shootBulletFromEnemy() {
2    const [_, height] = getWindowWidthAndHeight();
3
4    const randomEnemy = Phaser.Utils.Array.GetRandom(
5      this.#enemies.getChildren().filter(e => e.active == true && e.y <= height)
6    );
7    if (!randomEnemy) {
8      return;
9    }
10   const enemybullet = this.#bulletsEnemies.create(randomEnemy.x, randomEnemy.y, "bulletEnemy");
11
12   enemybullet.setScale(0.3);
13   this.#sounds.shoot.play();
14   const direction = new Phaser.Math.Vector2(
15     this.#player.x - randomEnemy.x,
16     this.#player.y - randomEnemy.y
17   ).normalize();
18   enemybullet.setVelocity(direction.x * 500, direction.y * 500);
19
20   if (this.#monster.visible) {
21
22     const monsterBullet = this.#bulletsEnemies.create(
23       this.#monster.x,
24       this.#monster.y,
25       "bulletMonster"
26     );
27     monsterBullet.setScale(0.4);
28     const direction = new Phaser.Math.Vector2(
29       this.#player.x - this.#monster.x,
30       this.#player.y - this.#monster.y
31     ).normalize();
32     monsterBullet.setVelocity(direction.x * 800, direction.y * 800);
33     this.#sounds.shoot.play();
34
35   }
36
37 }
```

Spieler-Waffen

Der Spieler in "Space Invaders" hat verschiedene Waffen zur Verfügung, um gegen die feindlichen Raumschiffe und das Monster zu kämpfen. Hier sind die verschiedenen Waffen des Spielers und wie sie funktionieren:

NORMALE BULLET:

Die Methode `#shootBullet()` ist für das Abfeuern der normalen Geschosse des Spielers verantwortlich. Wenn der Spieler die Schusstaste drückt, wird eine normale Bullet erstellt und von der aktuellen Position des Spielers aus in Richtung der oberen Bildschirmkante abgefeuert. Die Type der Bullet wird basierend auf der Upgradestufe des Spieler-Bullets festgelegt. Die Geschwindigkeit der Bullet ist negativ, da sie nach oben fliegen muss.

Die `Bullet`-Klasse enthält eine statische Methode `create()`, die dazu dient, normale Geschosse zu erstellen. Sie wird mit den notwendigen Parametern wie der Bullets-Gruppe, der Position, Skalierung und Geschwindigkeit aufgerufen.



```
1
2  export class Bullet {
3      static create(bullets, x, y, scale, velocity, bulletType) {
4          const bullet = bullets.create(x, y, bulletType)
5          bullet.setScale(scale);
6          bullet.setVelocityY(velocity);
7          return bullet;
8      }
9  }
```

RAKETENWERFER:

Die Methode `#shootRocket()` ermöglicht es dem Spieler, den Raketenwerfer einzusetzen. Wenn der Spieler die Raketenwerfertaste drückt und noch Raketen übrig hat, wird eine Rakete erstellt und von der aktuellen Position des Spielers aus in Richtung des Monsters abgefeuert. Die Größe der Rakete wird festgelegt, und die Rakete erhält eine Geschwindigkeit, um sich auf das Monster zu bewegen.

Die Richtung der Rakete wird mithilfe eines Vektors berechnet, der zwischen dem Spieler und dem Monster erstellt und normalisiert wird. Dadurch erhält die Rakete ihre Flugrichtung in Richtung des Monsters.

Die `Rocket`-Klasse enthält eine statische Methode `create()`, die dazu dient, Raketen zu erstellen. Sie wird mit den notwendigen Parametern wie der Rockets-Gruppe, der Position, Skalierung, Geschwindigkeit und dem Zielmonster aufgerufen.

```
● ● ●  
1 import * as Phaser from 'phaser';  
2  
3 export class Rocket {  
4     static create(player, monster, rockets, x, y, scale, velocity, rocketType) {  
5         const rocket = rockets.create(x, y, rocketType)  
6         rocket.setScale(scale);  
7  
8         const direction = new Phaser.Math.Vector2(  
9             player.x - monster.x,  
10            player.y - monster.y  
11        ).normalize();  
12        rocket.setVelocity(-direction.x * velocity, -direction.y * velocity);  
13        return rocket;  
14    }  
15  
16  
17 }  
18 }
```

```
● ● ●  
1 //player bullet  
2 #shootBullet() {  
3     this.#bullet = Bullet.create(this.#bullets, this.#player.x, this.#player.y, 0.9, -1000, `bullet${Math.min(this.#upgradebulletPlayer, 4)}`)  
4 }  
5 #shootRocket() {  
6     this.#rocket = Rocket.create(this.#player, this.#monster, this.#rockets, this.#player.x, this.#player.y, 0.3, 1000, 'rocket')  
7     this.#sounds.shootRakete.play();  
8 }
```

Bedienung: Um die normale Bullet abzufeuern, muss der Spieler die Schusstaste drücken (Space-Taste). Wenn der Spieler den Raketenwerfer einsetzen möchte und noch Raketen zur Verfügung hat, kann er die R-Taste drücken, um eine Rakete abzufeuern.

