



University Of Dhaka

Department Of Computer Science & Engineering

Project Report:

Fundamentals Of Programming Lab (CSE-1211)

Project Name:

The Untamed

Team Members:

1. Asfin Jannat Shamsi (Roll - 23)
2. Tanzeem Malia (Roll - 51)

Introduction

The Untamed is a 2D platform running based game where the player has to pass certain distances in each level without getting hit by any of the four obstacles introduced in the game & finishing the game. The UI/UX of the project will help the end users to navigate through the project: they will be able to start a new game from **NEW GAME**, take help/instructions from **SEEK HELP** & quit if they want to play the game later on by clicking on **QUIT GAME**. The credit is given to the members of this project group on **CREDIT**.

Objective

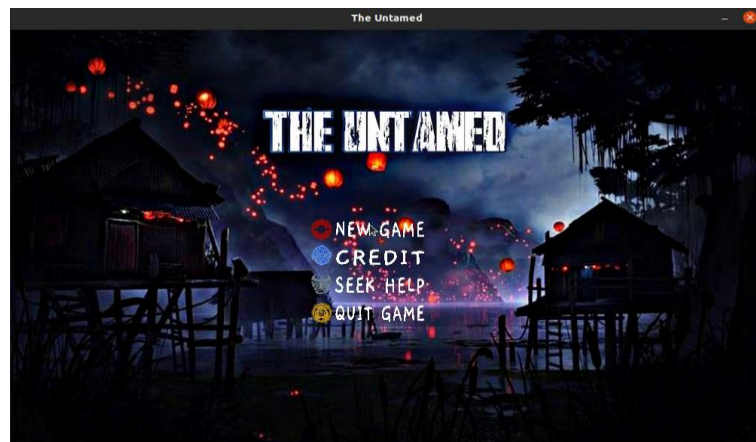
The Untamed is a 2D platform adventure game where the player has to travel a certain distance to pass each level. This game has 5 levels with different paced obstacles. Four types of obstacles are introduced in the game which are Static Block, Pillars, Falling Objects & Flame.

As the levels increase, the pace of these obstacles will also increase. So, the player will have to follow the **Seek Help** part from the menu to get instructions on how to pass all the levels without dying. There are also gems in each level so that the player can get extra points & have the High score.

In The Untamed, we aimed for different paced & textured obstacles in order to make the game more exciting & just like any other games, we also gave our avatar to be more flexible so that the player can pass each of these hurdles & go for a win. The levels are fast paced & the appearances of the obstacles are sudden : that way, the game is made more exciting.

Project Features

In the starting window, an end user will see **NEW GAME**, **SEEK HELP**, **CREDIT** & **QUIT GAME**.



Obstacles

The objective of this project is quite simple : pass each level without touching or making contact with any of the obstacles. There are 4 types of obstacles in The Untamed.

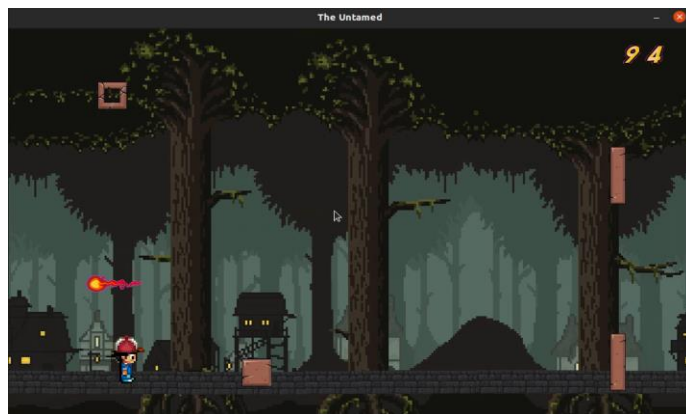
1 **A static object on the platform** - it creates hindrance for the player to move forward which will be different in appearance in every level.



