



PACMAN

VÍCTOR ORTEGA
HECTOR RODRIGUEZ

GROUP 97

FINAL PROJECT

PROGRAMMING





TABLE OF CONTENTS

INTRODUCTION

CLASSES, PARAMETERS* AND METHODS

MAIN ALGORITHMS

THE PROJECT

EXTRAS

CONCLUSIONS

*ATTRIBUTES WILL APPEAR IN THE DETAILED EXPLANATION
OF THE ENTIRE PROJECT, AS THERE ARE LOTS OF THEM



INTRODUCTION

THIS DOCUMENT IS A REPORT ABOUT OUR FINAL PROJECT,
WHICH CONSISTS ON THE DEVELOPMENT OF THE RETRO GAME
"PACMAN" THROUGH PYTHON OBJECT ORIENTED PROGRAMMING.

PAC-MAN IS A CLASSIC ARCADE VIDEO GAME RELEASED IN
1980. IT IS AN ACTION MAZE SURVIVAL GAME AND IT HAS
BECOME ONE OF THE MOST ICONIC AND INFLUENTIAL VIDEO
GAMES OF ALL TIME.

OUR PROJECT CONSISTS ON PROPERLY RECREATING THIS
CLASSIC GAME THROUGH PYTHON.

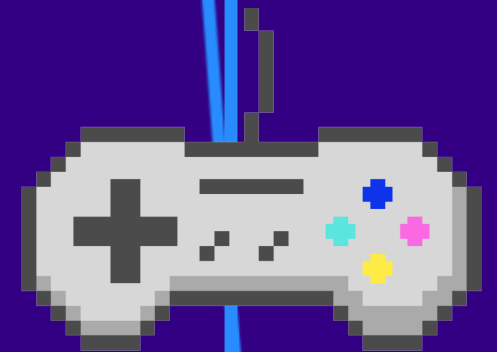
CLASSES



PACMAN

THE PACMAN CLASS IS THE FOUNDATION OF THE PROJECT, THE BASIS. IT CONTAINS THE INFORMATION ABOUT THE PRINCIPAL CHARACTER (PACMAN), ITS MOVEMENT THROUGH THE MAP AND THE COLLISIONS WITH IT.

THE CLASS CONTAINS TWO PARAMETERS, 'X' AND 'Y', WHICH DETERMINE THE POSITION OF THE MAIN CHARACTER IN THE BOARD. IT ALSO CONTAINS ITS CORRESPONDING PROPERTIES AND SETTERS, AS WELL AS TWO METHODS. THE FIRST ONE CONTAINS THE PACMAN MOVEMENT AND THE SECOND ONE CONTAINS THE PACMAN MOVEMENT ANIMATION.



BOARD

THE BOARD CLASS IS WITH NO DOUBT THE MOST IMPORTANT ONE IN THE ENTIRE PROJECT. IT PUTS TOGETHER EVERY OTHER CLASS, INTERPRETES THE INPUT OF THE KEYBOARD AND CONTAINS THE PYXEL INITIALIZATION IN ORDER TO EXECUTE THE GAME.

THE CLASS CONTAINS TWO PARAMETERS, 'MAZE' AND 'MUSIC', WHICH RECEIVE INPUTS ABOUT THE MAP THE PLAYER WANTS TO PLAY ON AND IF HE/SHE WANTS TO ENABLE THE MUSIC. IT ALSO CONTAINS PROPERTIES, SETTERS AND THREE METHODS, RELATED TO THE MOVEMENT OF THE CHARACTERS (PACMAN AND GHOSTS), THEIR STARTING POSITIONS AND THE VISUALIZATION OF THE BOARD, THE CHARACTERS AND THE EDIBLES.

