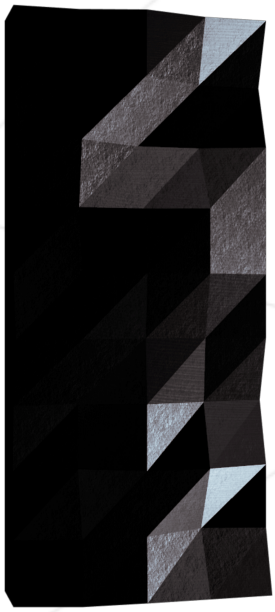

Unity Piscine - Rush01

Hack and Slash

Staff staff@staff.42.fr

Summary: This document contains the rush01 subject for the Unity Piscine of 42.

Contents

Chapter I

Consignes

Sauf contradiction explicite, les consignes suivantes seront valables pour tous les jours de cette Piscine.

- Seul ce sujet sert de référence : ne vous fiez pas aux bruits de couloir.
- Le sujet peut changer jusqu'à une heure avant le rendu.
- Les exercices sont très précisément ordonnés du plus simple au plus complexe. En aucun cas nous ne porterons attention ni ne prendrons en compte un exercice complexe si un exercice plus simple n'est pas parfaitement réussi.
- Attention aux droits de vos fichiers et de vos répertoires.
- Vous devez suivre la procédure de rendu pour tous vos exercices. L'url de votre dépot `GIT` pour cette journée est disponible sur votre intranet.
- Vos exercices seront évalués par vos camarades de Piscine.
- En plus de vos camarades, vous pouvez être évalués par un programme appelé la Moulinette. La Moulinette est très stricte dans sa notation car elle est totalement automatisée. Il est donc impossible de discuter de sa note avec elle. Soyez d'une rigueur irréprochable pour éviter les mauvaises surprises.
- Les exercices shell doivent s'éxcuter avec `/bin/sh`.
- Vous ne devez laisser aucun autre fichier que ceux explicitement specifiés par les énoncés des exercices dans votre dépot de rendu.
- Vous avez une question ? Demandez à votre voisin de droite. Sinon, essayez avec votre voisin de gauche.
- Toutes les réponses à vos questions techniques se trouvent dans les `man` ou sur Internet.
- Pensez à discuter sur le forum Piscine de votre Intra et sur Slack !
- Lisez attentivement les exemples car ils peuvent vous permettre d'identifier un travail à réaliser qui n'est pas précisé dans le sujet à première vue.
- Réfléchissez. Par pitié, par Thor, par Odin !

Chapter II

Today's specific instructions

Here is the final rush of your Unity piscine. This is the final opportunity to demonstrate everything you've learned and the skills you've acquired. For once, you will be able to use ALL the tools at your disposal. You can use the WHOLE asset store (even the scripts), all the 3D models/sounds/ music/etc you will find on the web as well as all the assets we have provided during this piscine.

The goal of this rush is to achieve the most stylish game you can in 2 days. That is the average length of a GameJam.