2 **A gap made of two pillars** - A top & a lower pillar will create a gap between their placement in the game background. The player will face difficulty while passing the hurdles & missing the gap as an escape route.



3 **A falling broken object** - For each level, the falling object will differ in appearance & try to crush the player with its weight. If the player is not able to dodge them on time, the game will be over.

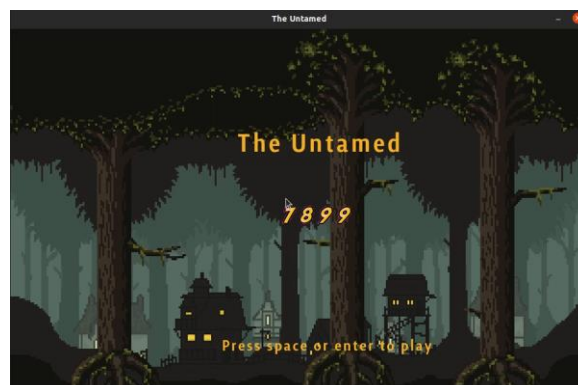


4 **A spark of flame** - This is the animated obstacle of our project which will try to burn the player & stop him from finishing the game. As it appears randomly in the game, the player has to be prompt enough to press the arrow keys to save himself from burning & eventually finishing the game.

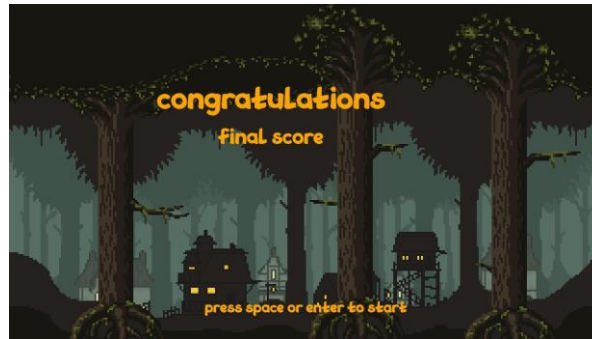


In order to enjoy this 2D game to its full potential, there are certain tricks.

1. The static object on the platform is visible throughout the game. So the player only has to control the keys to jump over those.
2. The gap made with the pillars may seem fast paced as the levels progress but the controlled use of key arrows will make the player pass each of these hurdles quite easily.
3. A red arrow will appear each time a falling object is supposed to fall. So, the player has to keep an eye on the screen to know the upcoming danger.
4. A flame is supposed to appear whenever a player is facing a smooth sail in The Untamed. So, they have to be on alert even when no visible indications of danger are there.

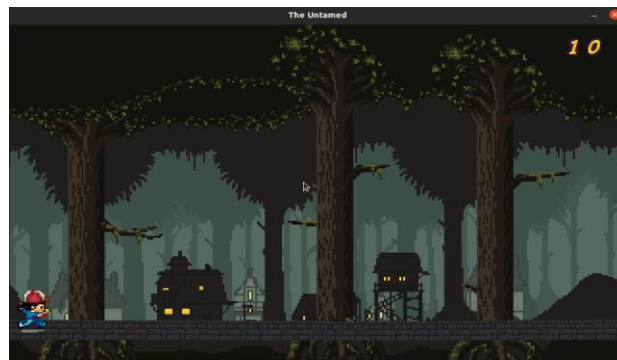


The following image will appear if a player is able to finish all of the five levels.



