Gem Catcher

Game Design Document

Brian Buckles

# Overview

You will play as a miner that tries to catch as many different gems as possible while avoiding falling rocks. As the time progresses more rocks will fall making the difficulty harder. The miner will have 3 lives and if the miner hits a rock it will lose 1 life. The game will end when all the lives have been spent.

It will use the pygame and simpeGE game engine to create the scenes and sprites. The scene will be a background of a cave. The sprites will consist of different gems, different rocks, and a miner. There will be 2 scenes. Scene 1 will be the main menu showing you a play button and an exit button. When you press the play button it will take you to scene 2 which is the game. When the game is over the user will be taken back to the main menu where they can play again or leave the game.

# State Transition Diagram

Main Menu

Game

# Scenes

## Main Menu Scene

# Main Menu

**Gem Miner**

High Score

Instructions:

Catch the gems while avoiding the rocks.

You can only use the left and right arrows.

Play

Exit

**Elements:**

* **lblInstructions** – instructions to play the game
* **lblHighScore** – shows the high score
* **btnPlay** – button that will take you to the game scene
* **btnExit** – button that will close the application

Add 1 to score if collides with user

## Game Scene

Lives Left

Spawn below this line

Score

Miner

Rock take life if collides with user

When rock or gem hits the bottom reset to random x at the line above

**Elements**:

* **spMiner** – miner sprite
* **spRock** – rock sprite
* **spGem** – gem sprite
* **lblScore** – score displayed
* **lblLives** – lives left before game over

**Variables**:

* **lives** – holds the lives left before game over
* **score** – holds the number of gems collided with

# Classes

## MainMenu

Main menu for the game. It will show the best score, instructions, and controls the actions of the game. Allows you to exit and play the game.

### Variables

* \_\_command – holds the button press command defaults to play
* \_\_high\_score – holds the highest game score since launch.

### Methods

* \_\_init\_\_ - initialization of the MainMenu class.
  + Sets the high score
  + Sets the default command
* process
  + check if btn\_quit is clicked
    - set \_\_command to quit
    - stop
  + check if btn\_play.clicked
    - set \_\_command to play
    - stop
* get\_command
  + returns \_\_command to caller
* set\_high\_score
  + sets the \_\_high\_score

## Game

Holds the game mechanics.

### Variables

* \_\_score – holds the current score. Defaults to 0.
* \_\_lives\_left – holds the current lives left before the game is over. Defaults to 3
* \_\_total\_gems – holds the number of gem sprites to create on the screen at one time.
* \_\_total\_rocks – holds the number of rocks sprites to create on the screen at one time.
* gems – holds the gem sprites created
* rocks – holds the rock sprites created
* lives – holds the life sprites

### Methods

* \_\_init\_\_ - initialization of the Game class.
  + Sets the score to 0 and lives to 3.
  + resets gem
* process
  + Check if miner collides with gem
    - play sound assets/bing1.wav
    - adds one to the score
    - reset gem sprite
  + Check if miner collides with rock
    - play sound assets/thud2.wav
    - substract 1 life from \_\_lives\_left
    - reset rock sprite
  + Check if there are more than 0 lives left. If there are 0 lives left return to main menu scene
* getScore – returns the \_\_score

## Miner

Miner sprite that can move left and right and collides with either a rock or a gem. Inherits from simpleGE.Sprite class.

### Variables

* moveSpeed – controls how fast the sprite moves horizontally.

### Methods

* \_\_init\_\_ - sets the moveSpeed to 5 and starts bottom middle of the screen.
* process – checks if left or right is pressed and moves that direction.

## Gem - Mineral

Gem sprite that moves vertically from top to bottom. Inherits from simpleGE.Sprite class.

### Assets

**Image** – assets/gem.png

## Rock - Mineral

Rock sprite that moves vertically from top to bottom. Inherits from simpleGE.Sprite class.

### Assets

**Image** – assets/gem.png

### Variables

* moveSpeed – controls how fast the sprite moves from top to bottom.

## Mineral

### Methods

* \_\_init\_\_ - sets size to 50 x 50
* checkbounds – checks bottom of screen and if it is resets it
* reset – sets the mineral to the to random horizontal value with a random vertical speed.

## Main()

Entry point into the game.

### Variables

- **bestScore** – stores the best score. Default 0

- **keepGoing** – determines if the loop keeps looping. Default true

- **game** – holds the game instance.

- **mainMenu** – holds the mainMenu instance

- **instructions** – holds the game instructions to be displayed.

# Milestone

General strategy is to create gameplay first, then instructions screen, and finally integrate with state management. Game process will be stored on github, with a marked branch for each milestone reached and multiple other commits as needed. Each milestone branch will run correctly with the milestone demonstrated. Each milestone is expected to take one programming session to complete.

1. Game scene with background image
2. Add basic Miner sprite
3. Add keyboard motion to Miner
4. Add single Gem with reset, falling and boundary behaviors
5. Add single Rock with reset, falling and boundary behaviors
6. Add collision effect between Miner and Gem, sound effect
7. Add collision effect between Miner and Rock, sound effect
8. Modify for multiple (ten) Gems including collision behavior
9. Modify for multiple (ten) Rocks including collision behavior
10. Add scorekeeping, timing, and appropriate labels
11. Add instructions class and state transition

# /assets

## Images

gem.png - <https://opengameart.org/content/batu-pack-1a128-gem-stone>

rock.png - <https://opengameart.org/content/batu-pack-1a128-gem-stone>

miner.png - <https://pixlr.com/express/>

mine.png - <https://pixlr.com/express/>

### Fonts

assets/Ranchers-Regular.ttf - <https://www.fonts4free.net/ranchers-font.html>

### Sounds

bing1.wav - <https://opengameart.org/content/metal-impact-sounds>

bong1.wav - <https://opengameart.org/content/metal-impact-sounds>

clink1.wav - <https://opengameart.org/content/metal-impact-sounds>

clink2.wav - <https://opengameart.org/content/metal-impact-sounds>

clink3.wav - <https://opengameart.org/content/metal-impact-sounds>

thud2.wav - <https://opengameart.org/content/metal-impact-sounds>

thud3.wav - <https://opengameart.org/content/metal-impact-sounds>