Individual Project Documentation

08th of July, 2018
Jule Randzio
Immatrikulation-Nr.: 753172
Project 99+

Team Members:
Samuel Rassy
David Sarbayeyev
Lazaro Beckmann
Julian Roehrig

Professors:
Carla Heinzel
Thomas Nickel

Content:

1 Project description:

Our PvP-Stealth-Action-Party-Game in topdown perspective is called 99+. In the game you take the role of one out of two characters: Mr. Ishikawa, an ex-military and very strucutured person, or some guy called the dude, whos I-don´t care-attitude is in no way compatible with his opponents lifestyle. They both enjoy their retirement in a nursery home which should provide a relaxing environment for their last days. However their differences lead to a miniature warfare, using whatever they can find in the nursing home and trying everything to get the other to stoop so low in the eyes of the other inhabitants that they ultimately have to leave the nursing home. Your objective is to gather items on the map, craft a stronger weapon out of it, turn you opponents radio on and avoid being seen while doing mischivious things. Damage his reputation as much as possible before your own gets irrepairably low.

2 Individual team role and contributions:

Because of a misunderstanding I got into a project I didn´t really want to do so when a swap became possible which led me to choose to code all by myself or do a project I wasn´t into, I chose the challenge. That included all the programming aspects as well as implementing almost everything except for the map assets in Unity. Additionally I also became the projectmanager for this game. To complete my design parts that were required I did most of the UI designs myself and last but not least the gameplay trailer as well as the teaser trailer.

3 Scope & methods:
Development:

After the first week our whole game idea needed to change since we couldn´t narrow it down to a logline and the complexity was way too high. During that time of concept finding I experimented a lot with Unity since I had no real experience and to spare me the time later on in the project. I experimented a lot with cameras and their behaviour. I coded a 2.5D Follow Cam which was later overthrown. We then changed the core game completely.

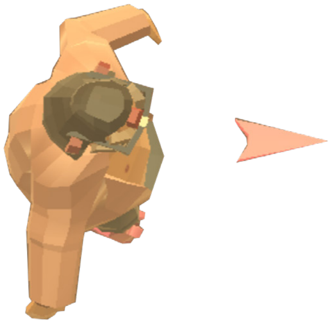
Networking:

Originally we had lots of ideas for game mechanics which set the scope we were going for again way too high. During the whole project we constantly needed to reasses what was really needed and what was possible in that short time period. We also wanted to implement networking so that people could play together over a server and not just on one computer.

So I started working myself into networking as early as possible. That cost me almost four weeks of time rather early in the project which hit me hard since it didn´t work out in the end. I got the basics running but it wasn´t fully my own code so at some point I didn´t understand what was going on enough anymore to fix the problems I was facing. I spent almost three of those four weeks on bug fixing and trying to get some basic features to work. I used Photon for networking and set it up using the Guidelines they provided. I got the concept of clients and such however I was struggling to get everything syncronised on all client PCs. Eventually we took a step back and decided to go for a splitscreen game as I could progress on such a game further and way quicker.

Movement:

I really made a lot more progress suddenly as I wrote my own code from scratch. I started setting up the movement for both a keypad together with a mouse and an Xbox One Controller. The walking direction and rotation is separated from the shooting direction. For the first player the direction input comes through the keypad (regular wasd-input) and for the controller with the left joystick. Shooting is then controlled by the mouse position on screen via a raycast to set that direction-vector seperatly from the direction the player is facing. For the controller the direction vector is generated through the right joystick input. To make the shooting direction clearer to the player, an arrow rotates in a circle around the palyer, indicating the shooting direction. It uses the same input as the shooting mechanic itself.



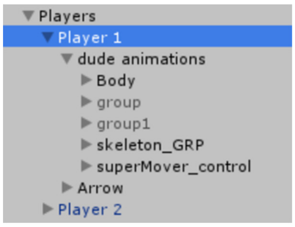
On button input the player can sprint for a certain amount of time after which he needs time to recover since they are old people. The conditions checks, wether the cooldown timer has reached zero and then again wether the player has been sprinting for a certain amount if time, after which the walking speed (as well as the animation) is set to normal speed again.

Camera:

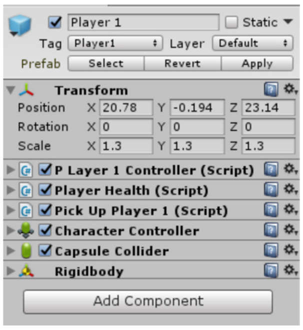
The two topdown cameras I used are always following the players in a constant distance with them in the middle of their half of the screen. Here I needed to make sure that the mouse input took the right camera into account.