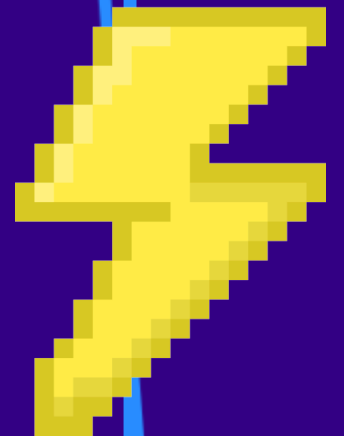
CLASSES



GHOSTS

THE GHOSTS CLASS CONTAINS THE PRINCIPAL INFORMATION OF EVERY GHOST, WHICH IS THEN USED THROUGH INHERITANCE TO CREATE THE CLASS OF EVERY DIFFERENT TYPE OF GHOST: BLINKY, PINKY, INKY AND CLYDE.

THE CLASS CONTAINS THREE PARAMETERS, 'PACMAN', 'COLOR' AND 'SUPERPELLET', WHICH DETERMINE THE GHOST COLOR AND ALLOW THE GHOSTS TO INTERACT WITH THE ENVIRONMENT (PACMAN, WALLS, SUPER-PELLETS, ETC.). IT ALSO CONTAINS ITS CORRESPONDING PROPERTIES AND SETTERS, AS WELL AS SIX METHODS, WHICH WILL BE DETAILED LATER DURING THE GENERAL EXPLANATION OF THE PROJECT.



GHOSTS

- REDGHOST (BLINKY)
- BLUEGHOST (INKY)
- PINKGHOST (PINKY)
- ORANGEGHOST (CLYDE)

THIS CLASSES ALL HAVE TWO PARAMETERS TO INTERACT WITH PACMAN AND BECOME VULNERABLE: 'PACMAN' AND 'SUPERPELLET'. THEY HAVE TWO METHODS IN ORDER TO GET OUTSIDE THE BOX AND MOVE AROUND: 'ACTIVATE CAGE' AND 'REVIVE AND EXIT JAIL'.



CLASSES



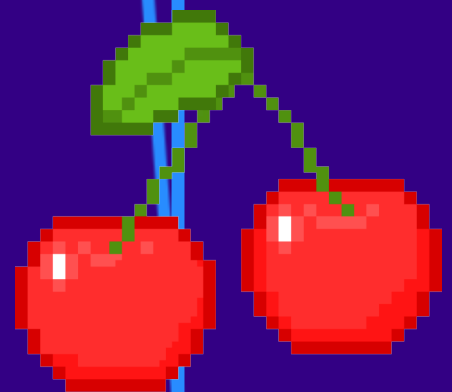
EDIBLES

THE EDIBLES CLASS CONTAINS THE GENERAL INFORMATION ABOUT THE PELLETS, THE SUPER-PELLETS AND THE FRUITS.

EACH OF THIS ELEMENTS WILL HAVE A CLASS WHICH USES EDIBLES THROUGH INHERITANCE.

THE EDIBLES CLASS JUST CONTAINS ONE PARAMETER ('SCORE_VALUE') AND ITS CORRESPONDING PROPERTY AND SETTER.

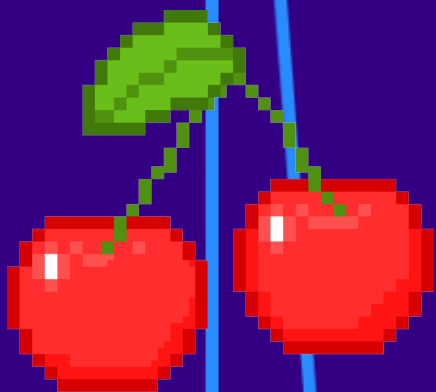
- PELLETS: THIS CLASS HAS MANY ATTRIBUTES, TO STORE THE EATEN AND NOT EATEN PELLETS, THE POINTS AND TO LOCATE AND PLACE THEM ON THE BOARD. IT HAS TWO METHODS, ONE TO KNOW IF THE PELLETT HAS BEEN EATEN OR NOT AND THE OTHER ONE TO PRINT THE PELLETT ON THE BOARD.



EDIBLES

- SUPERPELLETS: THIS CLASS IS EXACTLY THE SAME AS THE PELLETT ONE, SAME ATTRIBUTES AND SAME METHODS.
- FRUITS: THIS CLASS IS PART OF THE EXTRA ELEMENTS. IT HAS PARAMETERS 'MATRIX' AND 'PACMAN' AND MANY ATTRIBUTES. ALL OF THEM LOCATE THE FRUIT IN THE BOARD AND MAKE THE FRUIT ABLE TO INTERACT WITH THE PACMAN.

