**Pflichtenheft zum Spiel OceanBomber**

Im Folgenden werden alle wichtigen Klassen mit ihren wichtigen Attributen und Methoden genannt, die für die Implementierung wichtig sind.

**Flugzeug**

**Beschreibung:**

Flugzeug ist die Spielfigur die vom Spieler gesteuert wird und mit dem man die Gegnerischen Flugzeuge (Klasse: attacking) durch abschießen und die gegnerischen Boote (Klasse: Boot) durch den Abwurf von Bomben zerstören muss.

Zu dem Flugzeug gehört außerdem der Propeller (Klasse: Propeller), der für die Animation des Propellers verantwortlich ist.

Das Flugzeug des Spielers kann durch eine Auswahl im Menü verschiedene Aussehen erhalten.

Das Flugzeug befindet sich immer in der Mitte vom Bildschirm und der Rest bewegt sich außen rum.

**Wichtige Methoden sind:**

**touched():** Überprüft, ob das Flugzeug des Spielers von einem Schuss getroffen wird und zieht, wenn das der Fall ist ein Leben von der Lebensanzeige (Klasse: Lebensanzeige) ab.

**shoot():** Methode um Schüsse abzufeuern, um die gegnerischen Flugzeuge (Klasse: attacking) zu zerstören.

**steuerung():** Legt die Steuerung des Flugzeuges fest.

**Wichtige Attribute sind:**

**lastShot:** Integer der die Schussgeschwindigkeit festlegt.

**Propeller**

**Beschreibung:**

Propeller ist Teil der Grafik der Spielerfigur (Klasse: Flugzeug) und sorgt durch eine Abfolge von mehreren Bildern für die Illusion, dass sich der Propeller des Flugzeug dreht.

**Wichtige Methoden sind:**

**propellerFolgt():** Sorgt dafür, dass der Propeller des Flugzeugs sich immer an der Position des Flugzeug (Klasse: Flugzeug) befindet und die selbe Rotation/ Ausrichtung besitzt.

**propellerDrehen():** Sorgt für eine Abfolge von Bildern, um die Illusion von einem sich drehenden Rotor zu erzeugen.

**Wichtige Attribute sind:**

**DREHGESCHWINDIGKEIT:** Integer, der die Geschwindigkeit festlegt, mit der sich der Rotor dreht.

**counter:** Integer der die Ausgeführten Acts zählt.

**Boden**

**Beschreibung:**

Boden ist der Spielbereich in dem sich das Flugzeug des Spielers bewegen kann. Der Spielbereich wird durch den Rand (Klasse: BodenRot) eingegrenzt, in den der Spieler nicht rein fliegen darf.

**Wichtige Methoden sind:**

**getMapFaktor():** Getter um den Faktor mit dem die Größe der Grafik des Bodens multipliziert wurde zu erfahren.

**Wichtige Attribute sind:**

**mapFaktor:** Integer, der den Multiplikator mit dem die Größe der Grafik des Bodens multipliziert wird festlegt.

**BodenRot**

**Beschreibung:**

BodenRot ist der Rand des Spielbereichs (Klasse: Boden) in den der Spieler nicht reinfliegen darf, da er sonst stirbt.

**Wichtige Methoden sind:**

**Wichtige Attribute sind:**

**BodenDurchsichtig**

**Beschreibung:**

BodenDurchsichtig steuert die Bewegung vom Spielbereich (Klasse: Boden) und dessen Rand (Klasse: BodenRot).

**Wichtige Methoden sind:**

**bewegen(int):** Methode mit der BodenDurchsichtig und somit das Spielfeld in die entgegengesetzte Richtung der Bewegung des Spielers (Klasse: Flugzeug) bewegt wird und somit die Illusion entsteht, dass sich das Flugzeug des Spielers bewegt.

**bodenFolgt():** Methode, die dafür sorgt, dass sich das Spielfeld (Klasse: Boden) und dessen Rand (Klasse: BodenRot) immer an der Position von BodenDurchsichtig befinden und sie sich somit entgegengesetzt der Bewegung von dem Flugzeug bewegt.

**getGeschwindigkeit():** Getter um die Geschwindigkeit mit der sich der Spieler (Klasse: Flugzeug) bewegt zu erfahren.

**setGeschwindigkeit(int):** Setter um die Geschwindigkeit festzulegen, mit der sich das Flugzeug des Spielers (Klasse: Flugzeug) bewegt.

