Cutest Dungeon

Concept

by Nikolai Jay Summers

Introduction

Cutest Dungeon is a top down dungeon crawler which can be classified as a rouge-like.
You kill enemies to gain enough energy to advance to the next randomly generated dungeon.
On your way you pass shops and defeat bosses.
You also unlock different playable characters by completing mini quests.

Goals

The main goal of the game is to survive a set number of dungeon levels.
On your way you complete mini quests like for example “get more then 12 heart containers” or “kill all bosses” that unlock additional playable characters which you can play as on your next run and potions that help you survive.

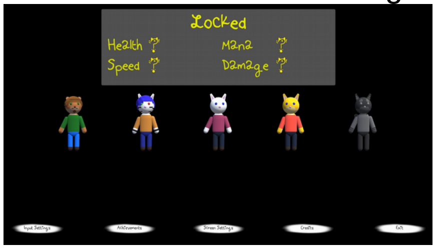
Game Mechanics and Mock up

The game starts with an intro screen with the name, which is followed by the main menu.
Here you can go into the input settings where you can change the controls, into the graphical settings to modify the display quality, access the credits and achievements and quit the game.
It is also the place where you can start a run by selecting a character.
Locked characters are grayed out and when you hover over them it says what you need to unlock them.

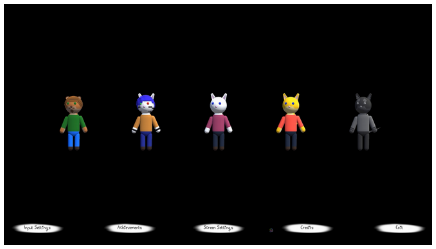
After the player has selected a character and a short introduction about their motives was shown, the first dungeon is randomly generated.
When the generation is completed the player has to kill a certain number of enemies.
Enemies can drop collectibles like coins, potions and similar items.
The generated level consists out of randomly generated dungeon tiles.
The enemies run towards the player as soon as they see them.
The UI the player sees has a health indicator made out of hearts.
On the right side there is an indicator for the currently selected item, the amount the player has of them and how much money the player has.
The selected item can be changed by scrolling if the input wasn’t changed in the input settings.
At the bottom of the screen a mana bar shows how much mana is left.
It is used to fire magic. It regenerates slowly. The player can’t fire magic if they doesn’t have mana.
Above the mana bar is an xp bar. It gets filled by killing enemies. When it is full the player has the option to advance to the boss fight by holding the “use” button while the portal is selected.
By pressing escape a menu is triggered that asks the player if they want to quit the game.
Saying yes will result in returning to the title screen.
Here the player can also change the volume.
If the players health drops below zero, they die and return to the character select screen after a dialog.

Logo Screen

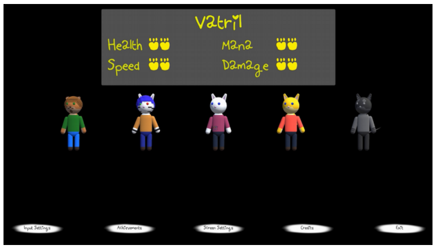
Title Screen



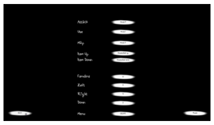
Locked Character



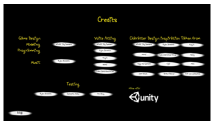
Main Menu



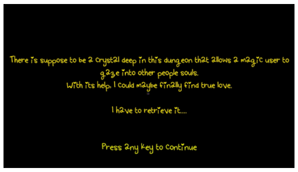
Unlocked Character



Input Settings



Credits



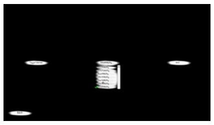
There is suppose to be a crystal deep in this dungeon that allows a magic user to gaze into other people souls.
With its help, I could maybe finally find true love.

I have to retrieve it....

