Game Dev Diary – Luisa

Date	Entry

24.04.2019	• Created Unity project (without real plan) and uploaded to GitHub
	• Found models for prison/dungeon scene in Asset Store and created first level version with it. However, the only thing I could find was a medieval dungeon, no modern prison setting.
	• Added post processing to the camera to make it look interesting/different
	• Discussion with Tim:
	  ○ Agreed to create a brawler with cute characters
	  ○ Set in prison, as both characters are criminals
	  ○ Cute controls? Wonky character movement?

26.04.2019	• Tim added character models → knights to fit dungeon/prison setting instead of modern criminals • Also added animations that really fit "cute" and "chubby" characters • First version of PlayerControl & HealthSystem look great so far • Only small changes on camera
27.04.2019	• Project clean up • Merge conflict yesterday on scene → separated test scenes to avoid this now! • Bug found: Weapons decided to inflict damage via OnCollisionEnter. This triggered as well when not attacking and only touching another player with weapon. FIX: Weapon now knows via Boolean if it's attacking. Boolean is set and reset again, weapon does not inflict damage when false. • Tim added more animations for kick & knockdown • Added health bars to characters to visualize damage in combat • Bug found: Health bars show that sometimes the player does "double-damage" to their opponent when only attacking once. Weird behavior that only happens sometimes and shouldn't be happening. Needs more investigation. • Tim also added functionality for knocking down enemies, picking up weapons and unarmed fighting → good progress in game control
28.04.2019	• Tim added first version of multiplayer and some more animations • Changed skybox to get rid of the standard unity one in background. Galaxy for now but that doesn't fit quite right to the setting.
29.04.2019	• Now focusing on intro and outro: Added first version "FIGHT!" before and "PLAYER X WON" after battle, always fading in and out with a blackscreen to give the gameplay a context to take place in • Fading to black allows for easy scene transitions without the player noticing too much: Added main menu scene that entices the player to press a button to begin the game • Main menu setup rotates around itself to give the impression of a rotating background (the skybox) inside the menu. This makes it look more interesting than static background. • No idea for a title or real "story" yet, placeholder text must be enough for now, to be displayed once the player agrees to start and before he presses button again to go to the game scene.

	• Also added auto-return to main menu after game end to make instant restarts possible • Small improvement: Players now display their player number to make them easier to distinguish in battle • TMPro Standard font does not fit the medieval setting – found better font online, created character atlas and changed it everywhere in game. Looks much better • Bug found: Input mapping flawed, both players move via one controller • FIX: Tim found the solution, both input settings were getting input from every controller axis, they need to specify the controller number explicitly
01.05.2019	• Added intro text as story to be displayed after starting the game in main menu → short text on a scroll. Placeholder text so far, still no great idea • Since Tim fixed controller input now, the game can easily be started by controller button press, making the ui buttons redundant (will keep them around just in case) • Some project and script clean up • Updated the main scene: placed more objects to make it more interesting to play in • Bug found: Players were able to pick up a weapon from the hands of a player that was already using it. Found that one by accident when attacking P2 and it somehow repeated the pickup animation I did earlier and just took my weapon :D • FIX: Weapon now knows if it's already in usage via boolean. Can only be picked up if not
02.05.2019	• Bug found: The "pickup in process" Boolean was not reset correctly, only once a weapon was picked up. That meant that a player could do the pickup animation without touching a weapon, then run up to a weapon and immediately pick it up once touching it without doing the animation again • FIX: the Boolean is now reset after a short time, regardless of whether a weapon was actually picked up or not • Thought I fixed the double hits, but they keep happening, wtf • To make the game a bit more scripted and give the avatars more live, the winning player now automatically taunts on game end. Also, both players do the blocking motion to signal that they're ready on game start • Bug found: The player character still moved during pickup, taunt, getting hit or blocking animation • FIX: read current animator state and return from Update() before movement to stop the player from moving, if they're doing an action that requires them to stand still • Also added more weapons to the scene, to test the weapon blocking that Tim added • We want sound! Found two (mostly) fitting tracks for the game scene (very cool) and the menu (sound a bit like elevator music but it shall do for now) • The music also fades in/out on scene changing to complement the blackscreen that comes up as well • Tim worked a lot on animations to fix a lot of bugs that came up when I tested multiplayer with Michel, like knocked down players standing back up immediately because something went wrong with the transition. I have no full idea of all his changes (mostly transition exit times I was told) but the bugs seem gone now

