Player Movement Prototype

Documentation

CyanX

SORA KHAN

SHARLENE VON DREHNEN

OCTOBER 2018

Introduction

The game 'CyanX' is based in the distant future in a game world known as Khandren. In this game prototype, we will be playing as the main character, Mia. The game consists of 3 levels. Level 1 is located in *Sandenko*, level 2 is located in *Cypro* and level 3 is located in *Icerun*. The mission is for Mia to rescue captured animals from the Nezbots who are guarding them.

Controls

Left Keyboard arrow or 'A'	Walk left
Right keyboard arrow or 'D'	Walk right
Spacebar	Jump
'M' Key	Open/Close the menu
'Z' Key	Attack

Running the Game

You can play the executable file, **CyanX.exe**, provided in the folder **Run CyanX Build.** To run the game using unity, please ensure you have installed Unity on your PC (2017 versions or higher).

System Requirements for Unity:

OS: Windows 7 SP1+, 8, 10, 64-bit versions only; Mac OS X 10.11+.

GPU: Graphics card with DX10 (shader model 4.0) capabilities.

1. Run Unity
2. Select 'Open'
3. Navigate to the Unity Folder. This is the folder called 'CyanX' and open it.
4. Select the play button at the top of the window.

Code Contributions

Created Code

The following code can be found in **Code Files/Assets/Scripts**

1. Implemented character to walk left/right.
   - **Player.cs** - the first line of the **Update()** function.
2. Implemented the jumping functionality and the restriction to only jump once.
   - **Player.cs** - the first if-statement in the **Update()** function, and the first if-statement in the **OnCollisionEnter2D()** function.
3. Implemented the main character's animation states between walking and running.
   - **PlayerMovement.cs** - In the **Update()** and **FixedUpdate()** function.
4. Implemented the Health UI (hearts update when character's health changes)
   - **HUD.cs** - In the **Update()** function
5. When the player touches one of the poisonous plants, the player's health is reduced.
   - **Player.cs** - in **OnCollisionEnter2D()** function.
6. Implemented the player dying and the checkpoint in the middle of the level.
   - **Player.cs** - in **die()** function (the if-else statement).
7. When an enemy touches the player, the health goes down by one.
   - **Player.cs** - in **OnCollisionEnter2D()** function.
8. Implemented moving platforms (platforms that move up and down).
   - **MovingObject.cs** - in **Update()** function.
9. Implemented the camera following function
   - **CameraMovement.cs** - in **Update()** function.
10. Implemented sound effects for when the character jumps or gets hit by an enemy.
    - **Player.cs** - in **Update()** function (call jumpSound.play()).
    - **Player.cs** - in **OnCollisionEnter2D()** function (call healthDamagedSound.play()).
11. Implemented pick-ups for XP, rescued animals and resources for the inventory.

- **InventoryAndSkills.cs** - all of the **pickUpItem()** function

12. Implemented the system that allows the player to level up (up to level 10).
    - **InventoryAndSkills.cs** - all of the **LevelUp()** function, part of **Update**()
13. Implemented the different skills that could be unlocked based on which level the player has reached.
    - **Weapon.cs** - part of the **craft()** function (restricts which weapon the player can create as some weapons need to be unlocked first)
    - **Player.cs** - Sections of the **OnCollisionEnter2D()**. For example, if the player is on level 7, they no longer get affected by spikes
14. When the player reaches level 9, XP floats towards them.
    - **Level10FloatingXP.cs** - whole class. Only the **start()** and **update()** functions
15. Implemented the UI screen that appears when the player levels up
    - **GameManager.cs** - In **TriggerLevelUpUI()**
16. Implemented the ability for the player to craft a weapon.
    - **Weapon.cs** - all of the **craft()** function. The functions **craftBaseballBat()**, **craftKnife()**, **craftDiamondSword()** and **craftShishkebab()** check whether or not the weapon has been crafted yet
17. Implemented the ability for the player to equip a weapon.
    - **Weapon.cs** - If the weapon has already been crafted, the **equip()** function is called instead. This uses **swapEquippedWeapons()** which unequips the currently equipped weapon
18. Implemented the enemy's movement. If the player gets too close, they will become hostile and follow the enemy.
    - **EnemyMovement.cs** - In the **Update()** function
19. Implemented the ability for the main character to attack the enemy, along with the animations for the different weapons.
    - **Attack.cs** - in the **hit()** function
20. Implemented the enemy's health system, and the damage done depending on the type of weapon equipped.

