ComPuzzle

**Abigail Dehart**

**Brianna Dulik**

**Brandon Mastin**

**Cameron McGill**

**Other ideas to add/ idk where else to put them**

**Make some levels have a sequence already made but is faulty that they fix**

**Make outdoorsy levels, grass car tree and more indoors, door, fountain, road, floor**

**Need people sprites other than mr. robot**

**Instructor Comments and Evaluation**

**Table of Contents**

**Project overview and application**

**ComPuzzle is a game that teaches the user some basic programming concepts while also keeping it fun and engaging.**

**Programming can be overwhelming to beginners so we made this game to make it less intimidating.**

**We use a visual representation of the early programming concepts, such as a loop and statement that can be easily converted on to the screen to see the results of the mini program.**

**we made it a maze to still invoke some challenge into the project, but in a way that keeps you hooked and entertained rather than a tedious project.**

**We made the UI simple to reduce confusion and to make things look more simplified**

**We wanted to give some sort of narrative to the game, so we made the levels based around a day at CalU, like a parking lot, classroom, crowded halls, and other similar stuff**

**The player will interact by using code blocks that get added to a list, the blocks have simple commands like, ‘take step’, ‘turn left’ etc. that can be executed by hitting the ‘run’ button**

**The player should learn what a loop is and how it functions**

**The player should learn how to use a loop and learn nesting loops**

**The player should learn how to ‘debug’ as in follow their code as it is executed and find the problems that arise if they did not follow the logic enough.**

**Target audience, beginners / younglings**

**Levels will get more challenging as you progress, to help reinforce the programming ideas**

**Game will allow the player to replay levels, and they will unlock after beating them**

**When completing a level, the user will get updated with the total blocks used, and total instructions executed (like in loop)**

**There is no direct feedback unless there is a logic breaking code structure, as in loops without end loops or trying to loop negative times**

**There is a restart button so you do not need to delete all and rebuild**

**All commands can be moved up/down the list and deleted at any time as long as it is not being ran**

**Motivation**

**We were motivated by trying to describe basic programming concepts to people who are unfamiliar with programming**

**We wanted an easy way to teach people programming ideas without it getting to intimidating too fast**

**We wanted to**

**Comparison to existing similar products**

**Community or social implications of the project**

**System block diagram**

**Project implementation details**

**Difference from design document**

**Way more functions than inticipated**

**Adding newer features than expected**

**Calu theming**

**Buttons**

**No drag/drop**

**Reset is available at all times**

**Parent block is now just a png with a value that the robot reads**

**Same as goal**

**Challenges during implementation**

**Loops not working right, all loops exit at first end while**

**Indexing for nodes in the vbox**

**Creating instances of scenes**

**Creating a collision system**

**// Making the robot adapt to differing map sizes**

**Getting the robot to read the map layers**

**Getting the robot to read the correct local position**

**Weird collisions, the robot being ever so slightly close to a wall so it thinks that its in it and will not move**

**Map switching**

**Blocks reading properly**

**Map setup, new robot and new map then deleting old ones**

**LOOOPS – crashing no ends / going into nest**

**Exiting the excutions when it is running, nextmap and reset**

**Map coords not lining up correctly**

**Accidental crashing when it tries to read a deleted node**

**Being able to delete / move nodes while excecuting**

**Use of software engineering principles**

**User’s manual (usage, etc)**

**Mouse required**

**Keyboard not needed, but has uses(for loop and few shortcuts // need to do those)**

**Monitor required**

**Headphones not required**

**Mic not required or utilized**

**Menu // don’t got one**

**The main menu is where you have the most options**

**Play will bring you to the play menu**

**Settings will bring you to the settings menu**

**Quit exits the game**

**Play**

**New game brings you to level 1**

**Continue brings you to the last level played**

**Map select will allow you to choose what level you can play (if its unlocked)**

**Back will bring you back to the main menu**

**Settings**

**Setting stuff goes here**

**Idk any // todo**

**game**

**Take step makes the character move forward a single tile**

**Turn left makes the character turn to its left, ie counter clockwise**

**Turn right makes the character turn to its right, clockwise**

**Loop will repeat any blocks between the start of its commands to its associated end loop**

**Each loop will have one, and only one, end loop**

**Loops can be put inside other loops (nesting)**

**End loops will be associated to the most recent loop block to be triggered**

**An end loop will only work with its associated loop block**

**The screen has 3 main sections, map, buttons, command list**

**The command list is on the right side, it stores the commands to be executed**

**The commands will be run in top down order**

**You can reorganize nodes by using the up or down button that is attached to each command block in the command list**

**The loop command has its own counter that ranges from 0 – 99**

**If there is a value that is not a number there will be an error message**

**The buttons section has the reset button, quit button, run button and the spawn buttons**

**The reset button will put the robot back to the spawn point // also clear the list**

**The quit button will quit the game // prob gonna be reworked eventually**

**The run button will lock all spawn buttons and command list from being edited and also disable the run button until the commands are finished running or the reset button is hit**

**The maps are what the player sees and allows them to plan out their commands**

**There are many obstacles to avoid and some of them may have a special function // need todo**

**To beat a level, the play must have the robot on the goal block at the end of the command list**

**If the player steps on the goal, but continues to walk past, it does not count as a victory**

**You must end on goal**

**Reaching the goal will unlock the next map**

**If the player ends on goal, they will see a victory screen with several stats**

* **Number of commands executed**
* **Number of code blocks used // need to do all 3**
* **Number of resets**

**After the victory screen you will get several options, replay, continue, main menu, quit**

**Replay will set all stats back to 0**

**Continue will move you to the next map**

**Main menu brings you back to the main menu**

**Quit exits the game**

**If all maps are unlocked, then you beat the game**

**References**

**Table of Index / Glossary**

**Appendix: Team Details and Individual Contributions**

**Appendix: Writing Center report**

**Appendix: Code listing (Not required to be in the hard copy)**

**Appendix: Workflow Authentication (individual signatures**