Player Set up:

Since the player is able to pick up things and carry them around visibly, the player is set up as follows: under an empty game object which holds most of the scripts the player fbx is located, to make it exchangable. Then under each last hand joint of the rig is a variety of objects that they can carry, all set inactive. They are then stored in a list in each script which needs to access them. The player has only got three scripts directly onto him: A player controller which handles movement and most actions, a pick up script, and a health class.



```
▼ Players
  ▼ Player 1
    ▼ dude animations
      ▶ Body
      ▶ group
      ▶ group1
      ▶ skeleton_GRP
      ▶ superMover_control
    ▶ Arrow
  ▶ Player 2
```



☑ Player 1 □ Static
Tag: Player1 Layer: Default
Prefab: Select | Revert | Apply

Transform
Position X 20.78 Y -0.194 Z 23.14
Rotation X 0 Y 0 Z 0
Scale X 1.3 Y 1.3 Z 1.3

☑ P Layer 1 Controller (Script)
☑ Player Health (Script)
☑ Pick Up Player 1 (Script)
☑ Character Controller
☑ Capsule Collider
Rigidbody

Pick Up:

The whole game is based on a naming system to verify each item that is able to be picked up. At each point where a new game object needs to be instantiated, picked up or swapped, the compiler checks for either the name but mostly the tag, the item has got. Almost every dependency is handled by observer. Each players inventory consists of 2 slots basically: his left and right hand. If he wants to pick up something else than what he has already got, it needs to be swapped.

Spawn points:

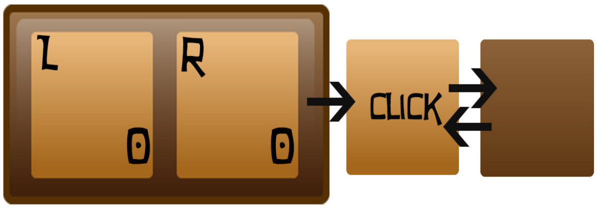
For the pick up ability I set up spawn points all over the map which instantiate a random item out of the pool of items we have got. Each spawn point has a trigger zone to activate the UI which tells the player that he can pick something up (X or A for controller, left mouse button/ right mouse button for the mouse). It is activated on each trigger enter as long as there is an item to be picked up in the spawn location. On pick up the spawn point gets deactivated.

Spawn Objects:

Each spawn object then has its own trigger zone for pickup to minimize dependencies. The players pick up script checks on entering the zone wether the tag of the object on the ground is the same as one of the players children in the list with pickable objects. If so, the hand held object is temporarily stored in an array for the left and right hand.
If now the player is still in the zone and decides to pick the object up on button press the object saved in the array is set active, the one on the ground destroyed and the spawn point set inactive as well. If the player has already got one object in its hand, and chooses to still pick up another object, the old one in that specific hand will disappear.

Shots:

Each object can have a specific amount of shots that the player can fire. They are also set in the pick up script of the player and stored in a variable for each hand. They get updated in the player controller as the pick up script is only for setting the conditions. The UI is also notified via Observer.



Shooting:

The shooting itself is triggered in the player controller script via mouse or controller input that differs for each hand. Each hand object has a fire controller scipt attached to it that stores a firing point which in this case is the hand of the player.
In the Update function of the player each frame is checked wether there is an active weapon in each hand. If so, it gets set as a firecontroller left or right depending on which hand has picked it up.
If the fireobject is null for either of both hands, firing is not possible for that hand. If the fireobject is not null then gets checked, wether the object is shootable or not. If it is shootable, the the shooting function gets called. They differ for each hand.

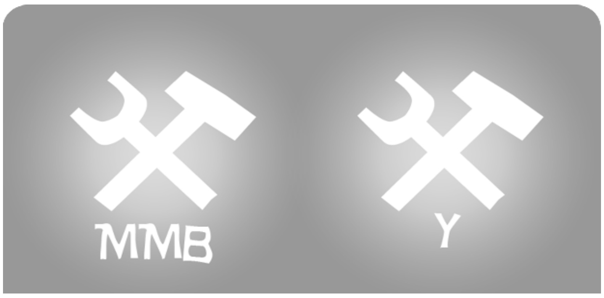
An animation gets triggered and the firecontroller script gets notfied that it should shoot now. The owner of the projectile is transferred as well so that one can not shoot himself. The firecontroller then checks the shooting direction and initialises a new projectile via the objectcontroller function "InitProjectile". It sets the speed, direction and ownership of the projectile. The type of projectile that is fired is set in the inspector for each firecontroller differently.v

Out of shots:

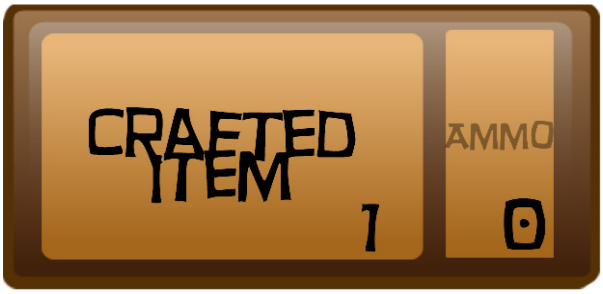
As soon as the players shot count for an object reaches zero, the weapon gets reset, the fireobject is null again and the shotcount zero. The UI is notified by this as well.