Press any key to continue

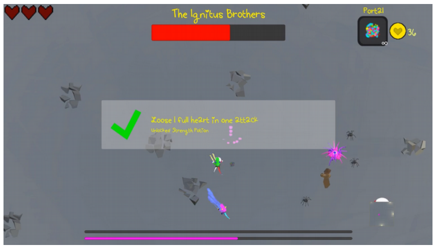
Story Screen

Achievements



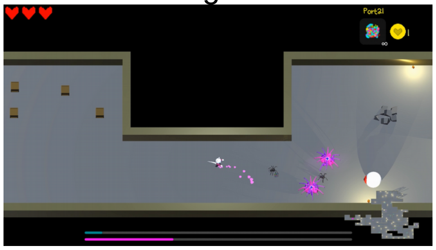
Graphics Settings

Loading Screen

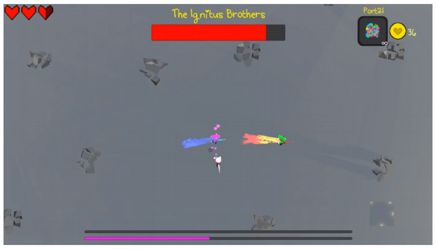


Loose 1 full heart in one attack
Unlocked Strength Potion

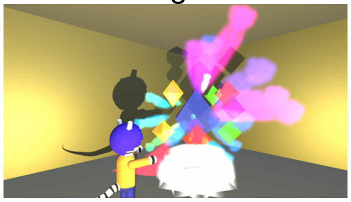
Got Achievement



Dungeon



Boss Fight



End Scene

Details and Technical Concept

Character Control

- The character is controlled with the the mouse and the keyboard.
- The look and fire direction is controlled by the mouse position.
- The firing is handled by a particle system and raycasts.

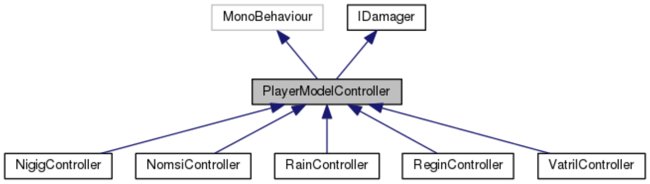


Illustration 1: PlayerModelController Inheritance

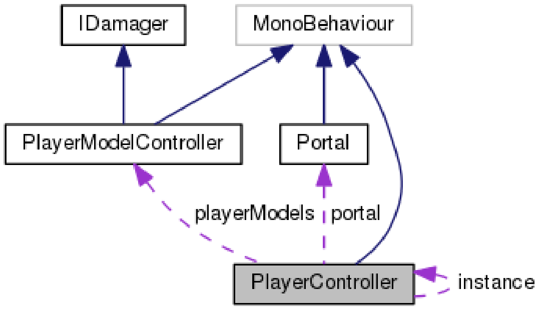


Illustration 2: PlayerModelController Dependencies

Dungeon Generator

- The dungeon is constructed out of tiles.
- Each tile randomly generates props and walls.
- The tiles are put together in a way that builds a dungeon surrounded by walls.
- A Navigation Mesh is also generated.
- Enemies are spawned after the Navigation Mesh is done.

