27.04.

-Erster Versuch Grafik darzustellen, Spieler-Objekt gibt größe, position und Farbe an GUI Objekt. Das Spiel wird aktiv gerendert, der Code dafür kommt von der Seite <https://www.gamedev.net/tutorials/_/technical/general-programming/java-games-active-rendering-r2418/>

<https://stackoverflow.com/questions/2442599/how-to-set-jframe-to-appear-centered-regardless-of-monitor-resolution> hat bei der zentrierung des Fensters geholfen

-Tasteneingabe in Bewegung umgesetzt, Spieler kann mit wasd vertikal und horizontal sowie diagonal bewegt werden. Syntax für Keylistener ebenfalls von <https://www.gamedev.net/tutorials/_/technical/general-programming/java-games-active-rendering-r2418/>

28.04.

Input von Keylistener zu KeyBinds geändert

<https://stackoverflow.com/questions/22741215/how-to-use-key-bindings-instead-of-key-listeners>

Diagonale Bewegunsgeschwindidkeit auf geradlinige Bewegungsgeschwindigkeit reduziert, position als Point2D gespeichert

29.4. und 30.4

-Kollisionssystem erstellt. Kollison von Rechtecken kann ermittelt werdn. Wand Klasse erstellt und Kollision mit Wand ermöglicht. Der bildschirmrand kann ebenfalls nicht mehr verlassen werden.

-Vollbildmodus <https://stackoverflow.com/questions/1155838/how-can-i-do-full-screen-in-java-on-osx> kann mit escape beendet werden

03.05.

-Mithilfe von Zeichnung die benötigte If-Abfrage überlegt um die Kollision von Kreisen mit Rechtecken zu überprüfen.

-kreis Rechteck Kollision in Code gebracht und Projektil Klasse erstellt

04.05.

-Kollisioon von Spieler und Projektil implementiert sowie funktion um Projektile malenzu können gemacht. MAxLeben und Leben als Eigenschaften des Spielers hinzugefügt

-Methode um das Fenster zu schließen ohne Fehlermeldung von <https://stackoverflow.com/questions/1234912/how-to-programmatically-close-a-jframe>

05.05.

-Um Projektil zu Punkt zu bewegen erst fehlerhafte Formel benutzt, dann statt Steigung eine Berechnung über winkel von <https://stackoverflow.com/questions/39818833/moving-an-object-from-one-point-to-another> genommen.

-Bei Mauscklick wird ein Projektil in Richtung Mausklick geschossen, wenn Projektile den Bildschirm verlassen werden sie gelöscht. Kontakt mit Wand-Objekten löscht das Projektil ebenfalls. Feuern nur auf linke Maustaste beschränkt.

10.05.

- Update und Start Klasse erstellt um gleichmäßige Updates und Multi-Threading zu ermöglichen. Gleichmäßige Updates von <https://stackoverflow.com/questions/63515194/how-to-run-a-code-60-times-per-second-in-java> Thread starten mit Hilfe von [https://dbs.cs.uni-duesseldorf.de/lehre/docs/java/javabuch/html/k100142.html#:~:text=Die%20Klasse%20Thread%20ist%20Bestandteil,Beenden%20von%20Threads%20zur%20Verf%C3%BCgung.](https://dbs.cs.uni-duesseldorf.de/lehre/docs/java/javabuch/html/k100142.html%23:~:text=Die%20Klasse%20Thread%20ist%20Bestandteil,Beenden%20von%20Threads%20zur%20Verf%C3%BCgung.)

17.05.2021

- Lösung für unterschiedliche Bildschirmauflösungen gefunden, Windows Einstellungen ignoriert mithilfe von <https://stackoverflow.com/questions/47613006/how-to-disable-scaling-the-ui-on-windows-for-java-9-applications>

19.05.2021

-Waffenklasse erstellt, subKlassen für verschiedene Waffentypen erstellt. Aktuell ausgerüstete Waffe wird auf dem Bildschirm angezeigt

24.05. und 25.05.

-Waffenpickup erstellt um Waffen im Spiel aufzusammeln. Die Shotgun schießt jetzt mehrere Kugeln und einer Kegelform, Projektile haben eine maximale Distanz die sie sich bewegen. Die Waffenpickups verteilen bei Berührung vorher zufällig erstellte Waffen und werden anschließend gelöscht.

-Magazingröße und übrige Kugeln werden gezählt und angezeigt. Wenn das Magazin leer ist wird ein Timer gestarten nach dem das Magazin wieder voll ist und man wieder schießen kann. Syntaxhilfe von <https://docs.oracle.com/javase/7/docs/api/java/util/Timer.html> und <https://docs.oracle.com/javase/7/docs/api/java/util/TimerTask.html>

-Nachladen wird durch Text angezeigt