Crafting:

If two specific objects are picked up and active at the same time, the player gets notified in the UI that he can craft now by pressing the middle mouse button or the B-Button on the controller.



The two carried objects get then set inactive and a new specific Object gets set active. The player can carry it only with both hands, if he wishes to pick another regular object up he needs to give up the crafted one. The crafted objects has no shots by default the player needs to collect ammunition which is spread all over the map. Unlike the regular objects, the crafted one does not disappear as soon as its empty. You can carry it as long as you wish and only get rid of it by swapping. You can damage you opponent harder with the crafted weapons. In all other aspects it works just like the normal weapons.

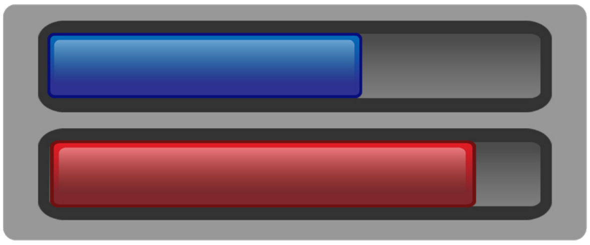


Ammunition:

Just like the pick up objects the ammunition is spred via spawn points and on start instantiated. The only difference is that they can´t be picked up if the player hasn´t got a crafted weapon and if he has one, they will immediately added to the inventory.

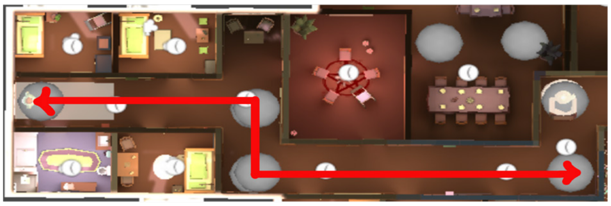
Health:

The health class changes the players stats. On trigger enter, if the circumstances are correct, a large value is substracted from the current health. If a special state is entered, each frame a minimal amount is substracted, until the state changes again. The health bars color changes on each value change for a split second to emphasize somethings happened.

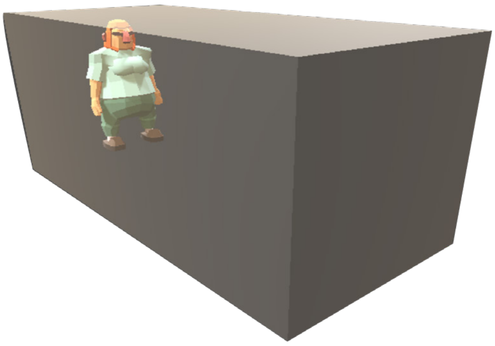


Nurse:

Nurses patrol all over the map in certain predefined paths. They have a path script attached in which you can enter any amount of waypoints (empty game objects) you like. They will iterate at a constant speed through the order given in the list in the inspector.



They also have a field of vision which you should avoid while doing something wrong, such as sprinting, being dirty or throwing something. If you´re only carrying something, the nurses won´t do anything. If you do one of the before listed things, wether ontrigger enter or stay, they will take your stuff away and you need to find or craft something new. Also your reputation lowers big time in that case.



NPCs:

There are three types of NPCs: standing, sitting or walking. They move in a slow pace same as the nurses, but they will stop in their tracks as you pass their much smaller trigger zone and perform a rant animation before they return to their normal idle or walk. The damage you get from disturbing them is the same as from the nurses however it is much more unlikely to disturb them since their vision field is much smaller, since they are old.

Bathrooms:

Once you have been hit you´re constantly loosing minimal amount of reputation until you´ve cleaned yourself in a bathroom. On enter a shower sound is triggered as well as the stopping of health reduction.