THE EDIBLES ARE PRINTED ON THE BOARD IN TERMS OF THE MATRIX, IF THE MATRIX HAS A '1', A PELLETT IS PLACED, IF THE MATRIX HAS A '2', A SUPER-PELLETT IS PLACED, AND IF THE MATRIX HAS A '3', A FRUIT IS PLACED



ALGORITHMS

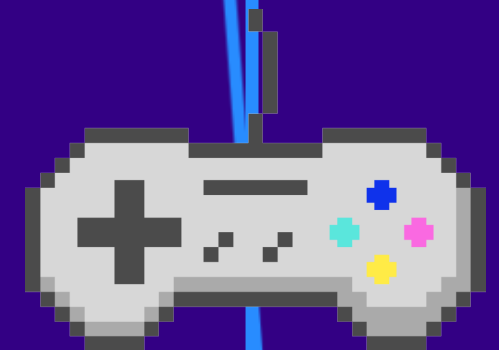


MOVEMENT

THE MOVEMENT WAS DEFINITELY ONE OF THE HARDEST PARTS OF THE PROGRAM. WE HAD TO CREATE THE 'MOVE' METHOD WITH 5 PARAMETERS, IN ORDER TO MAKE THE PACMAN INTERACT WITH THE BOARD.

ALSO, WE HAD TO CREATE THE 'LOOKING' ATTRIBUTE, IN ORDER TO MAKE THE TURNS 'AUTOMATIC' (FOR EXAMPLE, IF THE PACMAN MOVES RIGHT AND I PRESS 'UP', THE PACMAN WILL MOVE RIGHT UNTIL HE CAN GO UP). THAT HAPPENS THANKS TO THE 'LOOKING' PARAMETER.

 FURTHERMORE, WE CREATED ROADS BY UPDATING THE MATRIX POSITION ONLY WHEN PACMAN IS IN THE EXACT 16X16 SQUARE. THEREFORE, HE WILL ONLY BE ABLE TO TURN WHEN HE IS EXACTLY IN THE 16X16 EMPTY SQUARE, AND NOT IN THE HALF, BECAUSE IT WOULD WALK THROUGH WALLS.



BOARD

ANOTHER HARD PART WAS REALISING HOW TO CREATE THE BOARD.

FIRSTLY, WE HAD TO LEARN TO CORRECTLY USE PYXEL TILEMAPS IN ORDER TO DESIGN THE MAP. HOWEVER, THIS WAS PRETTY INTUITIVE. THE HARD PART WAS CREATING THE MATRIX FOR THE MOVEMENT AND COLLISIONS, AS WELL AS ASSIGNING AN ELEMENT IN TERMS OF THE PACMAN POSITION IN GAME.

 ALSO, THE BOARD IS THE FINAL CLASS, THE ONE THAT MIXES EVERYTHING, SO IT WAS PRETTY HARD TO CORRECTLY ASSEMBLING EVERYTHING. WE HAD TO CREATE LOTS OF ATTRIBUTES FOR EACH CLASS, AS WELL AS THE HEIGHT AND WIDTH OF THE BOARD. IT ALSO CONTAINS THE PYXEL INITIALIZATION AND PRINTS EVERYTHING.

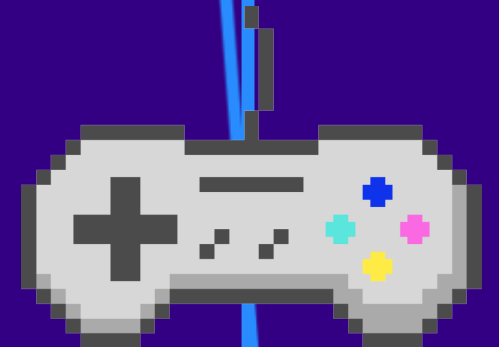
ALGORITHMS



GHOSTS

THE GHOSTS CLASS IS ONE OF THE MOST COMPLEX ONES.