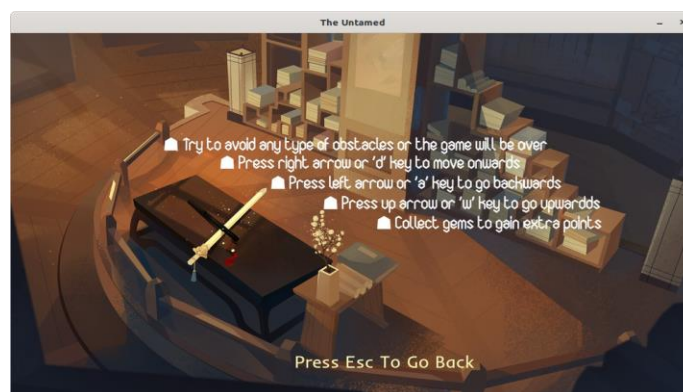
Character

The character of this project is inspired by the modern version of ancient Chinese heroes. The character is able to move forward, jump & move backward. Points are earned while the character is moving & this point is maintained sequential. Extra points are earned if the player is able to collect gems in each level.



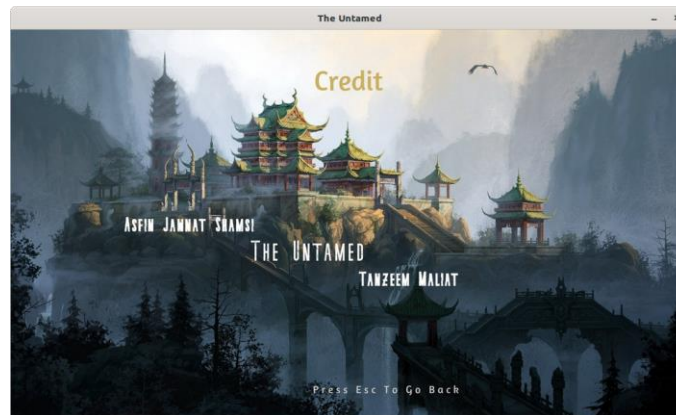
Seek Help

The instructions for this game are shown in this part. So, the player can read them or explore the game on their own.



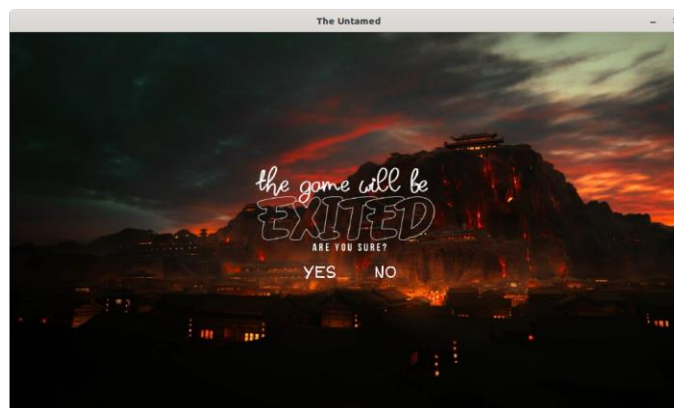
Credit

In this window, the credit has been given to the team members of group 4, Asfin Jannat Shamsi & Tanzeem Maliat.



Quit Game

The end user will be given the option to “Quit Game” in this window. If the player clicks on “NO” then it’ll be redirected to the main menu.



Project Modules

There are different types of modules to make sure that the future development of The Untamed can be continued without much problems. There are header library files for each element of the game. For instance:

The Arrow (Arrow.h)

This header file basically controls the size and position of the arrow texture in the project. The position of the arrow texture also incorporates with the character’s position in order to make the game more exciting. The arrow.cpp file is also used for the appearance of the falling objects in each level. We have used this

header file to control how frequent the arrow will appear depending on how frequent the falling objects will appear.

The Block (Block.h)

This header file controls the the static block shown in the game. The position of the block structure is controlled by the block.cpp file. The detection of collisions of the character with the blocks are also handled by this module. If the collision happens then the game will be over. We have used this header file to control for how long the blocks will appear and how frequent they will show themselves in a level.

The Falling Block (FallingBlock.h)

This is an attractive feature of our project. The falling block header file will control the pace, position & time of the random falling objects in the project. The gravity of this falling object is also fixed in this module. If these objects collide with the character then the game will be over. The .cpp file for this header file will be associated with the arrow.cpp file in order to give the player an exciting opportunity at winning.