Radios:

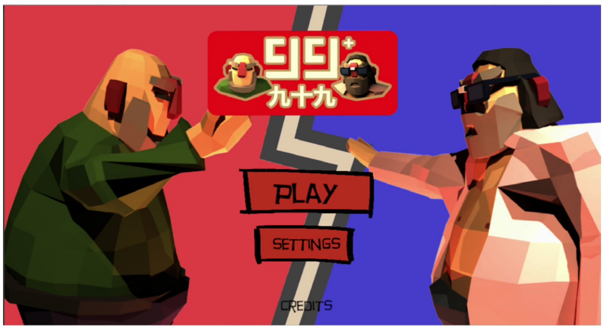
In each players room is a radio with music that is specific for each character. If the opponent enters the room a UI is triggered and with a specific key in the trigger zone a game event is triggered that activates a sound and starts the constant reputation reduction in the players health script via Observer.

UI:

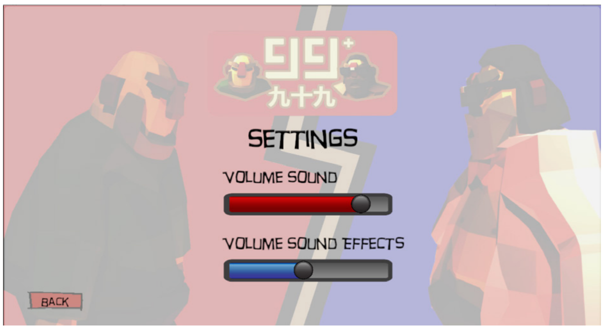
Our UI is kept very simple. It consists only of the players reputation bar and the two slots for his items to represent them more clearly. On pick up a comparison is made, wether the sprites tag is equal to the active objects tag. If so, it gets set active. Also the shotcount is always updated in the corresponding slot as well. If an object is crafted the two slots are swapped with a larger slot to represent the stronger weapon and a slimmer slot with a faded out ammunition slot. If ammunition is picked up, by entering the trigger zone, it gets full opacity and a counter. The health bars represent the percentage of reputation the player has currently left and undermine visually the moment he is damaged by a sprite swap for a split second to the other players color, to give the player an indication that he is damaged.

Menu:

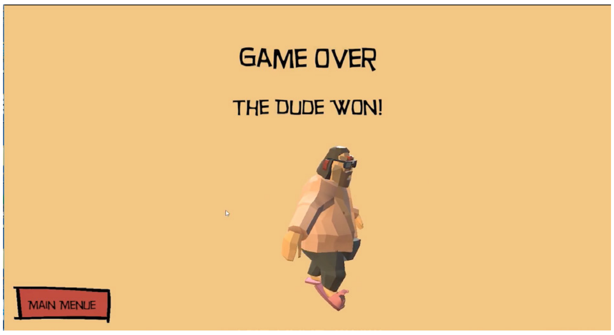
Our Menu consists of three sites basically: The main menu, the settings menu and the credits.



It is important to mention that instead of loading a new scene on pressing the play button i opted for only one scene and setting the unused canvases inactive on button click so I didn´t have to code functions for this and instead used the Unity built in options in the button script to set game objects inactive without a separate script needed. Each menue therefore has a back button, which sets the current canvas inactive and the main menu active. To each button is a sound connected as well as a hover and click effect.



In the settings menu there are two volume sliders which manipulate the regular background music on one hand and the SFX sounds on the other hand. They are directly connected to the Audio Manager. As a background for the Menus I used two videos: One starting animation, and after that one that is looped constanly to make the menu more alive and not setting key frame animations in Unity itself. On pressing play, the whole UI and its needed camera and video manager are set inactive, the in game UI is set active and the main game sounds and background music are triggered. If one player looses, the player health scripts sets the menu and camera active again and notifies it to activate the corresponding game over screen depending on who won.



In there a 3D model of the winning character rotates which is why I needed the camera overlay, or else the model wouldn´t have been visible on the canvas. On main menu click here, the scene gets reloaded to make a new round possible.

Sound:

To manage the sound I wrote a Sound Manager which makes a List that uses a Sound Class like an interface. You can add any sound in the inspector and vary the length of the list. Each sound is identified by its unique name and can be called via a play function that iterates through the list and checks for the same name, then plays it.
The sound manager gets notified all the time via Observer when something happens and plays the corresponding sound effect. It has a main game function which randomly plays varying snippets of background music in a constant loop.

Animations:

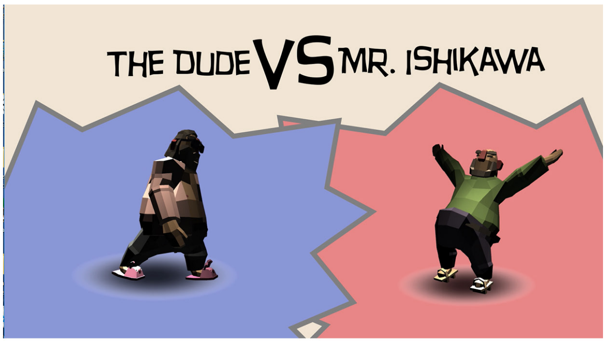
I implemented an Idle, walking, sprinting and throwing animation for the two main characters as well as a special craft shot. The nurses only hava a walking animation and the NPC was delivered to me with lots of different animations which I then split up and let them Idle around in either sitting, standing or walking state. Only on trigger enter they would break out into a rant and then continue idling.