THERE IS AN INFINITELY LONG LIST OF ATTRIBUTES, BUT THE IMPORTANT PART IS ACTUALLY THE SIX METHODS. THE FIRST THREE METHODS ARE IN CHARGE OF THE GHOST MOVEMENT AND INTERACTION WITH THE DISTANCE UNTIL PACMAN (WE USED THE EUCLIDEAN DISTANCE, WHICH WE HAVE LEARNT IN OTHER SUBJECT). THE NEXT ONE IS FOR THE VULNERABLE STAGE, WHERE EACH GHOST USES ITS OWN MOVEMENT PATTERN DEFINED IN ITS OWN CLASS IN ORDER TO GET AWAY FROM THE PACMAN. THE LAST TWO ARE ACTIONS: 'KILL' AND 'DEAD' (GET KILLED BY PACMAN).



SCORE, LEVELS AND LIVES

THE SCORE, LEVELS AND LIVES IMPLEMENTATION WAS AN EASY STEP, BUT A REALLY IMPORTANT ONE, AS THE SCORE, THE LEVELS AND THE LIVES ARE TWO FUNDAMENTAL ASPECTS OF THE GAME.

WE DECIDED TO IMPLEMENT INFINITE LEVELS AND THREE LIVES, AS IN THE ORIGINAL GAME OF CLASSIC PACMAN. WE DECIDED TO PRINT EVERYTHING BELOW THE MAP, SO THAT IT IS EASY TO VISUALIZE AND EVERYTHING IS IN THE SAME PLACE.



THE PROJECT

- **PLANNING**

BEFORE STARTING WITH THE PROJECT, WE ORGANISED THE WORK. WE THOUGHT ABOUT THE DIFFERENT ASPECTS THAT HAD TO BE IMPLEMENTED AND DIVIDED THE TASKS. ONE OF US WAS IN CHARGE OF CORRECTLY PROGRAMMING THE PACMAN, ITS MOVEMENT AND ITS COLLISIONS WITH THE MAP, AND CREATING THE DIFFERENT MAPS (PYXEL DESIGN). THE OTHER ONE HAD TO PROGRAM THE PACMAN INTERACTION WITH EDIBLES AND GHOSTS, AS WELL AS THE GHOSTS MOVEMENT PATTERNS AND THE SCORES AND LIVES VALUES.

- **TIMELINE**

 WE COPIED THE GUIDE IN AULA GLOBAL AND BEGAN TO WORK. FIRSTLY, WE FOCUSED ON GETTING THE PACMAN TO MOVE WITH THE KEYBOARD INPUTS, USING THE METHODS 'MOVE' AND 'UPDATE_IMAGE'. THEN, WE USED PYXEL TO CREATE A FIRST MAP (TILEMAPS) AND CREATED A MATRIX ATTRIBUTE IN THE BOARD CLASS TO ADD COLLISIONS. WE CREATED SOME "ROADS" FOR THE PACMAN TO MOVE, SO THAT HE COULD NOT WALK THROUGH WALLS.



THE PROJECT

AFTER WE HAD FINISHED WITH THE MOVEMENT (THE PACMAN MOVED WITH FREEDOM ALONG THE ENTIRE MAP WITHOUT WALKING THROUGH WALLS), WE BEGAN TO PROGRAM THE EDIBLES. WE USED THE MATRIX TO FILL THE MAP WITH PELLETS AND TO PLACE SOME SUPER-PELLETS IN SPECIFIC PLACES. THEN, WE PROGRAMMED THE FRUITS TO APPEAR AND DISAPPEAR AFTER A PERIOD OF TIME.

AFTER WE HAD FINISHED WITH THE EDIBLES, WE PROGRAMMED THE GHOSTS. WE CREATED A GENERAL CLASS WITH DIFFERENT METHODS FOR GHOST STAGES AND ACTIONS (VULNERABLE STAGE, KILL PACMAN, GET KILLED BY PACMAN, ETC.). THEN, THROUGH INHERITANCE, WE CREATED EACH GHOST CLASS WITH A UNIQUE MOVEMENT PATTERN AND BEHAVIOUR.

