Game Design Document Flower Soldier

Core Gameplay

Das Spiel ist ein 2D-Platformer, in dem man durch ein Level rennt über Gegenstände springt und Hindernissen ausweicht. Erleichtert wird dies durch spezielle Fähigkeiten.

Story

Man spielt den „Super Space Gardener", der ein von der Nasa erschaffener Supergärtner der Liebe ist. Jedoch hat er herausgefunden, dass die Liebe eigentlich ein Fehler in seinem Programm ist und seine Erschaffer ihn eigentlich für den Krieg erschaffen haben, deswegen möchte er seine Schöpfer, die eine furchtbare Pollenallergie haben, mit seiner Superblume einstauben in der Nasa ganz viel Liebe verbreiten. Die Nasa versucht das jedoch zu verhindern, indem sie gefühlslose, pflanzenhassende Steine erschafft und auf dem Weg des Gärtners platziert, die ihm seine Pflanzenkraft rauben wollen und seine allergische Reaktion umprogrammieren, wodurch er einen schrecklichen Hustenanfall bekommt und von neu anfangen muss.

Charaktere

Super Space Gardener

Der „Super Space Gardener", kurz SSG kann nach links und rechts laufen und sprinten. Er ist ebenfalls in der Lage zu springen und sich zu ducken, um so Hindernissen und Feinden auszuweichen. Ebenfalls ist er in der Lage an Wänden zu rutschen und an diesen hochzuspringen.

Seine Spezialfähigkeit ist, dass er einmal in der Luft die Zeit verlangsamen kann, was dem Spieler Zeit zum Nachdenken und verschiedene Möglichkeiten für Manöver bereitet. In der Zeitlupe hat der Spieler Zeit, mit der Maus/Joystick einen „Vektor" zu zeichnen, in dessen Richtung er dann „dasht". Hierbei ist er unverwundbar und kann werder Schaden erleiden, noch getötet werden.

Die Steine

Die Steine wurden von der Nasa erschaffen, mit dem Ziel, den SSG aufzuhalten und ihn kampfunfähig machen. Dies versuchen sie durch eine spezielle Fähigkeit, die ihnen die Nasa gegeben hat. Sie sind nähmlich in der Lage den SSG umzuprogrammieren und durch seine Allergische Reaktion zum Neuanfang zu bringen.

Die NASA

Die Nasa tritt nie explizit als einzelne Person auf, sondern nur als Kollektiv. Sie spiegelt die gesamte Grausamkeit der US-Regierung wieder, in dessen Auftrag der SSG erschaffen wurde.

Sie ist auch sein Hauptziel, da er hofft, von hier seine Liebe in der ganzen USA zu verteilen. Des weiteren sollen im verlaufe des Spiels mehrere Verschwöhrungstheorien von der Nasa ausversehen aufgedeckt und ins Lächerliche gestellt werden.

Level/environment design

Die Level bestehen aus einer Tilemap, welche durch einzelne Objekte, wie zum Beispiel die bösen Steine oder Jumppads erweitert wird. Die Jumppads dienen insbesondere zur Beförderung in neue höhen, um die Level abwechslungsreicher zu Gestalten geplant sind 5 Welten, mit jeweils 10 Leveln, einem Secretlevel, sowie einem Bosskampf. In diesem ist es das Ziel, den Boss durch eine Schwachstelle dazu zu bringen, auf die Seite der Liebe zu wechseln und sich dem SSG anzuschließen. Die ersten 4 Welten sollen thematisch verschiedene Regionen, wie Wald, Höhle, Eis und Wüste beinhalten. Die finale Welt soll allerdings in Houston bei der Nasa spielen. Nach dem finalen Bosskampf gegen den Erschaffer des SSGs(Name und Informationen noch nicht bekannt) und dem Abschließen aller Secretlevel schaltet man zusätzlich eine geheime Bonuswelt frei, welche aus 5 Leveln besteht. Ebenfalls schaltet man den Hard-Mode für alle bisherigen Level und Bosskämpfe frei, welche erschwert und mit einem Zeitlimit ausgestattet wurden.

Gameplay

Das Gameplay soll hektisch durch schwierige Jump-Passagen und Hindernisse sein, gleichzeitig aber auch ruhig sein, wenn man gerade die Zeit beim dashen verlangsamt, was dem hektischen Gameplay einige Nachdenkzeit für die nächsten Spielzüge geben soll.

Art

Der Artstyle des Games soll ein Pixel-Artstyle sein, da dieser sehr einfach um zeichnen ist und trotzdem gut aussieht. Orientiert wird sich hierbei am Artstyle von Spielen wie „Celeste" oder „Super Meat Boy".

Sound und Musik

Die Musik des Spiels soll vollständig selbst geschrieben werden. Hierfür werden Stücke mit Gitarre, Schlagzeug und Slap BASS geschrieben. Die Sounds sollen mit dem frei im Internet verfügbahren Sound-Generator generiert werden.

