Spielziel:
Zerstöre alle Steine mit einem Ball. Du musst den von oben kommenden Ball mit deinem Schläger vor dem Absturz bewahren. Sind alle Steine zerstört, ist dass Level geschafft.
Es sind 10 unterschiedliche Level vorhanden. Je nach Level sind die Steine anderes angeordnet.
Ein Stein kann bis zu drei Leben haben. Erst nach dem ein Stein kein Leben mehr besitzt, wird er zerstört. Trifft der Ball einen Stein, kann es passieren das ein Bonus oder Malus zu Boden fällt, trifft der Spieler diesen mit dem Schläger, wird dieser ausgelöst. Folgende Bonus und Malus sind vorhanden, zusätzlicher Ball, vergrößern der Bälle im Spiel, verkleinern der Bälle im Spiel, schnellere Bälle im Spiel, langsamere Bälle im Spiel, vergrößern des Schlägers, und verkleinern des Schlägers.
Der Spieler verliert das Spiel, sobald er keine Leben mehr hat. Der Spieler verliert ein Leben, sobald keine Bälle mehr im Spiel sind. Bei Start des Spiels besitzt der Spieler drei Leben, für jedes gemeisterte Level bekommt der Spieler ein Leben dazu. Erreicht der Spieler Level 10, wird dieses endlos wiederholt, Ziel ist es, einen möglichst hohen Punktestand zu erreichen.

Steuerung:
Schläger nach rechts bewegen: Taste D oder Pfeil nach rechts
Schläger nach links bewegen: Taste A oder Pfeil nach links
Schläger nach oben bewegen: Taste W oder Pfeil nach oben
Schläger nach unten bewegen: Taste D oder Pfeil nach unten
Alternativ kann der Schläger mit der Maus bewegt werden.

main.py
Die main.py ist der Einstiegspunkt in das Programm. In der Main Methode wird das Hauptmenü des Spieles gestartet.

screen.game_menu.py
Das game_menu stellt das Hauptmenü des Spieles da, im Hauptmenü sind 3 Buttons enthalten, Spielen, Anleitung und Beenden. Klickt der Benutzer auf einen der Buttons, wird das jeweilige Fenster geöffnet.

screen.game_manual.py
In diesem Fenster wird die Spielanleitung dargestellt. Das Fenster enthält einen Button "zurück zum Hauptmenü".

screen.game_over.py
Sobald der Spieler kein Leben mehr hat, wird, diese Fenster angezeigt. Hier sieht der Spieler sein erreichtes Level und seine erreichten Punkte. Über einen Button hat der Spieler die Möglichkeit, das Spiel erneut zu spielen.

game_loop.py
Beinhaltet die Spielschleife. Zuerst werden die Spielerfigur und ein Ball erstellt. Diese Objekte werden zu den Sprites hinzugefügt. In der Methode game_functions.check_events(player) wird überprüft, ob eine Taste gedrückt wurde oder die Maus bewegt wurde.

Ein neues Level wird erstellt, wenn keine Steine mehr im Spiel sind. Passiert dieses, wird der Levelzähler um eins erhöht. Sobald das Level zehn erreicht wird, wird der Zähler nicht mehr erhöht. Die Steine des Levels werden zu den Sprites hinzugefügt. Eine if-Anweisung überprüft, ob noch Bälle im Spiel vorhanden sind, ist kein Ball mehr vorhanden, wird das Leben um eins verringert und ein neuer Ball erstellt. Anschließen folgen mehrere If-Anweisung , in diesen wird überprüft, ob es zu Kollisionen zwischen den einzelnen Objekten gekommen ist.
Sobald eine Kollision stattgefunden hat, wird die zugehörige Aktion ausgelöst. Anschließend wird das aktuelle Leben, Punkte und Level angezeigt.

game_functions.py
Enthält unterschiedliche Spielfunktionen.
Die Methode check_events
überprüft, ob eine Taste oder die Maus bewegt wurde. Wurde die Taste W,A,S,D oder eine der Pfeiltasten gedrückt, wird die Spielerfigur bewegt.
Mithilfe der Methode check_level, wird ein neues Level erstellt. Dazu wird mithilfe einer if-Anweisung das Level ausgewählt. In der Klasse Level wird anschließend das Level erstellt und die benötigten Steine zurückgeben. Diese Steine werden zu den Sprites hinzugefügt.
Sobald eine Kollision zwischen dem Spieler und einem Ball stattgefunden hat, wird die Methode collide_player_vs_balls ausgelöst. Der Ball ändert die Richtung.
Die Methode collide_balls_vs_stones, sorgt dafür, dass der Ball ebenfalls die Richtung ändert.
In der Methode collide_stones_vs_balls werden die unterschiedlichen Bonus oder Malus erstellt, hierzu wird ein Zufallsgenerator genutzt.
Sobald es zu einer Kollision zwischen dem Spieler und einen Bonus oder Malus gekommen ist, wird die Methode collide_player_vs_bonus_or_malus ausgeführt. Je nachdem um welchen Bonus oder Malus es sich handelt, wird die entsprechende Funktion gewählt und ausgeführt.
Die Methode check_bonus_or_malus_null
erstellt eine Zufallszahl zwischen drei und acht.
Die Methode game_over öffnet ein neues Fenster, es werden der Punktestand und das aktuelle Level übergeben.

player.py
Die Klasse stellt die Spielerfigur da. Die Klasse enthält Methoden zur Steuerung des Spielers (nach rechts bewegen, nach links bewegen, nach unten bewegen, nach oben bewegen). Außerdem Methoden die, die aktuelle Position (links, rechts, Mitte) zurückgeben. Zwei Methoden ändern die Spielergröße. Die Spielergröße werden durch einen Bonus oder Malus verändert.

stones.py
Stellt einen einzelnen Spielstein dar. Enthält Methoden die, die Position zurückgeben und das Leben des Steines senken.

ball.py
Diese Klasse erstellt einen Ball. Die Methoden bounce_stone
und bounce_player ändern die Flugrichtung. Sobald es zu einer Kollision zwischen dem Spieler und dem Ball kommt, wird überprüft, wo genau der Spieler den Ball getroffen hat.

Je näher am Zentrum des Schlägers, desto kleiner ist der Winkel des abprallenden Balles. Wird der Ball mit der rechten Seite, des Schlägers getroffen fliegt er nach rechts, wird er links getroffen, fliegt der Ball nach links. Außerdem sind Methoden vorhanden die den Ball, vergrößern, verkleiner, verlangsamen oder schneller machen.

level.py
Diese Klasse enthält die Information, wo und wie, welcher Stein in dem jeweiligen Level angeordnet werden soll.

bonusOrMalus.py
In dieser Klasse werden die Bonus oder Malus erstellt.