GRAFIK ELEMENTE

HINTERGRÜNDE

Die Hintergrundgestaltung in "Space Invaders" spielt eine zentrale Rolle, um den Spielern ein fesselndes und immersives Erlebnis zu bieten. Drei Hauptelemente werden für den Hintergrund verwendet: Sterne, Meteoriten und Leuchtkörper (Luminaries). Jedes dieser Elemente trägt dazu bei, den Weltraum lebendig und dynamisch zu gestalten.

Die Sterne werden als sich wiederholendes Hintergrundmuster dargestellt. Durch die kontinuierliche Aktualisierung der Y-Position (`tilePositionY`) mit `this.#level` werden die Sterne langsam nach unten bewegt. Diese Bewegung erzeugt den Eindruck, dass sich das Raumschiff tatsächlich durch den Weltraum bewegt. Indem die Geschwindigkeit der Sterne mit dem Level ansteigt, kann man eine verstärkte Bewegung im Hintergrund erzeugen. Das beschleunigte Vorbeiziehen der Sterne lässt den Weltraum intensiver und dynamischer erscheinen, was dem Spieler das Gefühl vermittelt, dass er sich in einem sich verändernden und anspruchsvoller Umfeld befindet.

Die Meteoriten fliegen von rechts nach links und fallen dabei nach unten, bis sie den sichtbaren Bildschirmbereich verlassen. Die kontinuierliche Bewegung der Meteoriten verleiht dem Hintergrund eine lebendige Atmosphäre.

Die Leuchtkörper (Luminaries) hingegen werden von oben nach unten erzeugt und bewegen sich mit einer zufälligen Geschwindigkeit und Drehung. Die fließende Bewegung der Leuchtkörper verleiht dem Hintergrund eine weitere Dimension und erzeugt eine mysteriöse Stimmung.

Die sorgfältige Gestaltung der Hintergrundelemente in "Space Invaders" trägt maßgeblich dazu bei, dass das Spielerlebnis spannend und ansprechend wird. Die bewegten Sterne, Meteoriten und Leuchtkörper erschaffen eine lebensechte Umgebung, die den Spielern das Gefühl vermittelt, sich wirklich in den Weiten des Weltraums zu befinden und sich den immer schwieriger werdenden Herausforderungen zu stellen. Ein gut gestalteter Hintergrund kann die Immersion verbessern und das Spielerlebnis zu etwas Besonderem machen.

```
 1 #createBackground() {
 2     const [width, height] = getWindowWidthAndHeight();
 3     //create stars
 4     this.#background.stars = this.add
 5         .tileSprite(0, 0, width, height, "stars")
 6         .setOrigin(0)
 7         .setScrollFactor(0.5);
 8     //create meteors
 9     this.#background.meteors = this.physics.add
10         .sprite(width, 0, 'meteors')
11         .setScale(0.3);
12     this.#background.luminaries = this.physics.add.group();
13 }
14
15 #createLuminary() {
16     const [width, height] = getWindowWidthAndHeight();
17     const speed = Phaser.Math.Between(50, 100);
18     const direction = Math.random() < 0.5 ? -1 : 1;
19
20     const x = Phaser.Math.Between(0, width);
21     this.#background.luminaries
22         .create(x, -10, 'luminary')
23         .setScale(0.2)
24         .setVelocityY(speed)
25         .setVelocityX(speed * direction)
26         .setAngularVelocity(Phaser.Math.Between(-200, 200));
27     this.#background.luminaries.setDepth(-1);
28
29 }
30 #moveMeteors() {
31     const [width, height] = getWindowWidthAndHeight();
32
33     this.#background.meteors.x -= 2;
34     this.#background.meteors.y += 1;
35
36     if (this.#background.meteors.x < 0 || this.#background.meteors.y > height) {
37         this.#background.meteors.x = width;
38         this.#background.meteors.y = 0;
39     }
40 }
```

SPRITES

In meinem Spiel habe ich Explosion erstellt, um spannende visuelle Effekte zu erzeugen. Um diese Animationen zu realisieren, habe ich die Methode `#createExplosionAnimations()` implementiert. Diese Methode ist dafür verantwortlich, eine spezifische Animation mit dem Namen "explode" zu erstellen, die aus einer Reihe von aufeinanderfolgenden Bildern besteht.