- **Attack.cs** - in the **hit()** function
- **EnemyMovement.cs** - in the **die()** function

21. Implemented trapdoors. When a button is pushed, the trapdoor will open.

- **TrapdoorFunctions.cs** - whole class. Only the **start()** and **update()** functions
- **ButtonPushed.cs** - whole class. Only **start()** and **OnCollisionEnter2D()** functions

22. Implemented the navigation between levels and the 'mission complete' screens.

- **GameManager.cs** - In the **Update()** and **nextLevel()** functions

23. Implemented saving game data between scenes and deleting them when game is finished.

- **GameManager.cs** - Saves game data in **nextLevel()**
- **GameManager.cs** - Deleted game data in **BackToMain()** and in **Update()** if the player chooses to end the game from the pause menu

24. Implemented the in-game pause menu for when the player pauses the game. The menu includes the buttons 'Stats,' 'Crafting' and 'Back to main menu.'

- **Player.cs** and **EnemyMovement.cs** - code for pausing the game is added in the **Update()** functions for both classes
- **GameManager.cs** - **mainMenu()** function controls turning on/off the menu

25. Implements the sub-menu for the Stats page.

- **GameManager.cs** - uses **openStatsMenu()** and **closeStatsMenu()**
- **InventoryAndSkills.cs** - code throughout **levelUp()** and **pickUpItem()** functions update this page

26. Implemented the sub-menu for the Crafting page.

- **GameManager.cs** - uses **openCraftingMenu()** and **closeCraftingMenu()**
- **Weapon.cs** - code throughout **craft()**, **equip()** and **swapEquippedWeapons()** functions update this page based on which weapon is equipped or not

- **InventoryAndSkills.cs** - **pickUpItem()** updates the inventory in real time as the player picks up new resources (such as wood, metal or diamonds).

Re-used/Referenced Code

Note: Some resources or videos from Assignment 1 were re-used or modified when expanding the game for Assignment 2.

1. CameraMovement.cs
    - The movement of the camera to follow the player is from the online tutorial: https://www.youtube.com/watch?v=BQEsbOALKhc
2. MovingObjects.cs
    - The following video shows how to make a platform move from one point to another: https://www.youtube.com/watch?v=HMZnSZswTmU
    - This was modified to make trapdoors move when the button was pressed
3. MainMenu.cs
    - Created with the help of Unity Tutorial by Brackeys from YouTube: https://www.youtube.com/watch?v=zc8ac_qUXQY
    - This helped us understand how to create UI screens. Therefore, we were able to create the pause menu and crafting and stats sub-menu pages
4. Enemy Movement.cs
    - This webpage helped us to make the enemies sprites changed based on whether or not they were hostile towards the player: https://forum.unity.com/threads/how-to-change-sprite-image-from-script.212307/
    - The code in this thread was used and modified to get the enemy to follow the player: https://answers.unity.com/questions/274809/how-to-make-enemy-chase-player-basic-ai.html
5. GameManager.cs
    - The following thread explained how to load a scene which helped us load from one scene to another at the end of a mission: https://forum.unity.com/threads/how-can-i-load-a-specific-scene.395059/

- Game Over: https://www.youtube.com/watch?v=VbZ9_C4-Qbo&t=168s
- We watched this tutorial from the Unity website for the DontDestroyOnLoad() function, which we were unable to get working, so we used PlayerPrefs instead (both are explained in the video): https://unity3d.com/learn/tutorials/topics/scripting/persistence-saving-and-loading-data

6. Attack.cs

- This tutorial explained how to change the player's animation within the script: https://www.youtube.com/watch?v=hkaysu1Z-N8

