Breaking Red

Developers Manual

A logo with text on it

AI-generated content may be incorrect.

1. Overview

Breaking Red is a 2D top-down style platformer game in which the player plays Red, a girl who must embark on a journey solve the mystery of who murdered her grandmother, with only her trusty cane by her side. The player will travel through 5 different levels where they can talk to/attack NPCs for clues and collect power-ups. Their goal is to discover the true murderer of Grandma without running out of health or sanity.

1. Development Platform

The Green Sky Games development team used Unity version 6000.0.38f1 to create Breaking Red. Unity and Unity Editor must be installed in order to download the game.

**Installing Unity and Unity Editor**

* Go to <https://unity.com/download> and scroll down until you see 1. Download the Unity Hub
* Click Download for Windows and open it
* Review and agree to the Terms of Service
* Choose the location where you would like the installation to be saved
* Click Install
* Click Finish
* If a firewall blocker message comes up, click Allow Access. If not, disregard this step.
* It may prompt you to create a Unity account if you haven’t already at some point during this process.
* A prompt to Install Unity Editor should automatically pop up after that
* Click Install Unity Editor
* That should pull up Unity Hub where you can keep track of the installation progress. This could take several minutes.
* After it is finished, you will be able to open Unity Editor
* Make sure that it is the correct version

**Downloading Breaking Red**

1. Go to <https://github.com/GreenSkyGames/Breaking-Red>
2. Click the green Code button on the top right and select Download ZIP from the dropdown menu.
3. Navigate to where the Breaking-Red-main zip file was saved and click Extract all. Select the file path and then click Extract.
4. A screenshot of a computer

   AI-generated content may be incorrect.Open Unity Hub. It should look something like this.
5. Click Add in the top right and then select Add project from disk.

A screenshot of a computer

AI-generated content may be incorrect.

1. Select where the zip files were extracted. Once you find the Breaking-Red-main file, you may have to double click it to get to the correct file. You should see folders like UserSettings and Assets like shown below. Once you do, click Open.

A screenshot of a computer

AI-generated content may be incorrect.

1. The project will be added to your Unity Hub so then double click it to open it in Unity Editor. This will likely take a couple minutes.
2. Once Unity is open, make sure you are in the Assets folder at the bottom.
3. A screenshot of a computer

   AI-generated content may be incorrect.Click **Scenes** and then click **Start Menu.** This should open up the start menu of the game.
4. Under File in the top left corner, select Build and Run.

A screenshot of a computer

AI-generated content may be incorrect.

1. Choose where you would like to save the executable.
2. If a message saying that you are not a member of the project comes up, click Yes.
3. The game should open automatically after that.
4. Unity will automatically compile everything into the necessary files and create one convenient Breaking Red application file that you can click on whenever you would like to play. Note Unity creates several other files and folders that must be in the same directory of the Breaking Red application for the game to run.

**Quick Launch:**

After downloading the game for the first time, you should be able to launch it straight from your desktop the following times you wish to open it.

* Double click the Breaking-Red-main.exe file and it will run the game
* Check BC Mode if you do not want to take damage
* Use W-A-S-D keys to move
* Click P to pause the game at any time
* Click escape to exit the game

1. Code Organization

The code is organized based on the following context diagram and data flow diagram.

**Context Diagram:**

A diagram of a button input

AI-generated content may be incorrect.

A diagram of a company

AI-generated content may be incorrect.**Data Flow Diagram 0:**

The start menu will load our Level 1 game scene, where the player starts. Our game consists of 5 different levels in the world which can each be accessed by the preceding level. Our main features, set through prefabs and/or managers are the player, NPCs, collectibles and power-ups, character customization, the save system, UI elements, and the level builder.

* 1. Code Walkthrough

The game runs as follows:

* 1. The Start Menu scene is loaded from the executable. The player may choose to toggle BC Mode on or not; the scenes loaded will be nearly identical.
  2. From the Start Menu, the player will be directed to a character customization menu where they can change the color of Red’s cape.
  3. All level scenes contain the following elements:
     + Main Camera – Cinemachine Camera
     + Event System
     + Player (manages player controls and statistics)
     + Level Design (different maps are generated as the player moves through levels)
       - Platforms
       - Grid
       - Passageways
       - Fireplace Damage
       - Fade Panel
     + NPC Manager
       - NPC prefabs (controls NPCs’ reactions to proximity to and interaction with player. The player encounters different NPCs at different levels)
       - Dialogue Box Canvas prefab
       - Dialogue Manager
     + Game Events Manager (helps manage dialogue with NPCs)
     + Power Up Manager (controls power-ups’ reactions to collision with player. Each power-up has a different effect)
       - Power-Up prefabs
     + Canvas
       - Pause Menu
       - Power-Up Choice Prompt prefab
     + User Interface (keeps health bar updated)
     + Weather System (multiple different types of weather can occur)
     + Inventory Manager (manages inventory size and the functionality of adding items to inventory)
     + Save System (player’s progress will be saved at different intervals so that they can return to an in-progress game from the start menu)
  4. When the player’s health drops to or below zero, they fall off the map, or their sanity drops to or below 0, the Game Over scene is loaded with the option to Play Again or Quit.
  5. A diagram of a game

     AI-generated content may be incorrect.**Save System Class Diagram**
  6. A screenshot of a computer

     AI-generated content may be incorrect.**Level Builder Class Diagram**
  7. A diagram of a computer

     AI-generated content may be incorrect.**NPCs Class Diagram**
  8. A diagram of a game

     AI-generated content may be incorrect.**Character Creator Class Diagram**
  9. A diagram of a company

     AI-generated content may be incorrect.**Power-Ups/Collectibles/Inventory Class Diagram**
  10. A diagram of a company

      AI-generated content may be incorrect.**Scene Engine Class Diagram**
  11. A diagram of a game

      AI-generated content may be incorrect.**Menus/HUD Class Diagram**

1. Oral Exam

See the attached [Oral Exam Marking Key](https://webpages.uidaho.edu/drbc/cs383/assignments/OralExam_MarkingKey.pdf) for the areas that you will be marked on. This is worth 20% of your final grade. It is a completely individual mark. This will be a one-on-one 30 minute interview with the professor. (Or a 20 minute with the TA 20 minute with the professor for larger classes.) Walk through your feature and your code.

* Your code must function the way the client has requested.
* Your code must be approximately equal (or greater) in work requirement to your team mates.
* You must have at least two patterns and be able to justify their use.
* Your documentation must match your code.
* You must comply with your team's coding standards.

Walk through your time estimates and actual time spent.

**Where to go:**

JEB 324

**What to bring:**

(Arrive a bit early so you are logged in and ready to go when the time starts)

* Computer capable of running your code with access to your GIT repository.
  + Note: If it is not in your GIT repository it is not markable. (By definition, anything printed is out of date.
  + You cannot bring any study aids with you except what is in your code, but feel free to document your code as much as you want (for example if you think you might not remember what the key word ‘virtual’ does for static binding put a note about it in your code. )
  1. Contribution
* Gantt chart review – predictions vs. actual time, are you a good estimator?
* Run game and show where your code is called and run
* Show the code and walk through exactly what happens once it is called
  1. Technical
* Test plan
* Give an example where a test case later found a bug in your code due to things a teammate added later

4.3 Prefab

* Turning an object into a reusable asset
* Select Asset > Create Prefab and then drag an object from the scene onto the “empty” prefab asset that appears
* Or make a Prefabs folder in Assets and drag already made objects into the folder. That should turn them into Prefabs.
* Explain the purpose for each part of your code

4.4 Static or Dynamic Binding

* Have a class in your code where there could be static or dynamic binding
* Write mock code showing how you would set the static type and dynamic type of a variable. Choose a dynamically bound method.
  + What method gets called now?
* Change the dynamic type.
  + What method gets called now?
* Pick a statistically bound method.
  + Which one would be called in each of the previous two cases?
* When a method is virtual, it is resolved at runtime based on the actual object type.
* A screenshot of a computer code

  AI-generated content may be incorrect.When a method is not virtual, it is resolved at compile-time based on the pointer type.

4.5 Violate Copyright

* Show an example of reuse in your code where you violate copyright law
* How does it violate copyright?
* How did you integrate it into your code?
* Legal implications if you marketed it?
* Use fair use to argue that you can use it anyway

4.6 Software Patterns

Can refer to this link for descriptions of patterns: <https://sourcemaking.com/design_patterns>

* One big or two small patterns in code
* Why did you choose the pattern(s)?
* Draw class diagram for the pattern(s)
* Would something else have worked as well or better?
* When would be a bad time to use the pattern(s)?