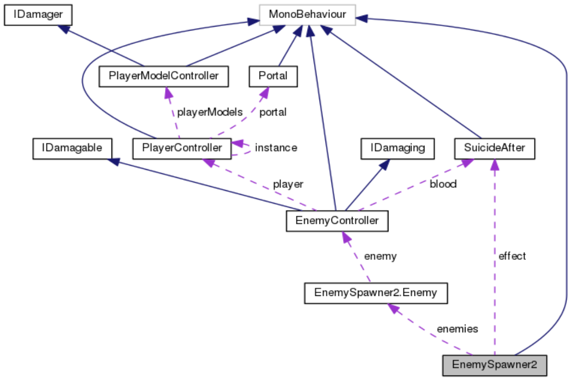


Illustration 3: EnemySpawner2 Dependencies

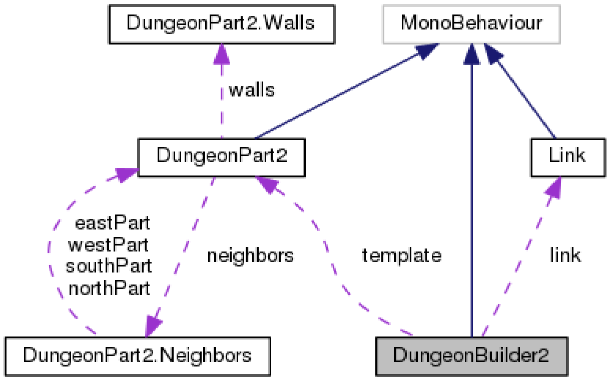


Illustration 4: DungeonBuilder2 Dependencies

Enemies

- All enemies inherit from the same base enemy controller.
- They use raycasts to see the player and run towards them if they do.
- For navigation they use a navigation mesh agent.
- On death they have a random chance to drop item.

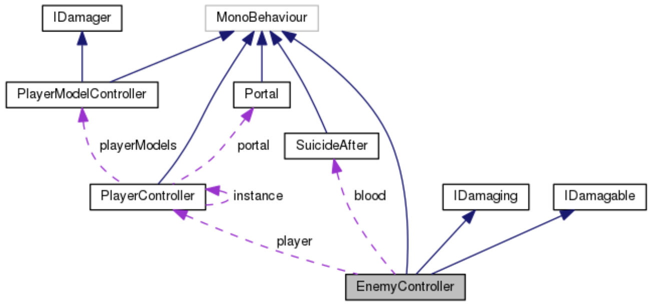


Illustration 5: EnemyController Dependencies

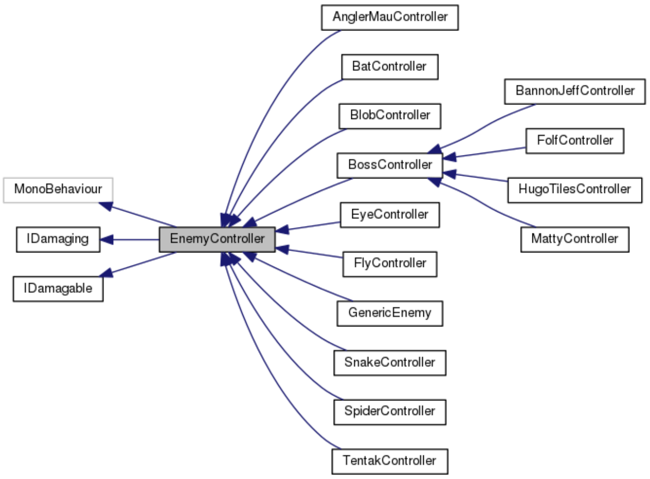


Illustration 6: EnemyController Inheritance

Customization

- An input manager has been written that lets the user change all inputs
- The graphics settings have been moved from the default unity launcher to a menu

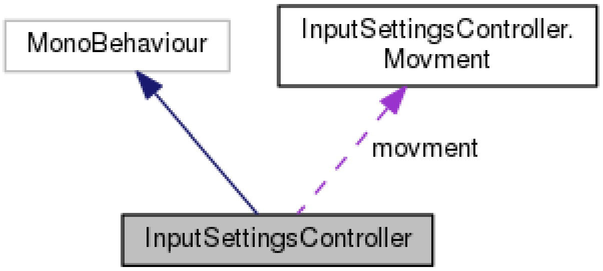


Illustration 7: InputSettingsController Dependencies

Bosses and Boss Fights

- The base boss controller inherits from the base enemy controller
- At the start of each boss fight a random undefeated boss is selected
- If all bosses have been defeated the player has to fight all at once with reduced life