**Wichtige Attribute sind:**

**geschwindigkeit:** Integer, der die Geschwindigkeit des vom Spieler gesteuerten Flugzeugs (Klasse: Flugzeug) festlegt.

**attacking**

**Beschreibung:**

Attacking ist das gegnerische Flugzeug, dass versucht den Spieler (Klasse: Flugzeug) abzuschießen und vom Spieler (Klasse: Flugzeug) abgeschossen werden muss, damit er gewinnen kann.

**Wichtige Methoden sind:**

**touched():** Überprüft, ob das Flugzeug des Spielers von einem Schuss getroffen wird und löscht das Objekt, wenn das der Fall ist.

**shoot():** Methode um Schüsse abzufeuern.

**turntoplain():** Methode die, dafür sorgt, dass das attacking immer in Richtung vom Flugzeug des Spielers (Klasse: Flugzeug) fliegt.

**Wichtige Attribute sind:**

**speed:** Integer, der die Geschwindigkeit von attacking festlegt.

**lastShot:** Integer der die Schussgeschwindigkeit festlegt.

**Boot**

**Beschreibung:**

Boot ist das gegnerische Boot, dass durch den Spieler (Klasse: Flugzeug) durch eine Bombe zerstört werden muss, damit er gewinnen kann.

Auf einem Boot Befindet sich immer ein Geschütz (Klasse: Geschuetz), dass versucht den Spieler (Klasse: Flugzeug) abzuschießen.

**Wichtige Methoden sind:**

**setLocation2(int, int):** Methode um PowerUps an einen neue Stelle zu zuweisen.

**welt():** Ist eine Referenz auf die Welt.

**Wichtige Attribute sind:**

**xPosition:** Integer, der x-Position auf dem Spielfeld (Klasse: Boden).

**yPosition:** Integer, der y-Position auf dem Spielfeld (Klasse: Boden).

**rotation:** Integer, der die Rotation/ Ausrichtung des Bootes bestimmt.

**Lebend:** Boolean, der bestimmt ob das Boot zerstört wurde, oder nicht.

**EXPLOSIONSLAENGE:** Integer, der bestimmt wie lange das Bild eines zerstörten Bootes angezeigt werden soll.

**counter:** Integer der die Ausgeführten Acts zählt.

**Geschuetz**

**Beschreibung:**

Geschuetz ist das Geschütz, dass sich auf einem Boot (Klasse: Boot) befindet und versucht den Spieler (Klasse: Spieler) abzuschießen.

**Wichtige Methoden sind:**

**shoot():** Methode um Schüsse abzufeuern.

**folgtBoot():** Methode die, dafür sorgt, dass sich das Geschütz an der Richtigen Stelle auf dem Boot befindet.

**Wichtige Attribute sind:**

**lastShot:** Integer der die Schussgeschwindigkeit festlegt.

**PowerUps**

**Beschreibung:**

PowerUps sind Hilfsmittel für den Spieler (Klasse: Flugzeug) die durch hinüberfliegen eingesammelt werden können. Sie können den Spieler auf Zeit unverwundbar (Schild), schneller (Schnelligkeit) machen oder ihm ein weiteres Leben/ Herz geben (Herz).

**Wichtige Methoden sind:**

**setLocation2(int, int):** Methode um PowerUps an einen neue Stelle zu zuweisen.

**welt():** Ist eine Referenz auf die Welt.

**Wichtige Attribute sind:**

**powerUpName:** String, der die Funktion/Art des PowerUps bestimmt.

**xPosition:** Integer, der x-Position auf dem Spielfeld (Klasse: Boden).

**yPosition:** Integer, der y-Position auf dem Spielfeld (Klasse: Boden).

**bewegen:** Boolean, der festlegt ob PowerUps eingesammelt wurde oder nicht.

**counter:** Integer der die Ausgeführten Acts zählt.

**ABBLINKZEIT:** Integer, der die Zeit bestimmt, wie lange der Spieler (Klasse: Flugzeug) das PowerUp behalten soll.

**Shot**

**Beschreibung:**

Shot ist ein Schuss der von den Flugzeugen (Klasse: Flugzeug, attacking) und dem Boot (Klasse: Boot) abgefeuert wird um andere zu zerstören.