Chapter III

Preamble

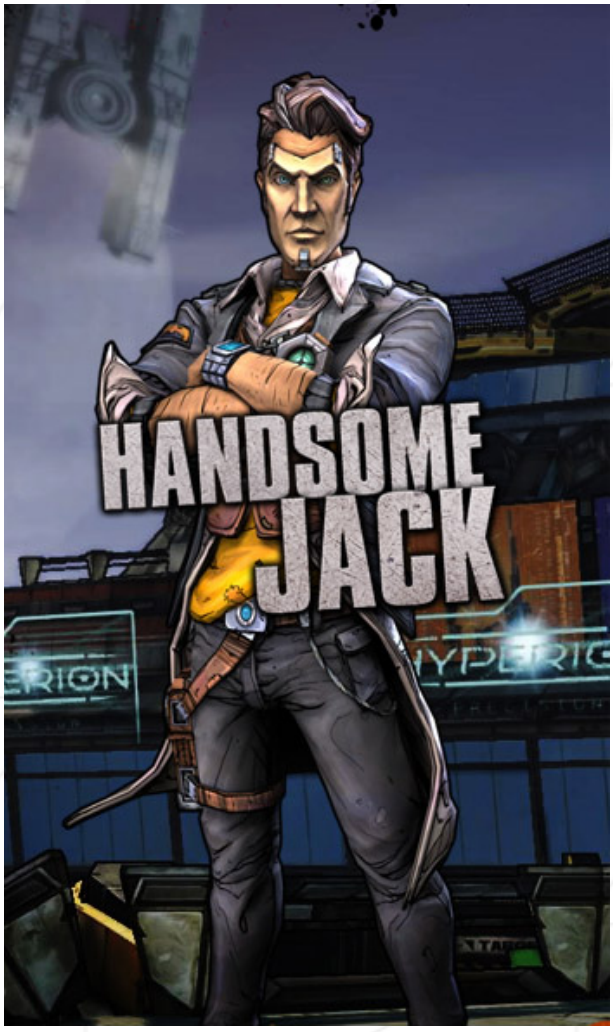

Figure III.1: "I'm rackin' my brain trying to think of a name for that diamond pony I bought. I was gonna call it "piss-for-brains" in honor of you, but that just feels immature. Maybe... "Butt Stallion"? Nah, that's even worse. I'll give it some more thought."

Chapter IV

About collaborations - the return!

Here is a little recap of what you should do to make sure you work on Unity as a group in serenity. To help you save hours, days and several headaches, here are precious advice you should definitely read before approaching team work on Unity.

- Because we care, you'll find a .gitignore in today's assets. It will help you keep files that don't need to push on the repo, local.
- Don't `EVER` work simultaneously on the same scene or prefab. Unity still has hard time managing the merge on this kind of asset.
- Watch your pair working's code when you modify an object containing a script. One missing tag can destroy a whole functionality.
- Communication is key! Always share with your partner the modifications you've made on the project.
- Dispatch the work properly. Try not to touch the same gameplay elements to avoid conflicts.
- Make several commits. In case of conflict, you will be able to go back and find a version that works.
- Set up "milestones", that is, operational versions of the game at key moments of its development. Synchronize your local git repos before getting to the next step. Thus, you'll have a reliable version if either of you breaks something and in the worst case scenario, you'll be able to push your latest valid "milestone" and avoid ending with a non-functional project.

Chapter V

Partie obligatoire

I suppose you've been a little frustrated by D08, because you've designed a playable Hack and Slash... with no gameplay. Which is rather sad, I reckon. Let's address the problem and finish this Hack and Slash making it top notch. Let's design real maps and implement a loot system, skills, a little more than 2 enemies, etc... Let's make an accurate listing.

V.1 La base

First, let's list the requirements of your basic Hack and Slash:

- A map with a NavMesh.
- A top down camera, centered on the hero controlled by the player (you can keep Maya or take any character as long as it's rigged/ animated).
- A hero controlled with the mouse: left clicking on the terrain moves it, left clicking an enemy makes it attack, the hero keeps attacking if the the player keeps the left click pressed.
- Enemies that chase and attack the hero if it enters their detection area.
- All the characters in the game have idle/run/attack/death animation.
- An advanced combat system based on stats: strength increases damage, agility increases chances of hitting and dodging, constitution increases life.
- An XP system. Each enemy killed earns X experience points to the hero. When the hero earned enough XP, they reach the next level and gains 5 skill points they can dispatch among their main stats.
- Enemies that spawn have the same level as the hero, whether it's 1 or 50.
- A HUD displays the hero's and selected enemy's life as well as their respective levels and the hero's XP.
- A character's window that sums up all the stats: STR, AGI, CON, minDamage, maxDamage, xp, xpToNextLvl, hpMax.

- Life potions the enemies drop randomly. They give the hero 30of their max life when they walk on it.

V.2 Skills

For our Hack and Slash to get a little more interesting, you will have to add a few skills and a nice little skill tree!