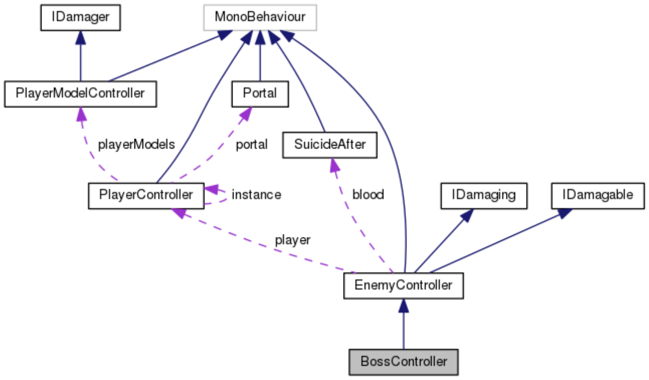


Illustration 8: BossController Dependencies

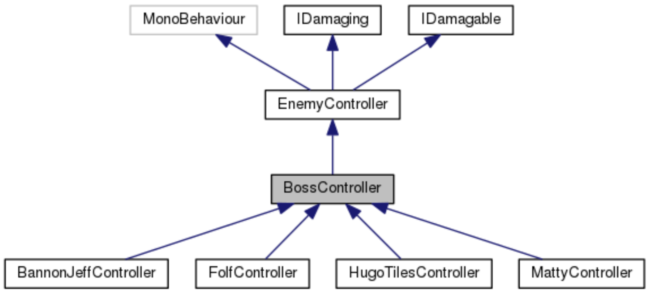


Illustration 9: BossController Inheritance

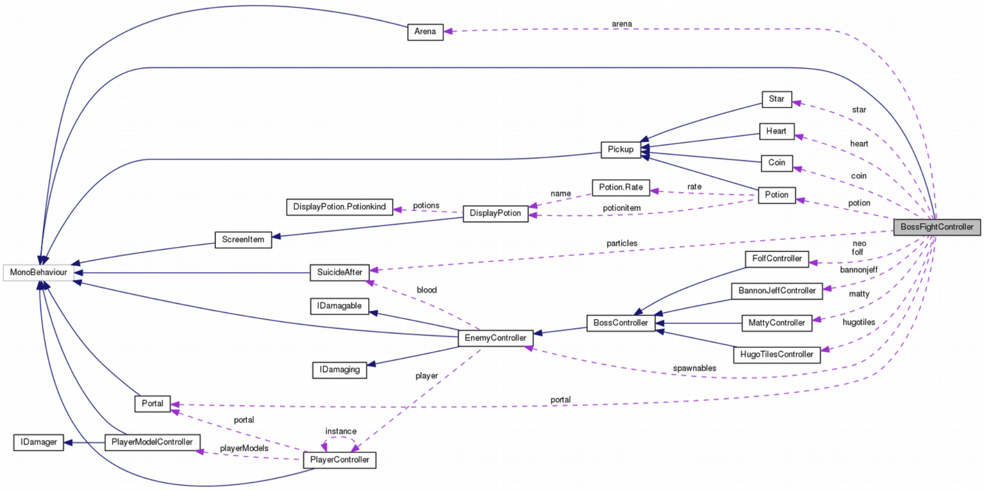


Illustration 10: BossFightController Dependencies

Damage

- All gameobjects with the IDamaging interface can hurt the player when they touch them.
- All gameobjects with the IDamageable interface can receive damage from the player.