	• Tim also fixed the catapult bug → players that are hit can no longer be catapulted into the air, the y-axis is now set to 0 via constraint • To be ready for the presentation tomorrow, I also winged it and wrote a short intro text to be displayed on the scroll. The story is not as elaborate as we once wanted, but it's fine. We also have a title now! Random game name generators only came up with useless ideas, so I was stuck on that, but somehow, I came up with "The winner takes it all" which has a nice ring to it. Guess we'll keep it for now. • Another FIX that is long overdue: The ground now has a gigantic collider to avoid weapons from accidentally being pushed through it and dropping into the eternal abyss. Same goes for the walls. • I also gave the players more lives to make the game last longer and to kind of counter the random double hits that still keep happening, but that doesn't solve the problem. Needs more investigation.
06.05.2019	• The presentation went fine, but of course the double hits kept happening all the time :D • We were told the give our game more "character", to make the player characters unique and help the player identify themselves with them. Therefore, we want to make the characters clumsier, by letting them stumble and fall when running against a wall for example. Not sure how much we're going to be able to achieve until Friday, but let's see • In the presentation, the players would often just stand in front of each other and keep hitting until one of them died. Blocking did not end up being useful yet, so now I added that successfully blocking an attack knocks down the attacker, giving the blocking player a concrete advantage by doing so. I'm curious if next Friday, players will maybe start blocking more often? • Also, a big change I made: our camera used to be static, which was not helpful to engage the player into the game, draw them in and keep them interested. Therefore, our camera now moves. Since we have all players (just 2 right now, but the system would support more) on the same screen, we cannot simply let each player control the camera movement. This is why our camera now calculates its position depending on the player positions. If the players are close together (maybe to engage in combat), the camera draws close as well to make the combat feel more personal and more thrilling. If the players run away from each other, maybe to try and find weapons on the map, the camera zooms out to ensure that all players are visible on camera at all time. I achieved this by calculating the point that's most in the middle for all players (all players have the shortest distance possible to it), called centroid. Simply calculating the middle would work for 2 players, but since we might plan on supporting more, that was the way to go. The camera now moves with this point and always looks at it. Also, I calculate the maximum distance between two of all players and use it to determine the necessary camera height and distance to get all of them on camera. Then, I lerp the camera position accordingly. So far, it works really great with two players, testing it with several players still needs to be done. APPENDIX: More detailed explanation and sketches for this can be found in the appendix. • I also changed the skybox to be more realistic and fitting, I guess a galaxy in the background wasn't the best choice for a medieval setting :D
07.05.2019	• It seems like Tim figured out animation events (I always keep forgetting how to set these up…) and used them to fix the annoying double hits that

	kept happening. We can now precisely set the attack Boolean on each weapon to only be true for a very short amount of time and in the right animation frame. Seems like this is finally FIXed!
08.05.2019	• Tim and I agreed that we wanted to implement a stamina system to ensure that players would need to plan their combat and could not just mindlessly attack each other while standing still. Tim mostly took care of it, I'll contribute tomorrow.
09.05.2019	• Stamina system works great, I cleaned up the scripts a bit and added stamina bars to the UI, now we're done here • I tried to make players get knocked back from walls, and it works as intended, but our current animation really doesn't convey the feeling of a clumsy character that falls down, he simply walks backwards a bit :/ Sadly we don't have enough time to find a new animation or create one ourselves, so this will have to do for now. • To make up for the disappointing bumping animation, I gave all scene objects rigidbodys so they can now be kicked across the level, which is really fun ^^ • Since we have candles with particle systems as fire, I took some time to make sure they would not only fly around when kicked but also go out instead of simply continuing to burn on the ground. Let's see if anyone notices :D
15.05.2019	• Nobody noticed the candles ☹ • To make our game feel more responsive as feedback wanted, I started adding sound effects to our characters. Some of the effects are called via script, others should be triggered via animation events to ensure that they are correctly synchronized • I'll leave most of the animation events to Tim as I don't want to mess with his stuff. I wrote the functions and supplied the sound effects, he just has to call them from inside the animations • Also, we now have a new wall bump animation – looks great, but I still think it isn't responsive enough and rather annoys the players… • Same goes for the knockdown animation: Somehow it takes forever for a knocked down character to actually start their animation. Weird, definitely a BUG.
16.05.2019	• FIX: Found the reason why knockdown wasn't immediate. The transition from idle to knockdown animation was told to have exit time, therefore the idle animation first finished a full circle before transitioning to knockdown. I changed that, so it's fine now
05.06.2019	• Did not get to work for a long time, many other deadlines :/ • But I thought about the decision process in our game, came up with some more decisions and documented it all, so that should be good for now • First thing I added: it is now possible to use the last bit of stamina for a final action, even if it actually isn't enough. However, this results in a temporary block for the player: They must wait until stamina recharged fully before being allowed to choose an action again • This results in a decision between having one more successive action at the cost of being incapacitated for a longer time, and not taking that one bonus action and not being forced to wait that long • Since Tim implemented different types of weapons (which also create another decision), I included them into the main scene and increased the players health to allow for longer combat