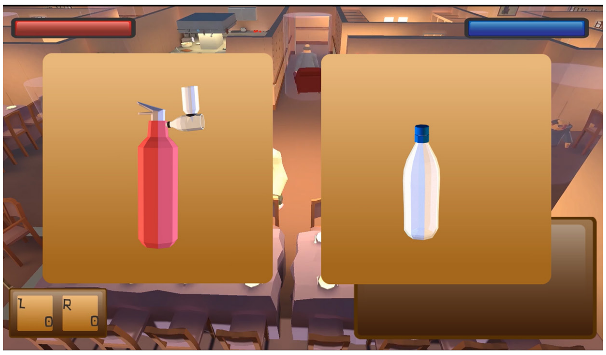
Design:
UI:

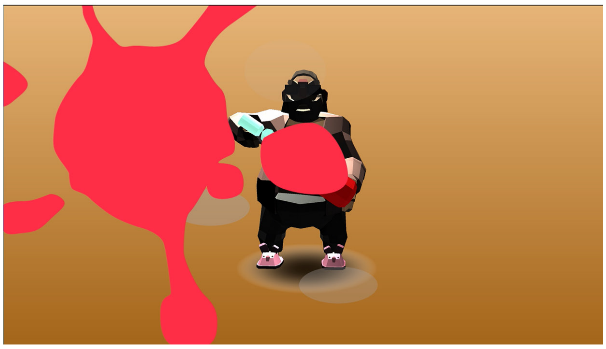
I did the background video for the starting menu in After Effects using keyframe animations and render that my artists delivered to me. I added some simple assets made in illustrator as backgrounds and clouds and sounds from our Soundie. The buttons for the menu were made by Sam by my directions and layout. The background of the credits is a render of our environment assets which I requested from Lazaro. The in game health bars vary in color to represent each player. I wanted to keep the inventory slots as plain as possible without being too boring so I added some shades and varying layouts for normal objects and crafted ones. All of the in game UI was designed and made in Illustrator. The sprites of the objects were again done by Sam after my design.

Teaser Trailer:

I developed a storyboard for our trailer. I used part of the menue video and expanded it to a total of 48 seconds and added some more effects and transitions. In there are also some in game scenes which I rendered in realtime and captured them directly out of the game. Compositing and Cutting was also my part.







PPM:

We tried to use Trello for organizing parts but quickly let that slip. We communicated a lot instead and wrote schedules on our whiteboard for everybody to see. Even though we were together most of the time in the project room anyaway, we discovered later on during the project that having discord open and working silently most of the time besides each other while at home was a great help at any time some issue came up. I checked moodle almost on a daily basis (sometimes twice or more a day) and kept everyone on track in our group chat on whatsapp. I also made sure everybody was there for coachings and meetings we decided on. I structured the way we exchanged files and data via a server that I organised however we still needed to use google drive since one team member didn´t have his own access on the work station and could therefore not download the client he needed. The structure I provided the guys with was very helpful as we separated materials from active files and final products so we could always find easily what we were looking for. We always met up the day before coaching to assess again what we had to show and how we would present it so we´d always be prepared.

4 Changes to the original IPAM:

My planned scope from the original IPAM was pretty close to the final outcome: coding was easily to plan ahead since it was solely my own task and everything has come together as foreseen. My PPM part as project manager was also the exact scope I expected. Only my design contribution changed during the game. I was so focused on coding that I left the game and level design mostly to the artists. I was always backchecking with them and giving feedback on what I thought they should keep in mind but I was not the driving force behind this. Instead I did the layout, design and functionality for the menu and the in game UI. That was good for me as I also had to implement it in the end and could set it up exactly as I needed them to work out funcionality wise. For example some things needed to be layered and in a certain size relation. Since I did all this myself there needed to be no briefing nor iteration.