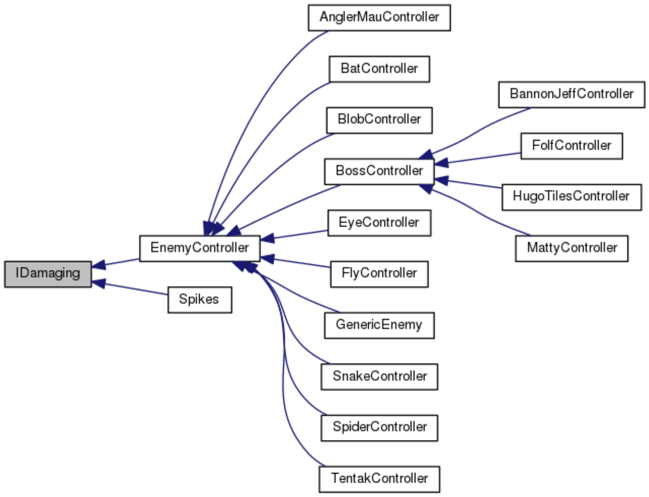


Illustration 11: IDamaging Inheritance

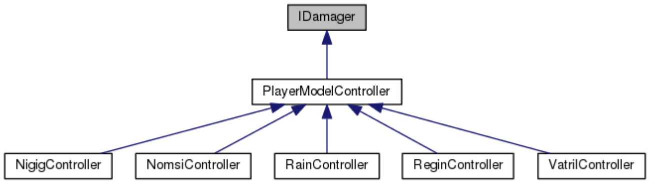


Illustration 12: IDamager Inheritance

Pick Up and Screen Items

- All gameobjects that inherit from the Pickup class can be picked up and do something with the player like increase their money or give them a potion.
- All gameobjects that inherit from the ScreenItem class can be displayed in the item section on the screen

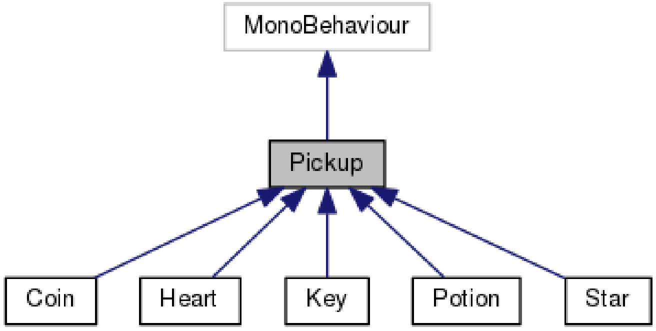


Illustration 13: Pickup Inheritance

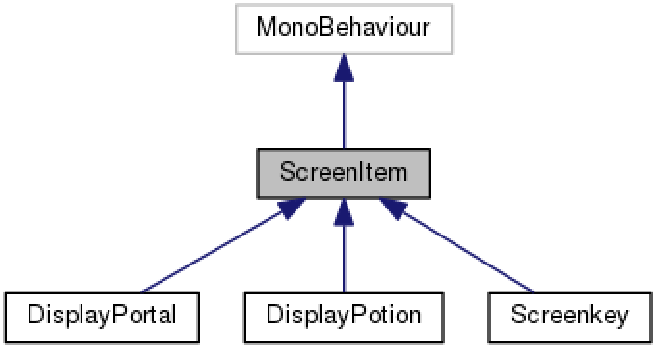


Illustration 14: ScreenItem Inheritance

Mini Map

- The mini map is rendered by a second camera that floats above the dungeon.
- When the level loaded it positions itself.
- A position marker indicates the players positions.
- It can be displayed in full screen with a button press.

Shop

- The shop can be entered when the player has a shop key.
- It can not be accessed during a boss battle.
- The player can buy unlocked potions and heart containers.