	• I also finally got around to fixing the Z fighting on the level ground, so that looks better now • FIX: To prevent the others from catapulting objects out of the level again (Looking at you Michel :D), I surrounded the whole level with colliders. Yes, even the ceiling. • I identified one source of information starvation in our game: The player had a bit of a hard time to locate weapons in the level and actually understand that they can be picked up. Therefore, all weapons now have a particle system that glows a bit as long as they are not picked up. This makes them easily identifiable
06.06.19	• To make the blocked recharging process even worse, I decreased the stamina recharge value during that time to slow it down even more. I also increased the time the player has to wait until the recharging process starts when a block is active. This will need to be further balanced but feels good so far. • FIX: I finally got around to fixing the sound bug in the main menu, so the other won't abuse that anymore :D Now the confirm sound will only play once for each button press, holding down the button will not loop it. How? Booleans of course :D • I also changed the icon color of the swords in the menu, black on black cannot really be seen • One final addition to making the weapons feel more distinct and different, but balanced: the weapons now manipulate the animation speed of the hit animation using a parameter in the animator component as multiplier. When a weapon is picked up and registers its player, it also sets this multiplier value to their own one. When being dropped and deregistering the player, they reset it to 1. Now, the small dagger is really fast, while the large hammer is slow. The slow version of the animation looks a bit ridiculous, so we might have to look into that at some point • Tim added a scene that includes a canvas depicting the controls. Took that and added it into the already existing main menu to be enabled/disabled when required, this works better/more fluent that switching scenes just for the controls to be displayed on a canvas. Also added png images of xbox controller buttons to the canvas to make it easier understandable
09.06.19	• To help the players better distinguish their avatars during the fight, I created two separate materials for the character models and changed their colors. Due to our post processing on the camera, the colors do not look exactly as planned, but they look different and that is all that we need. I made a few changes on the post processing behavior to make the different colors a bit more distinguishable, too. • I also removed the bounce back script on the walls, since that was incredibly annoying to the players and did not do much to make the game feel better. Having the avatars interact with the world through the object physics works a lot better than interrupting their movement at the walls through collision and falling down. • The block is now buffed: Whenever an attack was successfully blocked, the attacker is no longer kicked down as that is already handled (and a lot easier and faster) by the normal kick. Instead, a successful block now lets the player drop their weapon, which makes them vulnerable to counter attacks and forces them to stop and pick it up again. Now, the block is much more useful and powerful when used correctly, making it an actual

	valuable choice to consider during combat, but still requiring correct timing from the player to work out. • The trap door we had in the level since ages ago now actually opens and closes in random intervals. The players can now drop into it and die there. Bug found: However, to do that, I had to unlock the y-axis transform constraint that was on the players to stop them from catapulting into the air. We'll have to find another way to prevent that from happening. • Also, as Jonas suggested, I added dragon heads of stone onto the walls. These now spit fire across the map in random intervals and in random patterns (no head, some heads, all of them). Through particle collision, the player receives damage from the fire, therefore restricting the battlefield during these intervals. APPENDIX: More detailed description of the dragon head mechanics in the appendix.
13.06.19	• I wanted to improve the trap door and dragon fire mechanics that I added a few days ago, therefore I added sound effects to both of them. • Also, the camera now rumbles while the dragons spit fire, making it feel a lot more devastating and dangerous. Now the player will definitely notice it, giving the danger some weight. • I added colliders to the dragon heads, but then had to notice that the fire particles were blocked that way, totally ruining the fire effect I had going on. So now, the dragon heads have a few colliders that stop the player from running through it but leave a hole for the fire to get out. • I tried some things to stop the catapulting bug from happening but was not extremely successful. The players now have a high weight, which makes them catapult a lot less easy and not that high. This must suffice for now, I'll need to investigate that further.
	•