UI

Die UI soll möglichst schlicht gehalten werden, wie es in anderen Plattformern meist auch der Fall ist, damit der Fokus auf dem Gameplay liegt. Es soll höchstens einen Zähler für Collectibles geben.

Accessibility

Das Spiel soll unter Linux, OsX und Windows veröffentlicht werden. Ebenfalls geplant ist eine Version für die Wii, die mit dem Wii-Fit Balanceboard gesteuert werden kann. Der PC-Port hingegen soll mit der Tastatur oder optional mit einem Controller gesteuert werden. Bei Erfolg

könnte ebenfalls ein Konsolenport für Nintendo-Switch, Xbox und Playstation folgen.

Technische Umsetzung

Das Spiel soll in der Godot-Engine entwickelt werden. Die Map wird durch ein Tileset konstruiert. Hierdurch wird ebenfalls die Kollision mit der Map festgestellt. Der Player bekommt seine verschiedenen Animationen durch ein Animated-Sprite, wodurch es einfach ist, die Animationen zu wechseln.

Zum Feststellen und Ändern der Zustände des Players wird eine selbst Programmierte State-Machine benutzt, welche die State-Detection und die State-Execution trennt. Die State Detection ist hirbei das Feststellen des aktuellen Zustandes des Spielers durch verschiedene Variablen, zum Beispiel ob sich der Spieler auf dem Boden befindet. Die State-Execution ist das ausführen bestimmter Operationen bei gewissen Zuständen. Ein Beispiel hierfür wäre das Ändern von Player-Animation und Player-Hitbox beim Ducken.

Das Ducken, sowie andere Operationen die Kollisionen erkennen müssen, werden entweder durch Raycasts umgesetzt, dessen Kollision einfach in Godot durch eine if-Abfrage erkannt werden kann, oder durch Areas, welche Signale senden können, die durch Funktionen empfangen werden, wenn diese kollidieren. Das gesame Movement des Players wird über die Funktion move_and_slide umgesetzt, die den Spieler zu einer gegebenen Position bewegt.

Verschiedene Szenen, wie zum Beispiel nach dem Abschließen eines Levels werden durch andere Szenen umgesetzt, zu welchen in bestimmten Situationen gewechselt wird.

Die Umsetzung des Dashens steht noch nicht fest, allerdings gibt es bereits eine Idee, dessen Funktionalität nochnicht erprobt wurde. So soll für die Zeitlupe eine globale Variable zuständig sein, die bei Aktivierung dieser verringert wird und dadurch alle von der Zeit abhängigen Operationen verlangsamt. Der Dashpunkt soll über eine Funktion festgestellt werden, die die Position des Mauszeigers zurückgibt. Daraufhin soll mithilfe von move_and_slide der Spieler zu dieser Position hinbewegt werden.

Stand der Entwicklung

Aktuell sind die Animationen für Player, Steine und JumpPad vorhanden, sowie eine Übergangstilemap, welche teils aus eigenen und teils aus frei im Internet verfügbaren Blöcken besteht. Es wurden ebenfalls ein Deathscreen, sowie ein Winscreen gezeichnet. Das Hintergrundbild für das erste Level konnte leider nicht mehr eingebunden werden, da dieses auf dem Schulserver liegt und wir keine Zeit mehr hatten, darauf zuzugreifen.

Für den Sound wurde auf dem Bass ein Musikstück aufgenommen, sowie ein Hustsound aufgenommen (Hier hat Emil beste Arbeit geleistet).

Die Physik des Spiels funktioniert weitesgehend, so funktionieren laufen, springen, fallen, ducken und sprinten einwandfrei allerdings gibt es beim laufen im ducken, wallsliding, sowie beim Springen auf dem JumpPad noch Probleme. Beim Laufen im Ducken kann man nämlich nur einen Block geduckt laufen. Das Wallsliding macht öfters Probleme, so fällt man mitten im WallSlide herunter und kann nicht mehr an der Wand springen und beim JumpPad gibt es nicht

lösbare Kollisionsprobleme, so kann man seitlich einfach durch das JumpPad durchlaufen.

Allerdings konnten auch viele für uns unlösbare Probleme behoben werden, so schaften es wir nach viel Aufwand einen Bug lösen, bei dem man im geduckt laufen aufstehen konnte und im Tileset stecken blieb. Ebenfalls schafften wir es noch einen Todesscreen und einen Winscreen einzubauen auf deren implementierung wir sehr stolz waren.

Letzendlich steht unser Projekt vielleicht gerade mal bei 10 Prozent, trotzdem haben wir in der Zeit es geschafft eine gute Physik-Engine einzubauen, sowie eine funktionierende State-Machine selbst zu programmieren. Mit dem Leveldesign stehen wir zwar noch am Anfang, allerdings haben wir jetzt ja auch 5 Wochen keine richtige Schule in denen man viel weiterarbeiten kann.