Zuerst habe ich für jede Explosion eine Sammlung von Bildern vorbereitet, die den Verlauf der Animation darstellen. Diese Bilder habe ich in einem sogenannten "Spritesheet" zusammengefasst. Ein

Spritesheet ist eine einzelne Bilddatei, die mehrere Frames oder Bilder enthält, die in einer bestimmten Abfolge animiert werden können.



In der Methode `#createExplosionAnimations()` habe ich definiert, welches Spritesheet für die Animation verwendet wird ("explosion"). Dann habe ich festgelegt, dass die Animation die Frames 0 bis 6 aus dem Spritesheet verwendet, um die Explosion darzustellen.

Die Frame-Rate der Animation wurde auf 500 Frames pro Sekunde festgelegt. Dies bedeutet, dass jedes Bild in der Animation für 1/500 Sekunde (oder 0,002 Sekunden) angezeigt wird, bevor es zum nächsten Bild übergeht. Dadurch entsteht der Eindruck einer fließenden Bewegung und die Explosion wirkt realistisch.

Die Animation wird nur einmal abgespielt und nicht wiederholt, was bedeutet, dass sie sich nicht endlos wiederholt, sondern nach einmaligem Abspielen stoppt. Außerdem wird das Spielobjekt, das die Animation abspielt (z. B. das Explosionssprite), automatisch ausgeblendet, sobald die Animation vollständig abgespielt wurde.

Im Spiel rufe ich diese Methode auf, wann immer eine Explosion erzeugt werden soll. Wenn der Spieler beispielsweise einen Gegner besiegt oder wenn eine Kollision zwischen Objekten im Spiel auftritt, wird die Methode `#createExplosionAnimations()` aufgerufen, um die Explosion zu erstellen und abzuspielen.

Durch die Verwendung von Animationen wird das Spielerlebnis lebendiger und dynamischer. Die Explosionen ziehen die Aufmerksamkeit des Spielers auf sich, machen das Spiel interessanter und tragen zur Immersion bei, indem sie dem Spieler klare visuelle Rückmeldungen über das Geschehen im Spiel geben.

```
1 #createExplosionAnimations() {
2     this.anims.create({
3         key: "explode",
4         frames: this.anims.generateFrameNumbers("explosion", {
5             start: 0,
6             end: 6,
7         }),
8         frameRate: 500,
9         repeat: 0,
10        hideOnComplete: true,
11    });
12 }
```

SOUND / MUSIK

HINTERGRUNDMUSIK

Die Hintergrundmusik in "Space Invaders" spielt eine entscheidende Rolle dabei, die Atmosphäre des Spiels zu gestalten und die Spieler in das fesselnde Weltraumabenteuer einzutauchen. In den ersten beiden Szenen, der Game Info Scene und der Splash Scene, ist die Hintergrundmusik präsent, um die Spieler auf das bevorstehende Spiel vorzubereiten und eine angenehme Einführung zu bieten.

Game Info Scene: Die Hintergrundmusik in dieser Szene ist eingängig und motivierend, um die Spieler zu begeistern und auf das kommende Abenteuer einzustimmen. Sie schafft Vorfreude und steigert die Neugierde der Spieler. Die Musik stoppt, sobald der Spieler auf "Play" klickt. Datei: "assets/sounds/background2.mp3"

Splash Scene: Die Splash Scene ist die Eröffnungsszene des Spiels und zeigt das Logo und den Namen von "Space Invaders". Die Hintergrundmusik in dieser Szene ist episch und beeindruckend, um einen starken ersten Eindruck zu hinterlassen und die Spieler in die Weltraumthematik einzuführen. Die Musik spielt automatisch und endet nach 8,5 Sekunden, am Ende der Szene, ohne dass der Spieler etwas tun muss. Dadurch wird eine kurze, aber wirkungsvolle Einführung in das Spiel ermöglicht. Datei: "assets/sounds/background1.mp3"

SOUND-EFFEKTE

Soundeffekte sind ein wichtiger Bestandteil des Spiels und tragen dazu bei, die Spielererfahrung in der Game Scene zu intensivieren. Hier sind die verschiedenen Soundeffekte und ihre Funktionen im Spiel:

- **Shoot:** Dieser Soundeffekt wird abgespielt, wenn der Spieler eine normale Schussaktion ausführt, um die feindlichen Raumschiffe zu bekämpfen.