- With each level, the hero will gain a skill point. You can see how much they have to use in the skill tree window.
- The tree opens when you press the "N" key. You can spend the skill points to unlock skills. The skill tree is made of several tiers that gets unlocked every 6 levels. This means from level 1 to 5, the first tier skills will be the only ones available. The 2nd tier skills will be unlocked after the 6th level and so on.
- Each skill can be enhanced several times (5, for instance). Each new skill point spent in a skill enhances it.
- You must implement at least 6 skills among which: a area attack to the ground (with a template to aim) applied with the mouse. An area attack appearing around the hero and following their movements, a direct damage spell (like a fireball), a passive skill to enhance one of the hero's characteristics (except their basic skills), a healing spell.
- Each skill must have a tooltip explaining how it works, its damage, and its next level's stats (so that the player knows if it's worth enhancing it).
- Skills might require a special resource like mana or rage to be used. You can also implement a simple cooldown system (hint: it will be easier but feel free to choose your favorite technic).
- Skills must have a particle effect or something that lets the player know their action range.
- You must also create a skill bar that can contain a limited amount (let's say between 4 and more than 4) so that the player must choose and cannot use all of their skills at once. You must create a way to equip these skills on this bar and allow the player to shoot them pressing 1, 2, 3... keys.

V.3 Loots

What would be a Hack and Slash without loots? It's like playing Pong without rackets: it sucks. So, we're gonna add objects the player will be able to pick up and an inventory to stock them.

You will create a system so that the player can equip their objects. Weapons, for instance, are objects. When an enemy dies, it drops a weapon. Those weapons must be

randomly generated. Here are the instructions to observe:

- First, weapons will have to look different. The weapon model must also be randomly picked in a list of preset models.
- Each weapon will have to make a minimum damage and an attack speed, but feel free to add the stats you like.
- The weapon's stats will be randomly generated but not anyhow. You will have to take into account the current player's level in order to influence the weapon generation and its stats values. The weapons must not be "broken", that is: they must not be significantly stronger than the player's and enemies' strength and wipe enemies too easily.
- Weapons must have a tooltip that appears when the mouse's cursor passes on it, whether they're on the ground or in the inventory. This tooltip must sum up the weapon's stats.
- You will also create a rarity level. Some weapons must be way harder to get than others. Apply those ranks for instance: Common, Uncommon, Mythic, Legendary.

That's not all, folks! It's cool to have a weapon generator, but it's better if you can equip them. That's why you will need an inventory to carry them. You will create a system that allows the player to pick up the weapons left clicking them. These weapons then appear in an inventory window you can access pressing the "I" key. Once in this inventory, you can organize the weapons as you will and change the current equipped weapon. You can also throw away a weapon dragging and dropping it out of the inventory. Of course, the inventory will have limited storage availability (12 boxes sound like a fine limit).

V.4 A bit of variety

Now the gameplay is operational, it's time to get creative! You will create an interesting map in the wide open and a dungeon with several floors (let's say 3). The last floor of this dungeon will feature a boss with a LOT of life points, different attack skills (you can use some of the hero's skills to gain some time) that will inflict TONS of damage... Anything to make this EPIC battle worthy of the player's precious time!

You're completely free, here. Enjoy and have fun!

V.5 The soundtrack

To make your game even more immersive, you will need a soundtrack. Once again, you're free to pick the tracks and sounds you like to make your game as stylish as possible. Just try to consistent. Funky music and spring sounds on attacks might not be the best choice for this kind of game - unless you took a very personal creative path... and why not! Anyway, it's also time to test the infamous reverb zones we mentioned earlier this week.

V.6 A cheat code to rule them all

The point of a Hack and Slash is to progress through various levels. However, to help the evaluation and allow the assessor to test every required implementation, you will set up a cheat mode that helps level up the hero instantly and spawn a weapon with the hero's level. Your game must be balanced on every level, 1, 20 or 50. That's why the assessors will test the game with different hero levels.

Chapter VI

Bonus part

If you have achieved everything that was required here and still have time to kill, you're `NUTS!`

But if you really want to go further, here are a few bonus ideas...

- Objects you can loot other than weapons. Shields, helmets, boots, rings, amulets, candies, pets...
- Different character classes.
- Range weapons.
- Elementary effects on weapons, like fire or ice, and stats to go with it.
- Money management: enemies drop credits that allow the player to buy stuff from NPC's.
- A title screen with a saving system keeping the player's progress in memory so they can resume a new game with their level and gear.
- Implement "Butt Stallion".