7. Player.cs

- 2D Movement in Unity - https://www.youtube.com/watch?v=dwcT-Dch0bA
- Used the following tutorial to help learn how to import and use sounds: https://www.youtube.com/watch?v=4cwNx-Lhymc
- Player Health Implementation - A tutorial that shows how to change the current health - https://www.youtube.com/watch?v=l6rOHJitheY
- The following website shows the code to restart the level, which was used after the player's health reached <= 0 (and before reaching the 2nd checkpoint): https://answers.unity.com/questions/890561/reset-scene-when-player-dies.html.
- Code for trying to change the current player's x position: https://answers.unity.com/questions/188998/transformposition.html

8. CharacterController2D.cs & PlayerMovement.cs

- **CharacterController2D.cs** and **PlayerMovement.cs** are scripts shared by YouTuber Brackeys via GitHub which manages the player's controls, such as measurement to jump, and speed to walk/run.
- https://github.com/Brackeys/2D-Character-Controller
- https://www.youtube.com/watch?v=dwcT-Dch0bA

9. HUD.cs

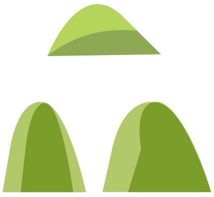
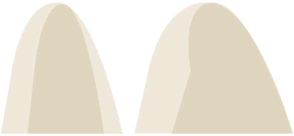
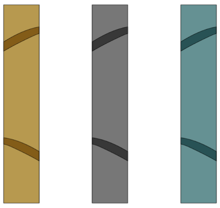
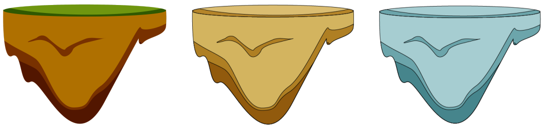
- The following tutorial showed us how to create changing images for the heart UI: https://www.youtube.com/watch?v=5KwkfGfaRNU

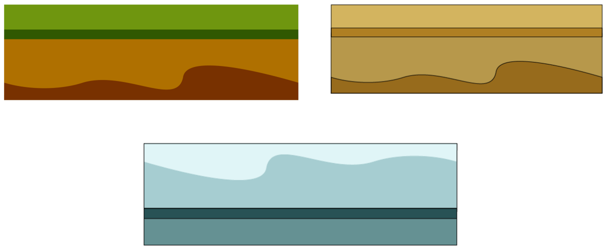
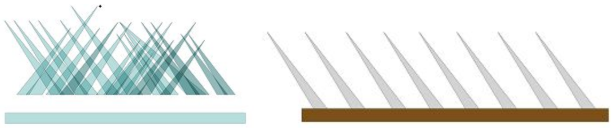
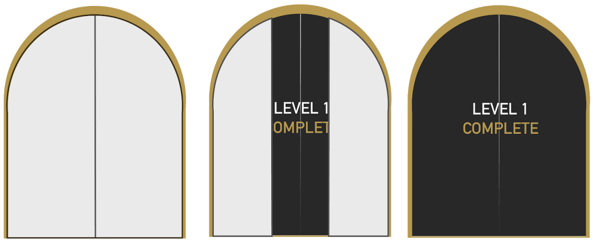
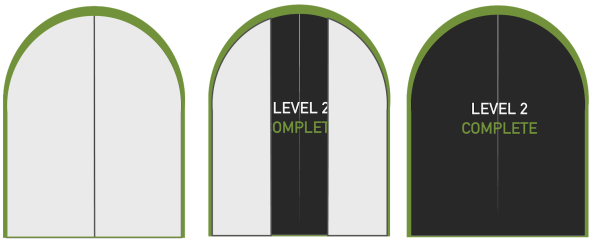
Many parts of the game such as storing resources in the inventory, crafting or equipping items were created based on knowledge acquired in the videos mentioned and were not made by directly following tutorials.