5 List of Items:
Scripts:

Sound
AudioManager
VideoManager
GameEvent
GameEventManager

ObjectController
PickUpObjects
SpawnObjects
AmmoCount
AmmoSpawn

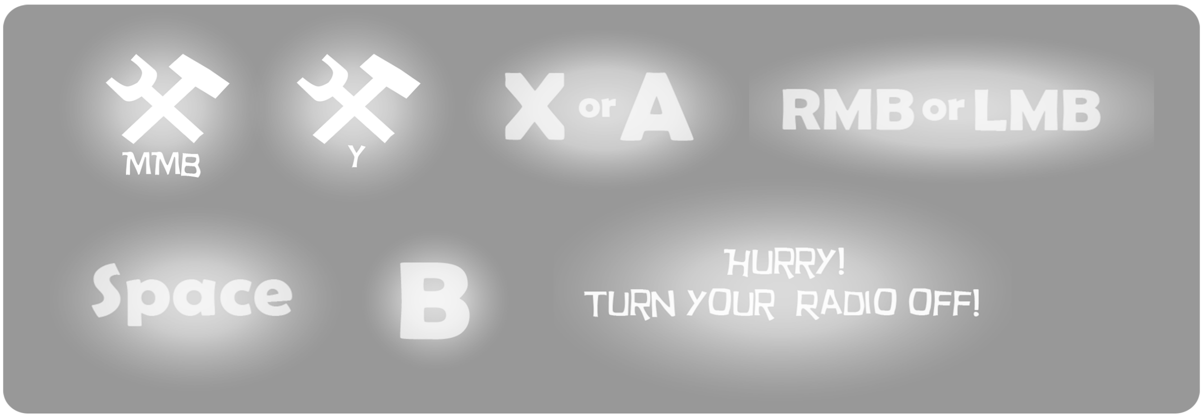
Bar
UI Menue
UI Trigger
UI Script

CameraFollow
Splitscreen

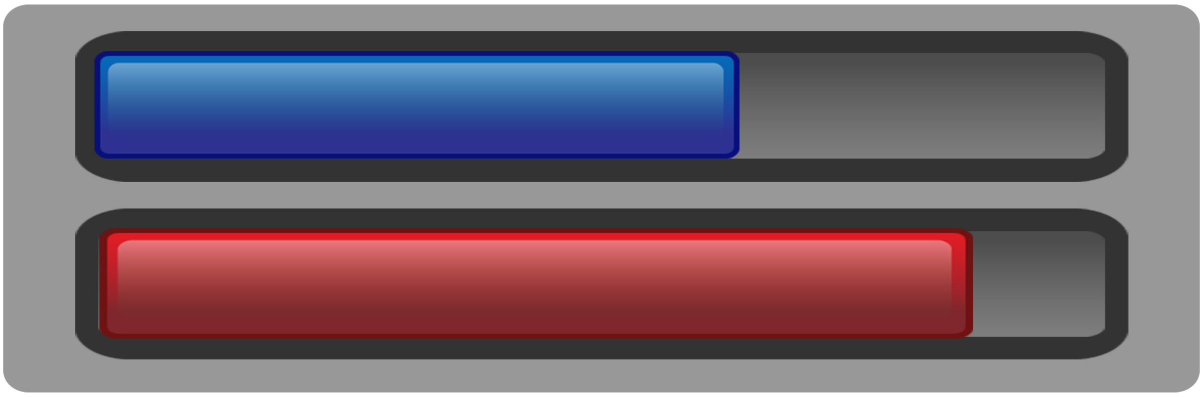
FireController
PickUp
PickUpPlayer1
Player1Controller
Player2Controller
PlayerHealth
Arrow
Stats