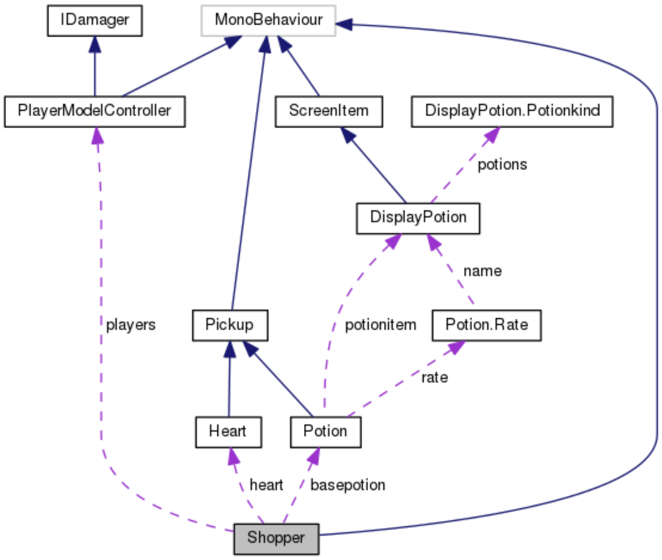


Illustration 15: Shopper Dependencies

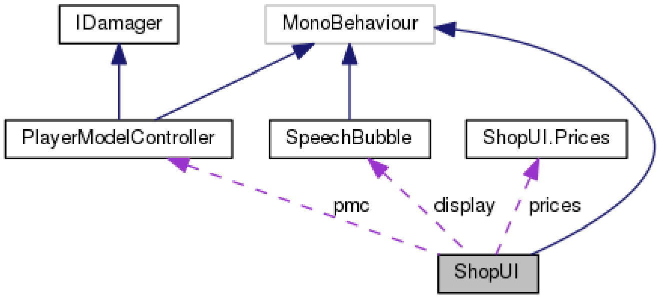
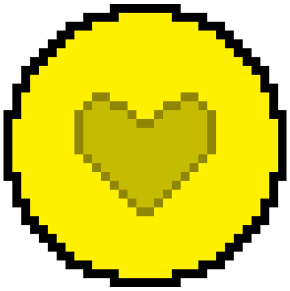


Illustration 16: ShopUI Dependencies

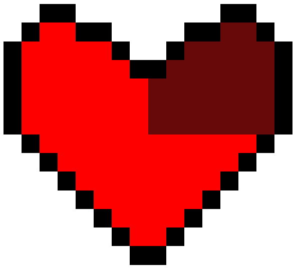
Art Concept

Modeling and Animation

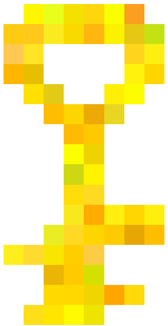
- Five playable characters are modeled and animated with different animations for idling, shooting and walking
- Multiple enemies, bosses, props and items are also modeled and animated.
- Screen items, ui-elements like buttons and shooting particles are drawn.



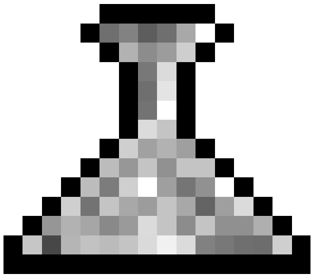
Drawing 1: Coin



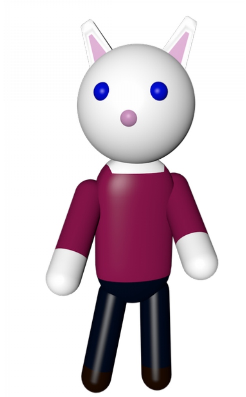
Drawing 2: Three Quarter Heart



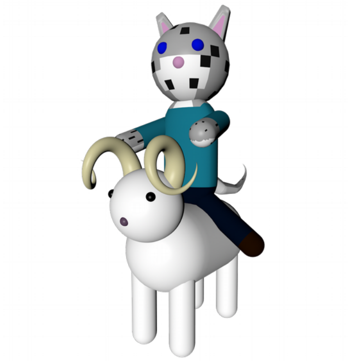
Drawing 3: Key



Drawing 4: Potion



Model 1: Vatril



Model 2: Dimensiona

Third Party Content

Frameworks and Libraries

- The game runs in the Unity Engine
- The Unity Technology Extras NavMeshComponents from the Unity GitHub package are used.
- Various examples from unity owned websites have been used as inspiration.

Art

- The music was made by Ryan Strobach
- Font “Kittyspoon” was made by Lauren Krunk
- The voice acting that was not done by me was done by:
  - Ryan Strobach
  - Evan Chartrand-McCue
  - Simon*
  - Regin*
- Character design inspiration has been taken from:
  - Ryan Strobach
  - Evan Chartrand-McCue
  - Simon*
  - Regin*
  - Matty*
  - Hugo De Guardia
  - Tyler*
  - Matt*
  - Jeff Davis
  - Brian Fahrner
  - Eric*

  * didn't want to be named by complete/real name