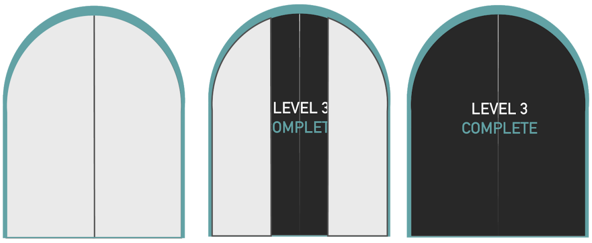
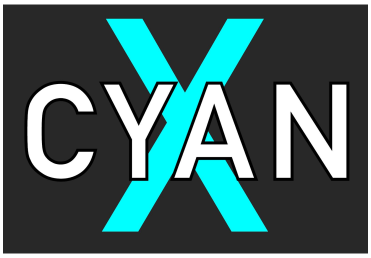
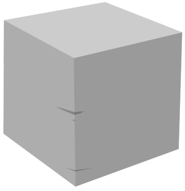
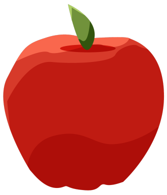
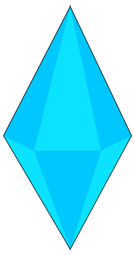
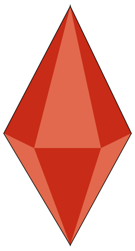
Asset Contributions

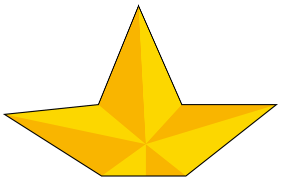
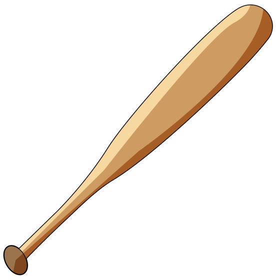
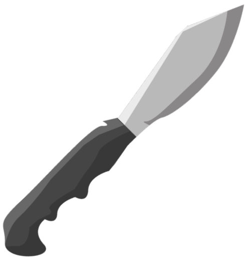
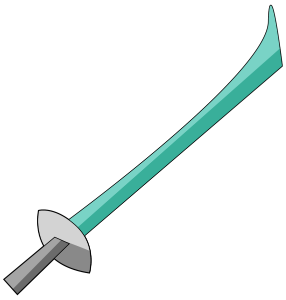
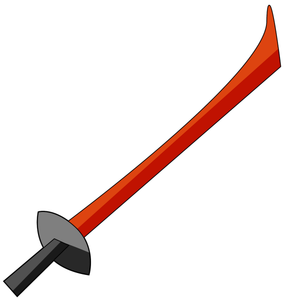
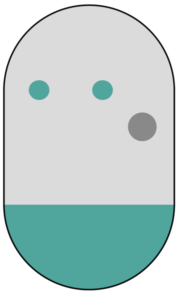
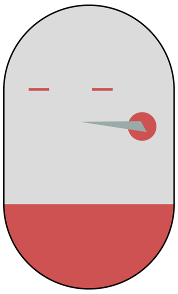
Created Assets

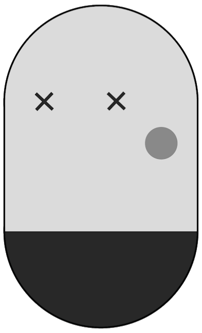
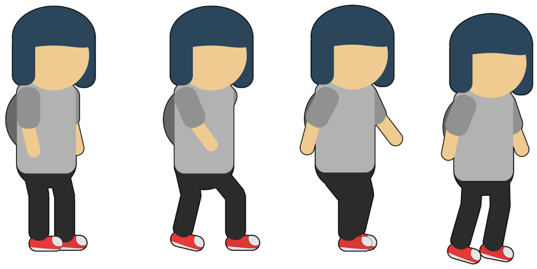
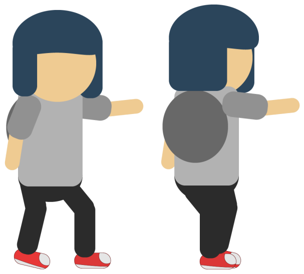
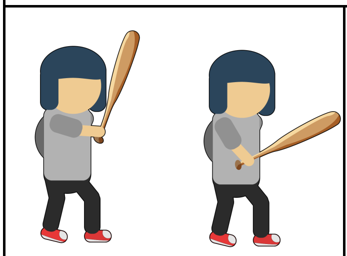
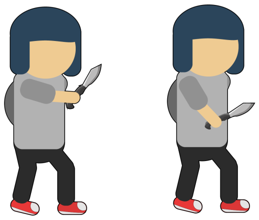
1. Sounds
    - *Theme music* **"Cyan"** was made in **Figure** music app on iPhone.
2. Sprites
    - Below is a list of sprite assets we created, including screenshots and where the asset was made.

