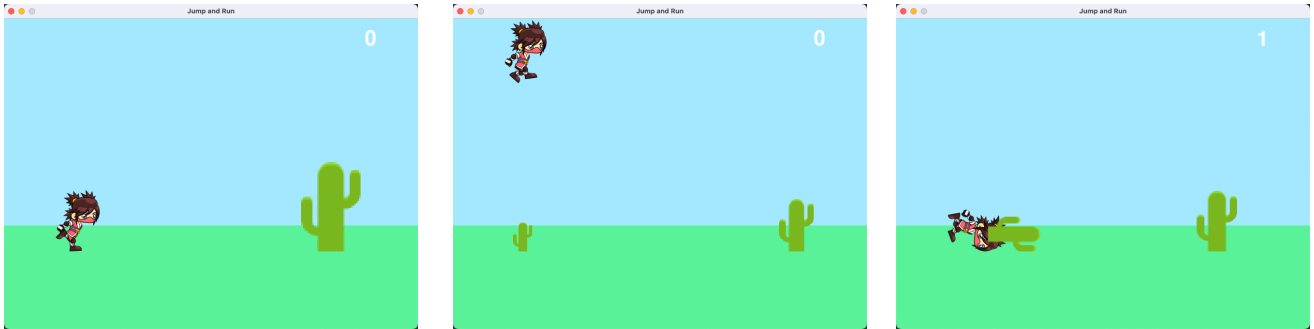


Jump and Run

Diese Game stammt aus einem früheren Video. Ich habe es damals in ähnlicher Form mit Pygame Zero realisiert.



Anforderungen

- Eine Musik wird im Hintergrund abgespielt.
- Ein Hintergrund mit Boden und Himmel stellt das Spielfeld dar.
- Rechts oben befindet sich der Punktestand.
- Der Spieler befindet sich immer an derselben x-Position.
- Der Spieler ist animiert.
- Mit der Leertaste kann der Spieler springen. Sobald er in der Luft ist, bewegen sich die Beine nicht mehr.
- Er kehrt automatisch auf den Boden zurück.
- Durch fortlaufende Betätigung der Leertaste kann er sich länger in der Luft halten. Dabei bewegen sich die Beine.
- Von rechts her laufen in wechselnden Abständen verschieden grosse Kakteen durch das Bild.
- Das Spiel soll immer etwas schneller werden. Dabei wird die Animation und die Geschwindigkeit der Kakteen erhöht.
- Der Spieler muss sie überspringen. Wenn er einen Kaktus berührt, fallen Spieler und Kaktus um und das Spiel ist beendet. Dabei wird ein Sound abgespielt.
- Wenn ein Kaktus das Spielfeld links verlässt, verschwindet er und der Spieler erhält einen Punkt.
- Ein beendetes Spiel kann mit der Return - Taste neu gestartet werden.

Media - Dateien

https://www.aposteriori.com.sg/wp-content/uploads/2020/02/image_pack.zip

<https://opengameart.org/content/nas-shooter-music-5-tracks-3-jingles>

<https://opengameart.org/content/game-over-soundold-school>

run_000.png .. run_009.png: der Spieler (Animation)

cactus.png: der Kaktus

gameover.wav: das Spiel ist beendet

venus.wav: die Hintergrundmusik

Vorgehen

- **AUFGABE:** Erstelle das Spielfeld, den Hintergrund und einen Punktezähler.
- Das Spielfeld hat eine Grösse von 800 x 600. Die Grenze zwischen Himmel und Boden liegt bei $y = 400$.
- Die Farbe des Himmels ist (163, 232, 254) und die des Bodens (88, 242, 152).
- Das Spiel hat eine Startgeschwindigkeit von 150.
- Das Spiel ist am Anfang noch nicht aktiv. Dieser Zustand wird in einer Eigenschaft **active** gespeichert.
- In der Initialisierung haben wir noch keinen Spieler. Dieser wird erst beim Spielstart mit Hilfe einer Funktion **neuer_spieler** erstellt.
- **neuer_spieler** wird in der Funktion **neues_spiel** aufgerufen.
- Der Spieler befindet sich immer an der Position $x = 100$.
- Die Fusslinie des Spielers und aller anderen Spielobjekte befinden sich 50 Pixel unterhalb der Grenze zwischen Himmel und Boden.
- Die Spielerklasse und ein Spieler können jetzt erstellt werden.
- Die Animation des Spielers soll aktiviert werden.
- Ein Kaktus ist zwischen 50 und 300 Pixel hoch.
- Er wird rechts erzeugt und bewegt sich nach links. Die Geschwindigkeit wird durch **game.speed** festgelegt.
- Sobald der Kaktus am linken Rand verschwindet, wird er zerstört und rechts ein neuer Kaktus erzeugt. Dabei soll ein Punkt hochgezählt werden.
- Wir erstellen eine Klasse **Kaktus**.
- Wir erzeugen eine Gruppe **kaktus_gruppe**.
- Den ersten Kaktus erstellen wir in **neues_spiel** mit Hilfe einer Funktion **neuer_kaktus**.
- Wenn die **kaktus_gruppe** leer ist, erzeugen wir einen neuen Kaktus.
- Die Geschwindigkeit des Spiels soll mit den Tasten + und - in Schritten von 20 verändert werden.
- Alle Kakteen sollen sich dieser Geschwindigkeit anpassen.
- Die Laufgeschwindigkeit des Spielers soll sich ebenfalls anpassen.
- Der Spieler kann springen, das bedeutet eine Bewegung in y - Richtung. Diese ist aber nach oben und unten begrenzt. Das können wir mit Constraints festlegen.
- Wir platzieren für einen ersten Test den Spieler in einer Höhe von $y = 200$.
- Jetzt läuft er in konstanter Höhe in der Luft. Er soll aber durch die Schwerkraft nach unten gezogen werden.
- Die Pygame - Tools unterstützen das. Wir müssen dem Spieler nur eine Masse geben (self.mass)
- Der Spieler läuft jetzt wieder am Boden.
- Mit der Leertaste leiten wir einen Sprung ein. Dabei wird ein Impuls gegeben (impulse = 350).
- Wenn ein Sprung gestartet wird, stoppt die Animation des Spielers
- Wenn bereits ein Sprung im Gang ist, geben wir nur einen Impuls von 25 und die Animation läuft wieder für 0.5 Sekunden.
- Die Kakteen sollen nicht mehr regelmässig erzeugt werden.
- Es sollen in unregelmässigen Abständen neue Kakteen erzeugt werden. Die Abstände liegen zwischen 300 und 800 Pixeln.
- Das Spiel soll alle 3 Sekunden etwas schneller werden (+ 20).
- Eine Kollision zwischen Spieler und Kaktus muss erkannt werden. Spieler und Kaktus fallen dabei um und das Spiel ist beendet.
- **AUFGABE:** Mit Return kann ein neues Spiel gestartet werden. Ausserdem soll das Spiel mit Hintergrundmusik und einem Kollisionsgeräusch ausgestattet werden.