Appendix

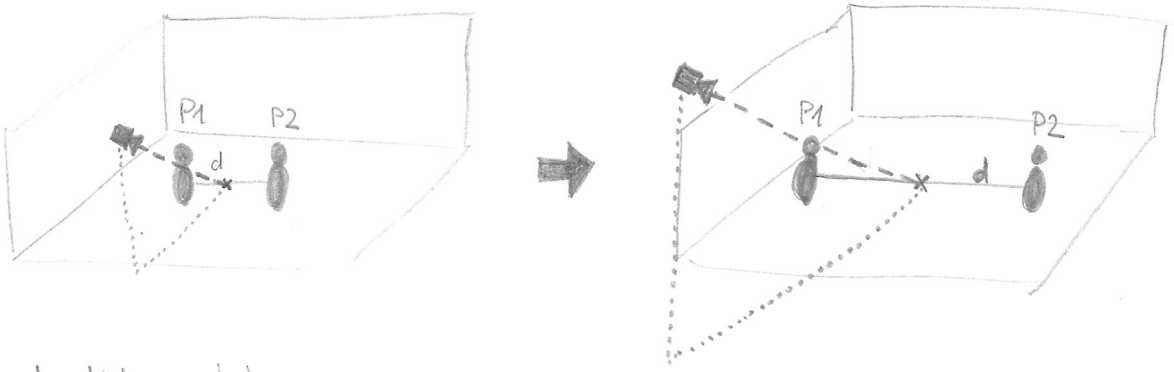
A) Camera Movement

B) Dragon Head Coordination

GamesSpin Project
-Camera Movement-

1. Let camera look at & follow the centroid between all alive players
2. Calculate the greatest distance between two players and lerp camera distance to the action according to that
   → players close together = camera draws closer
   → players far apart = camera moves back to fit all on screen

Example, 2 players alive



d = distance between players

× = centroid, the central point between them

- - - = camera looks at centroid...

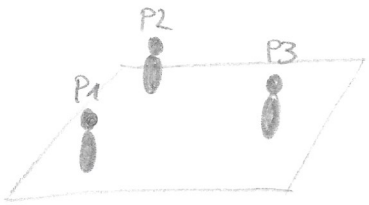
..... = ...while following it in a height / distance that depends on d

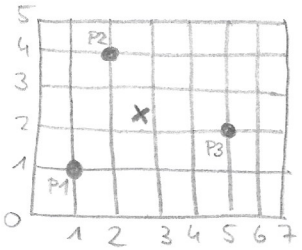
The players moved further apart, d increased
→ the camera is still looking at the centroid and following it at a certain distance and height.
→ But since d increased, this distance & height have been lerped closer to their max. value, increasing the overall distance of the camera to the action to still have all players inside the camera view.
→ Min/Max value for camera distance/height have been found through experimentation

Auszeichnung Initiative Engagement Mentor WIR-Gefühl Stipendium Auszeichnung Leistung Projekt Netzwerk Türöffner Neugier Mentor Neugier Orientierung WIR Chancen Freiräume Praktikum Talente Potential

Calculation A: How to calculate the centroid

→ centroid = where the balance point would be if all objects (= players) weighed the same amount & were standing on a weightless plane





P1 = (1,1)
P2 = (2,4)
P3 = (5,2)
n = 3

$$x = \text{Centroid} = \left(\frac{(x_1+x_2+\dots+x_n)}{n}, \frac{(y_1+y_2+\dots+y_n)}{n}\right) = \left(\frac{(1+2+5)}{3}, \frac{(1+4+2)}{3}\right)$$

$$= \left(\frac{8}{3}, \frac{7}{3}\right) = (2{,}667, 2{,}334)$$

Calculation B: How to calculate the greatest distance between the players

→ Take each player individually & calculate their distance to each other player on the x and z axis (no difference in y possible)

→ compare that to the saved values for max. distance on x and z
- if that value is bigger than the saved one ⇒ overwrite!

→ use the determined vector of (maxX, 0, maxY) to modify the camera distance and height
= lerp these values according to the magnitude of the determined vector.

Gamespin Project
-Dragonhead coordination-

(1) How to determine when to start action:

a) random value (between min. duration between actions & max. duration between actions) = new timer until next action
Start() - min./max. duration set in script, can be modified

b) compare the difference between time since level loaded and time of the last action to the current timer
Update() - if time since last action exceeds current timer → start action

(2) How to determine which dragon heads to activate:

- each dragon head has a chance to start breathing fire (50% atm)
- roll random for each head: if roll is below fireChance (below 0,5), then activate this dragon head
- if at least one dragon head was activated
  → tell camera to rumble for that duration
- set the time of the last action to the current time
- find a new random value as the next timer until next action

(1) if ((TimeSinceLevelLoaded − timerLastAction) > currentRandomTimer)
→ start action

(2) timerLastAction = TimeSinceLevelLoaded
currentRandomTimer = Random(actionMin, actionMax)