**Wichtige Methoden sind:**

**isFrom(Actor):** Methode um den Schützen zu erfahren, der den Schuss abgefeuert hat.

**Wichtige Attribute sind:**

**attacker:** Referenz au den Schützen.

**flew:** Integer, der die geflogene Strecke angibt.

**speed:** Integer, der die Geschwindigkeit des Schusses bestimmt.

**LebensAnzeige**

**Beschreibung:**

LebensAnzeige zeigt die Leben des Spielers (Klasse: Flugzeug) an. Die Leben können maximal drei sein und minimal null sein.

**Wichtige Methoden sind:**

**setLeben(int): Setter um die Anzahl der Leben zu setzen.**

**getLeben(): Getter um die Anzahl der Leben zu erfahren.**

**lebenAnzeigen(): Methode um die Anzeige zu aktualisieren, die die Leben anzeigt.**

**lebenAbziehen(): Methode um ein Leben abzuziehen.**

**lebenHinzufuegen(): Methode um ein Leben hinzuzufügen.**

**Wichtige Attribute sind:**

**leben:** Integer, der die Lebensanzahl festlegt.

**Schrift**

**Beschreibung:**

Schrift ermöglicht es einen beliebigen Text auf dem Bildschirm erscheinen zu lassen. Definiert außerdem die Funktion/ den Text der angezeigt werden soll.

**Wichtige Methoden sind:**

**textAnzeigen(String, int):** Methode um den Text zu definieren, der in einer bestimmten Größe angezeigt werden soll.

**levelAnzeigen(int):** Methode um das ausgewählte Level in einer bestimmten Schriftgröße anzuzeigen.

**warnungAnzeigen(int):** Methode um die Warnung anzuzeigen, die erscheint wenn der Spieler (Klasse: Flugzeug) zu nah am Rand des Spielfelds (Klasse: BodenRot) ist.

**nameEingeben():**Methode, die einen Text (in dem Fall der Name) der eingeben wird mit Curser anzeigt und die Eingabe ermöglicht.

**getName():** Getter um den Name zu erfahren, der eingegeben wurde.

**Wichtige Attribute sind:**

**name:** String, der den eingegebenen Namen speichert.

**taste:** String, der die zuletzt gedrückte Taste speichert.

**buchstaben:** Integer, der die Anzahl von bereits eingegebenen Buchstaben speichert.

**counter:** Integer der die Ausgeführten Acts zählt.

**strich:** Boolean, der Bestimmt ob der Curser gerade sichtbar ist oder nicht.

**schriftName:** String, der die Funktion und somit bestimmt, was für ein Text angezeigt werden soll.

**BLINKEN:** Integer, der bestimmt wie schnell der Curser blinkt.

**MAXBUCHSTABEN:** Integer, der die maximale Buchstabenzahl bestimmt, die eingegeben werden darf.

**Tasten**

**Beschreibung:**

Tasten ist die Klasse die für alle Tasten auf dem Bildschirm verantwortlich ist, die mit der Maus angeklickt werden können und deren Funktion/ die Aktionen die ausgelöst wird.

**Wichtige Methoden sind:**

**mausScale():** Methode die Taste vergrößert, wenn sich der Mauszeiger auf dieser befindet.

**mausNeuScale():** Methode die den Faktor für die Vergrößerung der Methode mausScale() festlegt.

**reglerBewegen():** Methode die dafür sorgt, dass sich der Regler, der für die Lautstärke zuständig ist, bewegen lässt.

**welt():** Ist eine Referenz auf die Welt.

**Wichtige Attribute sind:**

**Menue**

**Beschreibung:**

Menue ist der Hintergrund für alle Menüs (Einstellung, Hauptmenü, Pause, GameOverBildschirm,...). Außerdem ist Menue die Klasse in der Sämtliche Informationen, wie das ausgewählte Level, die Lautstärke, der Spielzustand, die Erfolge (Informationen, die im Profil angezeigt werden),... gespeichert werden. Des Weiteren ist es die Klasse die für alle Sounds verantwortlich ist.

**Wichtige Methoden sind:**

Setter und Getter für fast alle Attribute dieser Klasse und Methoden um die verschiedenen Geräusche abzuspielen oder zu stoppen.