NPC
NPCTrigger
NPCSitting
NPCStanding
PathFollower
Rotate
Nurse

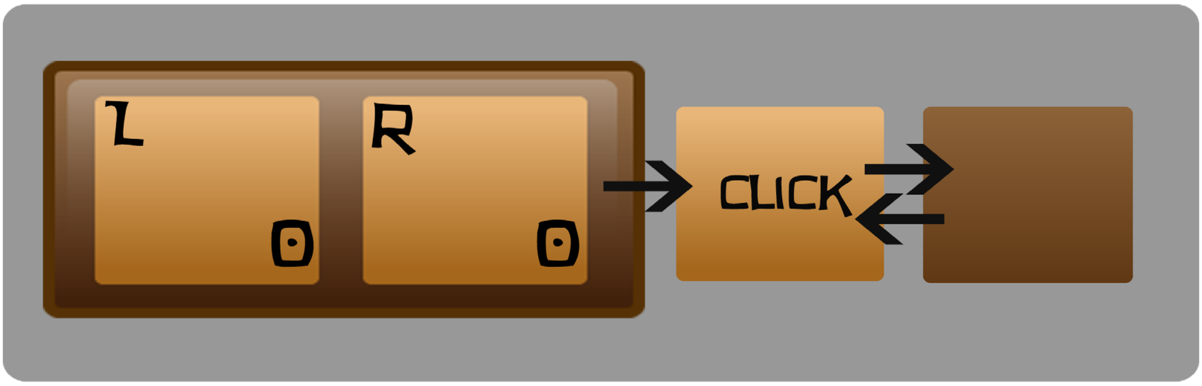
Assets:
Notification Visuals



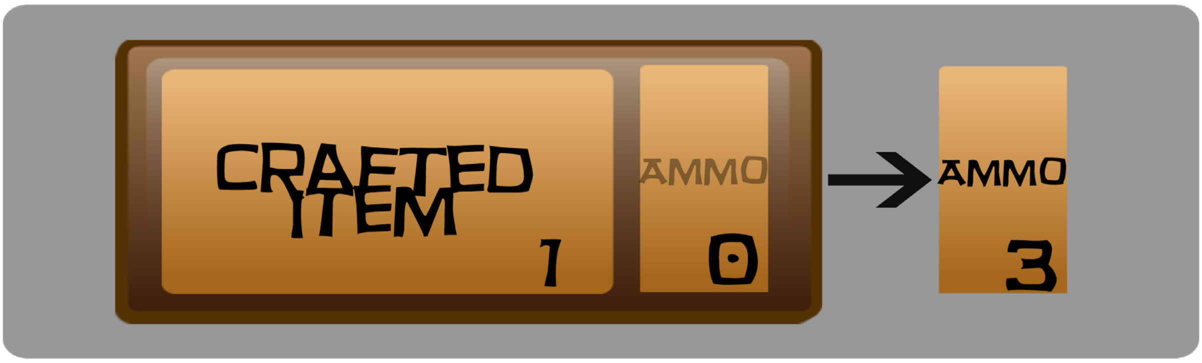
Health Bars



Inventory Base



Inventory Slots with clicked State



Teaser Trailer
Gameplay Trailer
Main Menu Background Video

6 Individual Project Research Paper – SS2018

08th of July, 2018
Jule Randzio
753172
Project 99+

How does the player emphasize and identify with the character he/she is playing during the Gameplay and why is this important?

Content:

1 Introduction

I am the programmer in the team so for me it is important to intensify the players experience during the game with well executed game mechanics. Since the age of the expected target audience differs from the age of our explicitly old characters, they still need to get captivated and connect with the character that is so different from them. Apart from Character Design and such things the feeling comes through the mechanics. So the objective is now to find out how the player emphasizes and identifies himself with the character to improve our gameplay and our mechanics.

2 Methods

Personally i feel like relating to a character always starts from a psychological standpoint. People experience empathy all the time and can relate to all sorts of things. So where does that come from exactly and what makes a character relatable? I would start with some sources on 'empathy', 'relating to ...' and 'identifying' to grasp the concept of the psychology behind it. Then I would head over to 'Making characters relatable in storytelling' as I´d assume that the conclusions one could draw from that could also be applied on games. For a good ending, to look up what happens when a player is not relating to a character would be a good idea.

3 Results

Identifying with a character starts with feeling empathy towards the character. Every human is born with the capability to feel empathy however women and men detect another persons emotions differently. To act accordingly, women focus on movements or facial expressions as to where men react best to certain behaviours. (6, 7) Feelings need to be well communicated and tolerable to be accurately identified.(4, 5) The reaction that is triggered then is to understand or feel what another person is experiencing (Wikipedia) and lessen the distinction between oneself and another subject. (1) That is exactly what we´re trying to achieve. To go even further than just relating is to wholly or partially transform into the character. We identify ourselfes as others all our lives, starting with our parents at a young age. This primary identification comes from emotional attachment whereas the partial (secondary) identification is more on an inspirational level. People look up to leaders. They seek for personas that represent something they stand for. It makes them feel like they have something in common.

Still, making a character relatable is not always about making them have something in common with the target group. Target groups can differ widely in various aspects. That would make the character very stereotypical. Character can have traits which are questionable. In that case however we need to build the character so that people want to like him. (9) For example equipping him with very cool abilities that support his actions. The player enjoys these abilities and wants to like the character even though he might be bad. Moreover certain abilities can definitely support his story line. It entices the player to like the character as the coolness of the actions overweighs the bad traits. Such mechanics can place the player in a similiar worldview as the character.

You don´t always have to show all the characters traits. Sometimes you can place them in mechanics and make them feel intuitive. Mechanics can imply traits, e.g. you can only interact with certain sorts of people, means you dislike the other type of people.