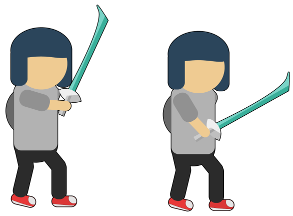
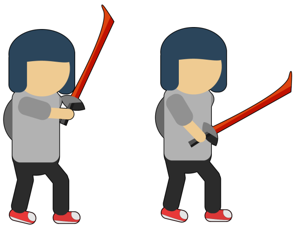
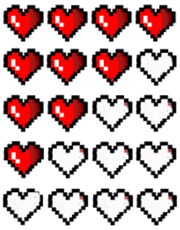
ASSETS		DESCRIPTION
World Objects		
 **Green Hills**	 **Sand Dunes**	Background filler depending on the levels. **Created with Adobe Illustrator**
 **Sliding doors/floors**		Sliding doors or floors the user has to try and figure out to open to access pickups or rescue animals. These doors/floors need to be opened by a button! **Created with Adobe Illustrator**
 **Moving Platforms**		Moving platforms that are different depending on the level/place the player is in. **Created with Adobe Illustrator**

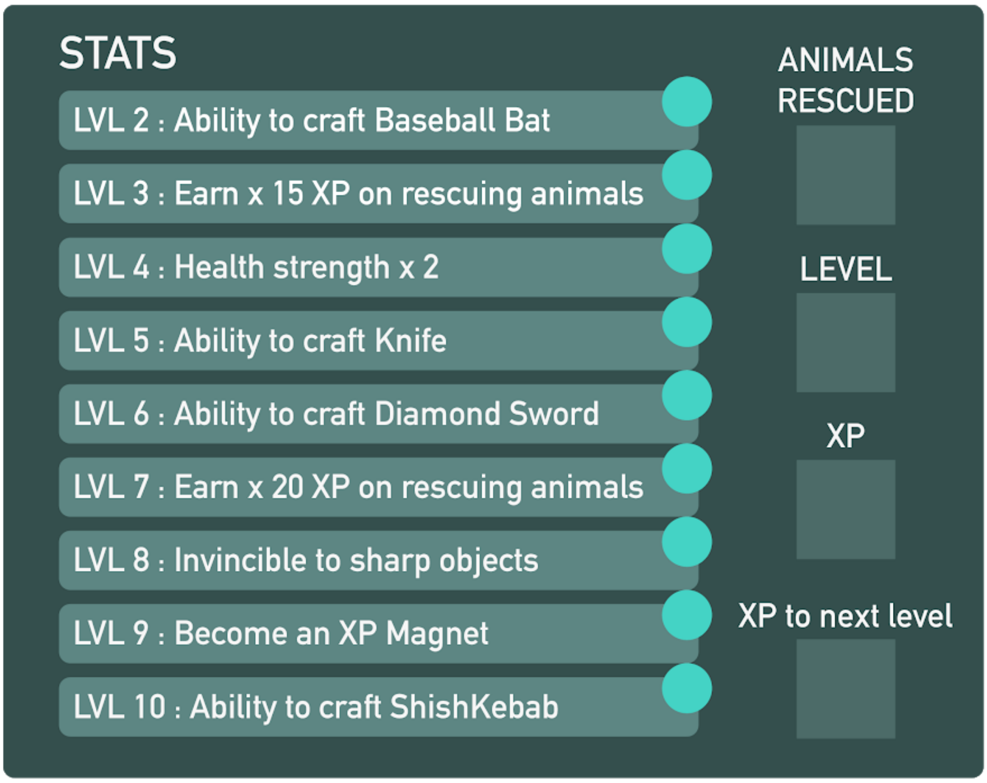
 **Ground/Ceiling**	Ground/ceiling sprites that are different depending on the level/place the player is in. **Created with Adobe Illustrator**
	Sharp spikes that change depending on the level the player is on. **Created with Adobe Illustrator**
 Cloud	Cloud used as a background filler. **Created with Adobe Illustrator**
 **Level 1 - Exit Door**	Exit doors depending on the level player is at. **Created with Adobe Illustrator**
 **Level 2 - Exit Door**	