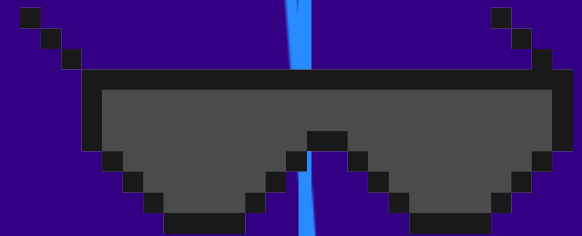
FINALLY, WE IMPLEMENTED THE SCORE, THE LIVES AND THE LEVELS, FINISHING THE PROJECT AND THE ENTIRE GAME.

EXTRAS



MAZES AND MUSIC

- MAZES: WE HAVE CREATED 3 DIFFERENT MAZES. THE PLAYER CAN CHOOSE THE MAZE HE WANTS TO PLAY ON THROUGH THE PARAMETER MAZE IN THE BOARD CLASS. THE FIRST AND SECOND MAZES ARE PRETTY SIMPLE, THE THIRD ONE IS A BIT MORE COMPLEX, SIMULATING AN ACTUAL MAZE, A LABERYNTH.
- MUSIC: WE HAVE CREATED A MUSIC COMPOSITION TO MAKE THE GAME MORE ENJOYABLE AND IMMERSIVE. ONE OF US IS A MUSICIAN, SO WE HAVE BEEN ABLE TO CREATE AN AMAZING SOUNDTRACK INSPIRED ON THE GAME GEOMETRY DASH.



GHOSTS AND FRUITS

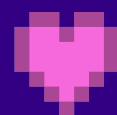
- GHOSTS: WE HAVE CREATED DIFFERENT TYPES OF GHOSTS, EACH ONE WITH A DIFFERENT MOVEMENT PATTERN, EXACTLY AS IN THE PACMAN ORIGINAL GAME. BLINKY CHASES DIRECTLY PACMAN, PINKY TRIES TO HIT PACMAN FROM THE FRONT, CLYDE IS COMPLETELY RANDOM AND INKY SOMETIMES BEHAVES AS CLYDE AND OTHERS AS PINKY.
- FRUITS: WE HAVE CREATED DIFFERENT FRUITS (CHERRY, ORANGE, BANANA...) WHICH GIVE EXTRA POINTS TO THE PLAYERS. THEY APPEAR AND DISAPPEAR AFTER A CERTAIN PERIOD OF TIME.





CONCLUSIONS

- SUMMARY: WE HAVE RECREATED WITH ALMOST EXACT PRECISION THE CLASSIC PACMAN GAME. FURTHERMORE, WE HAVE NOT ONLY STUCK TO THE OBLIGATORY SPRINTS, BUT ALSO WE HAVE ADDED MANY EXTRA ELEMENTS, GIVING THE GAME A MUCH MORE PROFESSIONAL STYLE. WE HAVE USED LOTS OF KNOWLEDGE WE HAVE LEARNT IN THE PROGRAMMING SUBJECT, SPECIALLY RELATED TO OBJECT ORIENTED PROGRAMMING.
- MAIN PROBLEMS: OUR MAIN PROBLEMS HAVE BEEN TIME MANAGEMENT AND LACK OF KNOWLEDGE AND EXPERIENCE RELATED TO PYXEL AND GAME DEVELOPMENT. IT WAS A COMPLETELY NEW 'WORLD' FOR US, WHAT HAS MADE THE PROCESS MUCH MORE SLOW AND DIFFICULT.
- COMMENTS, EVALUATION AND ASPECTS TO IMPROVE: WE CANNOT BELIEVE THAT WE HAVE BEEN ABLE TO DEVELOP AN ENTIRE GAME BY OURSELVES. IT HAS BEEN A HARD JOURNEY BUT IT HAS DEFINITELY BEEN WORTH IT. WE ARE REALLY HAPPY WITH THE RESULT AND WITH THE WAY WE HAVE PUSHED OURSELVES TO ACHIEVE IT. THE ASPECT WE HAVE TO WORK ON THE MOST IS THE TIME MANAGEMENT, AS WE SHOULD HAVE STARTED BEFORE.



GAME OVER