Giving a character an objective is always a good start. This can work differently though. Making a character a guardian of someone else invokes protective feelings in the player. Making him smart makes the player feel like he/she has the upper hand. Making him stand out from the rest of people and environment always attracts the players curiosity. This character becomes a self important mystery. (10)

4.1 Conclusion

In conclusion I would say that since we don´t get to put in that much facial animations we should focus on making movements and actions really explcicit so they show emotion. The characters can show amusement or anger through body movement. Even though old people don´t really represent something the younger audience can relate to they will still emphasize with the mischief that they can cause and also the multiplayer component of the game adds up to that. Playing against somebody else and destroying their reputation gives the player great satisfaction and puts the player in the characters worldview as they bond over something. The objects that will be put in the environment will be very divers and creative so technically using those objects as they wish and make them really versatile will act as the abilities of the character. The objective here is very simple: Ruin the opponents reputation. However we leave a lot of rooms for decisions to the player on how they want to go further.

4.2 Conclusion after the game

After the game is now finished I am very much delighted with how our artist showed emotion through animation. The over exxagerated reactions really worked out well. Also the walking animations fitted perfectly with the slow pace I implemented and the limited sprinting time since they are old people and can´t run forever without having to take a break. People were amused by our characters and really got that competetive feeling that we wanted to achieve despite their differences to the characters. The actions the characters could perform really helped the players to relate, it was interesting for them to find out how they could lower the others reputation wihtout harming them.

To sum it up I´d say that thinking about what to take into consideration before I started implementing the features really helped me during the process of the game. The artist and the game designer really have to work very close together and communicate a lot, to achieve the best result in those features.

Jule Randzio
Immatrikulation-Nr.: 753172
14.05.2018

5 Sources

- Identification (psychology), Wikipedia, (01/05/2018), ttps://en.wikipedia.org/wiki/Identification_(psychology) , 12/05/18
- Empathy, Wikipedia, (02/05/2018), https://en.wikipedia.org/wiki/Empathy, 12/05/18
- https://www.welt.de/gesundheit/psychologie/article5393400/Warum-der-Mensch-zur-Schadenfreude-neigt.html

(1) Hodges, S.D., & Klein, K.J. (2001). Regulating the costs of empathy: the price of being human. Journal of Socio-Economics., 12/05/18

(2) Dietrich, Cindy. "Decision Making: Factors that Influence Decision Making, Heuristics Used, and Decision Outcomes". Inquiries Journal. Inquiries Journal/Student Pulse LLC., 12/05/18

(3) Roth-Hanania, Ronit; Davidov, Maayan; Zahn-Waxler, Carolyn (2011-06-01). "Empathy development from 8 to 16 months: Early signs of concern for others". Infant Behavior and Development. 34 (3): 447–458. doi:10.1016/j.infbeh.2011.04.007. , 12/05/18

(4) Roth-Hanania, Ronit; Davidov, Maayan; Zahn-Waxler, Carolyn (2011-06-01). "Empathy development from 8 to 16 months: Early signs of concern for others". Infant Behavior and Development. 34 (3): 447–458. doi:10.1016/j.infbeh.2011.04.007. , 12/05/18

(5) Roth-Hanania, Ronit; Davidov, Maayan; Zahn-Waxler, Carolyn (2011-06-01). "Empathy development from 8 to 16 months: Early signs of concern for others". Infant Behavior and Development. 34 (3): 447–458. doi:10.1016/j.infbeh.2011.04.007. , 12/05/18

(6) Roth-Hanania, Ronit; Davidov, Maayan; Zahn-Waxler, Carolyn (2011-06-01). "Empathy development from 8 to 16 months: Early signs of concern for others". Infant Behavior and Development. 34 (3): 447–458. doi:10.1016/j.infbeh.2011.04.007. , 12/05/18

(7) Singer T, Seymour B, O'Doherty JP, Stephan KE, Dolan RJ, Frith CD; Seymour; O'Doherty; Stephan; Dolan; Frith (Jan 2006). "Empathic neural responses are modulated by the perceived fairness of others". Nature. 439 (7075): 466–9. Bibcode:2006Natur.439..466S. doi:10.1038/nature04271. PMC 2636868. PMID 16421576. , 12/05/18

(8) Rogers K, Dziobek I, Hassenstab J, Wolf OT, Convit A (Apr 2007). "Who cares? Revisiting empathy in Asperger syndrome" (PDF). J Autism Dev Disord. 37 (4): 709–15. doi:10.1007/s10803-006-0197-8. PMID 16906462. , 12/05/18

(9) What makes a video game main character relatable?, Mike Prinke, (19/01/2017), https://www.quora.com/What-makes-a-video-game-main-character-relatable , 12/05/18

(10) What makes a video game main character relatable?, Mike Prinke, (19/01/2017), https://www.quora.com/What-makes-a-video-game-main-character-relatable , 12/05/18

(11) Why is Multiplayer Shooting Games so much fun, TechyBlog (19/12/2016), http://techyblog.org/why-is-multiplayer-shooting-games-so-much-fun/ , 12/05/18