 **Level 3 - Exit Door**		
 **CyanX Main Screen Logo**		Logo used in the main menu screen **Created with Adobe Illustrator**
PICK-UPS		
 **Metal Block**	 **Wood Block**	**Metal** and **wood** blocks that are collectable items to craft weapons! **Created with Adobe Illustrator**
 Rock	 **Apple**	The **rock** is also a collectable to build weapons. **Apple** is for health! **Created with Adobe Illustrator**
 **Blue Diamond**	 **Ruby**	Both again are pickups to craft weapons. Rubies are rare! **Created with Adobe Illustrator**

 **XP (haffsta)**		XP points that the player collects throughout the game in order to have upgraded abilities, upgrade their current level/skill. This all depends on how many they collect. **Created with Adobe Illustrator**
WEAPONS / CRAFTABLE OBJECTS		
 **Baseball Bat**	 **Pocket Knife**	Craftable weapons that the player will need to create by collecting certain numbers of a various combination elements that make them up. **Created with Adobe Illustrator**
 **Diamond Sword**	 **ShishKebab**	
CHARACTER AND ENEMIES		
 **Friendly Nezbot**	 **Hostile Nezbot**	Nezbots are enemies. They are harmless when blue and hostile/harmful when red. They turn red when the main character is close to them. Cute, yet deadly. **Created with Adobe Illustrator**

<table>
<tr><td colspan="2">
Disfunctional Nezbot</td><td>Disfunctional Nezbot. Does not chase of attacks Mia. **Created with Adobe Illustrator**</td></tr>
<tr><td colspan="2">
Idle, Walking and Jumping Mia</td><td>Main/playable character named Mia. She can stay still, walk/run, and jump!
Created with Adobe Illustrator</td></tr>
<tr><td colspan="2">
Mia attacking without weapon equipped</td><td>If Mia tries to attack when she has no weapons equipped.
Created with Adobe Illustrator</td></tr>
<tr><td>
Mia with Baseball Bat</td><td>
Mia with Pocket Knife</td><td>Sprites of Mia attacking with different weapons. Once equipped with an item, she does not like the idea of punching robots with her fists!
Created with Adobe Illustrator</td></tr>
</table>

 **Mia w. Diamond Sword**	 **Mia w. ShishKebab**	
STATS, INVENTORY / CRAFTBOX + MORE		
 **Health**	This is a sprite sheet containing all possible state of the player's health. Each are stored in an array that corresponds with the current health of the player. **Created using Krita.**	
CHECKPOINT **Checkpoint Indicator**	The sign that lets the player know that they've passed a checkpoint. **Made using Krita.**	



