Inhalt

[1. Einleitung 2](#_Toc527571987)

[2. Spielablauf und -mechanik 3](#_Toc527571988)

[2.1 Spielidee 3](#_Toc527571989)

[2.2 Spielmechanik 3](#_Toc527571990)

[2.3 Spielbalancing 3](#_Toc527571991)

# 1. Einleitung

Bei dem Softwareprojekt „Time Raider“ handelt es sich um ein 2D-Plattformer Spiel, welches von vier Studenten der Hochschule Bochum entwickelt wurde. Wir haben uns für ein 2D-Spiel entschieden, da wir so die Bereiche Programmierung und Design verknüpfen und die Entwicklung abwechslungsreich gestalten können.

Das Spiel wird vorerst für Windows-PCs entwickelt. Mit Unity sind wir allerdings in der Lage „Time Raider“ kompatibel für weitere Betriebssysteme und Plattformen, wie beispielsweise einer mobilen Applikation, zu machen.

# 2. Spielablauf und -mechanik

# 2.1 Spielidee

Die Hauptfigur ist aufgrund einer Zeitreise ungewollt in die Vergangenheit gereist und muss wieder in die Gegenwart zurückreisen, allerdings wurde die Zeitmaschine zerstört. Ab hier übernimmt der Spieler die Kontrolle der Hauptfigur und bewegt sich durch verschiedene Orte aus unterschiedlichen Zeiten. Jeder der vier Akte spielt sich jeweils in einer Zeit ab. In jedem Akt beziehungsweise Zeit, gibt es jeweils eine verschiedene Anzahl an Level (Orte). Damit die Hauptfigur wieder in die Gegenwart gelangt, muss der Spieler verschiedene Aufgaben bewältigen. Gegner und Hindernisse aus den jeweiligen Zeiten versuchen den Spieler davon abzuhalten, die Aufgaben zu erledigen. Der Spieler muss Gegner und Hindernisse bewältigen, um Spielfortschritte zu erlangen.

# 2.2 Spielmechanik

Die Hauptfigur bewegt sich durch verschiedene 2D-Welten. Dieser kann nach links und rechts laufen beziehungsweise kriechen. Der Spieler startet mit 100 Lebenspunkten. Wird der der Spieler von Hindernissen oder einem Gegner getroffen, verliert er Lebenspunkte. Sobald der Spieler keine Lebenspunkte mehr besitzt, hat er das Level verloren und muss von Beginn des Levels anfangen. In jedem Level befinden sich Herzen, die der Spieler aufheben kann, um einen Lebenspunkt zu erhalten. Um Hindernisse bewältigen zu können, kann die Hauptfigur springen. Einige Hindernisse sind zum Beispiel Schluchten. Fällt man in diese hinein, kann man nicht mehr hinauskommen. Der Spieler verliert in diesem Fall Lebenspunkte und wird zum letzten sicheren Checkpoint befördert, den er zuvor erreicht hatte. Gegner können mit Fern- und Nahkampfwaffen angreifen und vom Spieler angegriffen werden. Mit dem Kaufsystem kann der Spieler sich mit Gegenständen ausrüsten, um sich stärker werdenden Gegnern zu stellen (siehe 2.3 Spielbalancing).

# 2.3 Spielbalancing

Im Laufe des Spiels werden Hindernisse schwieriger zu absolvieren sein. Der Spieler kann sich zur Bewältigung der Hindernisse keine Vorteile erschaffen. Hier zeigt sich das Können des Spielers. Die Gegner hingegen werden stärker, indem ihre Lebenspunkte erhöht werden. Damit der Spieler die Möglichkeit hat, die Gegner zu besiegen, sollte er sich für den jeweiligen Akt ausrüsten. Dabei muss er sich das Kaufsystem zu Nutzen machen und passende Gegenstände kaufen, die ihm einen Vorteil verschaffen. Eine wichtige Rolle spielen zeitgemäße Fern- und Nahkampfwaffen, damit die Hauptfigur bei stärkeren Gegnern, mehr Schaden anrichten kann.

# 3. Software

# 3.1 Unity

Bei der Wahl der Laufzeit- und Entwicklungsumgebung von „Time Raider“ haben wir Unity vor Unreal Engine bevorzugt, da Unity sehr einsteigerfreundlich ist und viele Lernmaterialien, sowie Beispielspiele bereitstellt. Außerdem ist die Community sehr aktiv und bietet unzählige Tutorials an. Unity unterstützt C# und JavaScript. Mit nur wenig Arbeit ist es möglich, Spiele für verschiedene Plattformen zu generieren.

# 3.2 Inkscape

Inkscape ist eine Software zur Erstellung und Bearbeitung von zweidimensionalen Grafiken, welche vektorbasiert arbeitet. Dies hat den Vorteil, dass wir Grafiken ohne Detailverlust vergrößern oder verkleinern können.

# 3.3 GitHub

GitHub wird zur Versionsverwaltung genutzt. Mit wenigen Klicks sind wir in der Lage unsere Neuerungen und Veränderungen untereinander auszutauschen. Zuvor haben wir „Unity Teams“ genutzt, welches in der Unity Engine eingebunden ist. Diese wurde allerdings im Laufe unseres Projekts kostenpflichtig.