- **Shoot Rakete:** Wenn der Spieler eine Rakete abfeuert, um mächtigere Feinde zu besiegen, ertönt dieser Soundeffekt.
- **Enemy Death:** Wenn ein feindliches Raumschiff besiegt wird, erklingt dieser Soundeffekt, um den Spieler zu belohnen und die Zerstörung des Feindes zu verdeutlichen.
- **Explosion:** Dieser Soundeffekt wird abgespielt, wenn das Raumschiff des Spielers entweder von einem feindlichen Angriff getroffen wird und explodiert oder wenn der Spieler mit feindlichen Raumschiffen kollidiert.
- **Monster Hit:** Wenn ein feindliches Monster getroffen wird, spielt dieser Soundeffekt, um den Treffer zu markieren und das Spielerlebnis zu verstärken.
- **Monster Explosion:** Wenn ein feindliches Monster besiegt wird und explodiert, wird dieser Soundeffekt verwendet, um die Action und Zerstörung zu unterstreichen.

Durch das geschickte Einsetzen dieser Soundeffekte in der Game Scene wird die Immersion und die emotionale Reaktion der Spieler während des Spielens verstärkt. Die richtigen Soundeffekte können die Spielwelt lebendiger machen und das Gameplay noch fesselnder gestalten.

LEVEL

Das Spiel besteht aus insgesamt vier Levels, und zu Beginn der Level 2, 3 und 4 erscheint eine Upgrade Bubble, die es dem Spieler ermöglicht, sein Raumschiff zu verbessern. Der Spieler hat die Möglichkeit, die Feuerkraft seines Raumschiffs zu verstärken und mehr Bullet-Shots abzufeuern.

LEVEL 1 - DIE INVASION BEGINNT:

Im ersten Level startet die Alien-Invasion, und der Spieler muss sich den anrückenden außerirdischen Eindringlingen stellen. Der Spieler beginnt mit seinem Basisraumschiff und der Standardfeuerkraft. Das Ziel ist es, so viele Eindringlinge wie möglich zu besiegen, um Punkte zu sammeln. Um das Level erfolgreich abzuschließen, muss der Spieler 15 Punkte erreichen.

LEVEL 2 - BULLET-UPGRADE VERFÜGBAR:

Im zweiten Level wird die Bedrohung noch größer, und du benötigst dringend eine stärkere Feuerkraft, um die wachsende Anzahl von Eindringlingen abzuwehren. Doch keine Sorge, eine Upgrade Bubble erscheint zu Beginn dieses Levels. Sobald du sie einsammelst, verwandelt sich dein Raumschiff und deine Feuerkraft wird verbessert. Du kannst nun mehr Bullet-Shots auf deine Feinde abfeuern! Stelle dich erneut der Herausforderung und erreiche 15 Punkte, um zum nächsten Level zu gelangen.

LEVEL 3 - STÄRKERE FEUERKRAFT:

Jetzt wird es richtig intensiv! Im Level 3 wird die Alien-Invasion noch gefährlicher. Doch bevor die Feinde über dich hereinbrechen, erscheint eine weitere Upgrade Bubble. Sammle sie ein, um die Feuerkraft deines Raumschiffs weiter zu verstärken und eine beeindruckende Menge an Bullet-Shots abfeuern zu können. Es wird schwieriger, die Eindringlinge zu besiegen, aber du darfst nicht nachgeben! Sammle erneut 15 Punkte, um das finale Level zu erreichen.

LEVEL 4 - SHOWDOWN MIT DEM MONSTER:

Das vierte und letzte Level erreicht den Höhepunkt der Spannung! Hier siehst du dich einem riesigen Alien-Monster gegenüber, das die gesamte Galaxie bedroht. Doch bevor der epische Showdown beginnt, erscheint eine letzte Upgrade Bubble für dein Raumschiff. In dieser Level bekommst du etwas ganz Besonderes: Zwei mächtige Rocket-Upgrades! Diese Rockets sind unglaublich stark und fliegen direkt auf das Monster zu, um ihm enormen Schaden zuzufügen.

Mit dieser beeindruckenden Feuerkraft ausgestattet, kannst du das Monster mit Rocket-Shots attackieren. Die Rockets sind weitaus stärker als die normalen Bullet-Shots und bringen das Monster gehörig ins Wanken.



**Mohamad Alaskari
5244256**