**Wichtige Attribute sind:**

**int breite:** Integer, der die Breite der Grafik bestimmt.

**breiteNeu:** Integer, der die Breite des Bildes nach der Vergrößerung bestimmt

**hoehe:** Integer, der die Höhe der Grafik bestimmt.

**hoeheNeu:** Integer, der die Höhe des Bildes nach der Vergrößerung

**spielZustand:** String der den Zustand des Spiels bestimmt.

**skin:** Integer, der den Skin/ das Aussehen des Flugzeugs des Spielers bestimmt (Klasse: Flugzeug).

**level:** Integer, der das ausgewählte Level bestimmt.

**counter:** Integer der die Ausgeführten Acts zählt.

**musikVolumen:** Integer der die Lautstärke der Hintergrundmusik bestimmt.

**soundVolumen:** Integer der die Lautstärke der restlichen Geräusche bestimmt (alles außer der Hintergrundmusik).

**grafik:** Boolean, der bestimmt, ob Animationen (Bewegung der Propeller) an oder aus sind.

**tode:** Integer, der die Anzahl der Tode speichert.

**abschuesse:** Integer, der die Anzahl der Kills speichert.

**verloren:** Integer, der die Anzahl der verlorenen Level speichert.

**gewonnen:** Integer, der die Anzahl der gewonnen Level speichert.

**heard:** Integer, der die Anzahl der eingesammelten PowerUps der Art Herz speichert.

**shield:** Integer, der die Anzahl der eingesammelten PowerUps der Art Schilder speichert.

**speed:** Integer, der die Anzahl der eingesammelten PowerUps der Art Schnelligkeit speichert.

**name: String, der** den Namen des Spielers speichert.

**LEVELANZAHL:** Integer, der die Anzahl der Level bestimmt.

**SKINANZAHL:** Integer, der die Anzahl der Skins bestimmt.

**Welt**

**Beschreibung:**

Welt ist die Klasse in der vermerkt ist, wann welches Objekt erschaffen werden muss.

**Wichtige Methoden sind:**

**menueErstellen():** Erstellt alle Objekte die für die Oberfläche des Menüs wichtig sind und setzt das Spieler Flugzeug richtig.

**spielErstellen():** Erstellt alle Objekte die für die Oberfläche des Spiels wichtig sind und setzt das Spieler Flugzeug richtig.

**neuesSpielErstellen():** Erstellt alle Objekte die für die Oberfläche des Spiels wichtig sind und setzt das Spieler Flugzeug richtig.

**pauseErstellen():** Erstellt alle Objekte die für die Oberfläche des PauseMenüs wichtig sind und löscht die PauseTaste.

**winErstellen():** Erstellt alle Objekte die für die Oberfläche des WinMenüs wichtig sind und löscht die PauseTaste.

**gameOverErstellen():** Erstellt alle Objekte die für die Oberfläche des GameOverMenüs wichtig sind und löscht die PauseTaste.

**allesLoeschen():** Löscht alle Objekte bis auf das Spieler Flugzeug und den Boden.

**next():** Startet das nächste Level.

**menueUndFlugzeug():** Erstellt das Menü mit dem Flugzeug nach dem der Spieler den Namen eingegeben hat.

**schriftLoeschen():** Löscht alle Texte, die Angezeigt werden.

**gegnerLoeschen():** Löscht alle Gegner (Klasse: attacking, Boot)

**settingsErstellen():** Erstellt die Oberfläche für die Einstellungen.

**namesMenueErstellen():** Erstellt die Oberfläche für die Namenswahl.

profilErstellen():Erstellt die Oberfläche für das Profil.

**newShot(int, int, int, Actor):** Erstellt einen neuen Schuss an einer bestimmten Stelle, der von einem bestimmten Actor in eine bestimmte Richtung abgefeuert wurde.

**tasteErstellen(int , int, String):** Erstellt einen neue Taste an einer bestimmten Stelle und legt die Funktion der Taste fest.

**schriftErstellen(int, int, String):** Erstellt einen neues Feld an einer bestimmten Stelle und legt fest, was für ein Text angezeigt werden soll.

**levelErstellen():** Erstellt, gemäß dem ausgewählten Level Boote (Klasse: Boot), Flugzeuge (Klasse: attacking) und PowerUps (Klasse: PowerUps).

**Wichtige Attribute sind:**

**warnung:** Boolean, der bestimmt, ob die Warnung, dass der Spieler zu nah am Rand des Spielfeldes ist nagezeigt wird oder nicht.