STATS

LVL 2 : Ability to craft Baseball Bat

LVL 3 : Earn x 15 XP on rescuing animals

LVL 4 : Health strength x 2

LVL 5 : Ability to craft Knife

LVL 6 : Ability to craft Diamond Sword

LVL 7 : Earn x 20 XP on rescuing animals

LVL 8 : Invincible to sharp objects

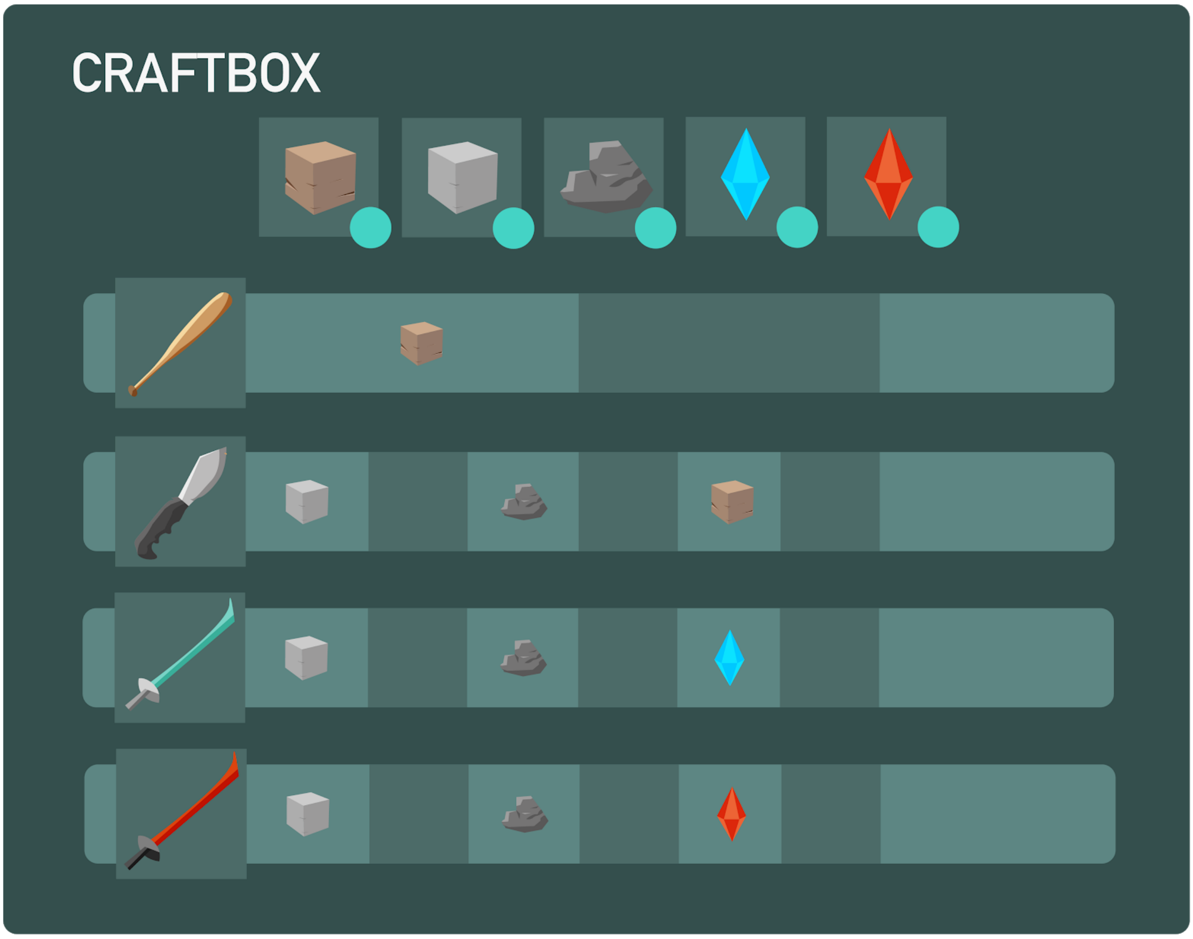
LVL 9 : Become an XP Magnet

LVL 10 : Ability to craft ShishKebab

Character Statistics

Shows what different level ups can do to their abilities.
Also shows the number of animals rescued, what level they currently are in, how much XP they have. Including how much XP they need to go to the next level.

Created with Adobe Illustrator



Character Inventory / Crafting Area
Shows the number of pickups the player has collected throughout the game. This is an asset that would have numbers displayed dynamically in-game - which is why there is empty blocks in the image. This area is also where the player can craft items depending on the number of pickups collected.
Created with Adobe Illustrator

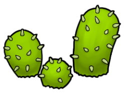
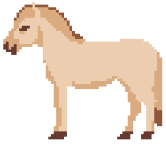
Referenced Assets

3. Sounds

- Player’s sound effects were made using https://www.bfxr.net/

4. Sprites

- The following is a list of sprites that we used from the internet. For Assignment 3, we focused mostly on creating our own sprites.

Name	Image	Reference
Cactus (poisonous plant)		Cactus.png was downloaded directly from: http://1.bp.blogspot.com/-J_EktHWrtfA/T_QsusglTHI/AAAAAAAAA50/D95c2fXfKrg/s1600/Edge_Obstacle_Cacti.png
Animal to Rescue		Pheurus.png was downloaded directly from: http://machineboy.com/wp-content/uploads/2015/09/245.png