The Top Pillar (TopPillar.h)

The top pillar header file is an important part for The Untamed as this part will help with another feature for the project which is the escape route between the two pillars. The header file has been used in the class structure just to help set the position of the top pillars throughout the game & this file also syncs up with the player's position in the game in order to make the game more enjoyable. The position and size of the top pillars are maintained by this module. If the player collides with these the game will be over. How frequent the top pillars will appear in each level is decided by this module.

The Bottom Pillar (BottomPillar.h)

The bottom pillar header file, just like the top pillar .cpp file, is an important part to enjoy The Untamed to its full potential. In this .cpp file, the texture, the size, the position of the bottom pillars are maintained. If the player collides with these the game will be over. How frequent the bottom pillars will appear in each level is decided by this module as well.

The Gem (Gem.h)

The gem helps the player to earn the extra points. The gem header file handles the texture, the position, the size, the speed & the equivalent marks for collecting each gem in different levels. If the player touches the gem texture then 1000 extra points are earned.

The Character (Character.h)

Just like any other games, the character sprite has to be managed quite precisely so that the end user feels charmed by it and doesn't lose the curiosity to try out the game and that has been our purpose while managing the character header file. In this header file, the texture, the velocity, the in-air velocity, the gravity, the position, the size and the maximum height one can get upto using the arrow key buttons are

maintained. The header file handles the relevant class so that upon the mentioned function getting called, the appropriate character is loaded and can be altered by the player's will.

The Flame (Flame.h)

The flame is an important obstacle for our project as this one appears suddenly. The player or the end user has to be vigilant enough so that they don't die just after the game starts. The flame header file manages the texture, the size, the position, the collision, the frame rate (or the animation) of the flame so that the game is more exciting. We introduced necessary variables in the header & .cpp files so that upon getting the function called, the necessary graphics are synced up to the main game loop.

Game Variables (GameVariables.h)

In this module, necessary variables of the game are declared to easily link up with other header files.

The Game (Game.h)

The game header file is the header file where we call each of the graphics and the obstacles and link them up with the game loop. Here, we deal with the velocity, the conditions for ending the game (or more likely the player failing to dodge the obstacles). Each level is thoroughly coded according to their respective features.

The menu part is controlled by the Game.cpp file. SDL_ttf header file has been used to show the options in the menu. Keyboard and mouse events are controlled in this module as well.

The Score Digit (ScoreDigit.h)

The score digit header file is the part we wrote for showing the highscore & the current player's score on the screen. The basic number theory has been used in order to make the score part work smoothly. The texture, the position, the frame rate & the reset version of the score while the game is crossed out are also done in the relevant header & .cpp file.

The Sprite (Sprite.h)

The sprite header file is the part which incorporates the character header file and makes the character more lively. We are handling the collision, the frame rate, the movement and the position of the sprites in this particular header file.

The Sprite Group (SpriteGroup.h)

The sprite group header file is basically incorporated with the character, the sprite & the game variables header file so that each part and feature of the game is properly working. The SpriteGroup.cpp file handles when to load the sprite groups, how many sprite groups should be loaded, when to delete the sprite groups, when to set new animation. And if the sprite groups collide with other obstacles the game will be over.

Team Member Responsibilities

Responsibilities have been a tough decision while working on this project. Still, we tried our level best to distribute the workload equally. The workloads are distributed as below:

1. Asfin Jannat Shamsi :

- Arrow.h
- Block.h
- FallingBlock.h
- TopPillar.h
- BottomPillar.h
- Flame.h

2. Tanzeem Maliaat :

- Character.h
- Game.h
- GameVariables.h
- Gem.h
- ScoreDigit.h
- Sprite.h
- SpriteGroup.h

The elements used in our project had been chosen and edited by both of the team members simultaneously. The .cpp files have been handled according to the work distribution by both of the team members.

Platform, Library & Tools

SDL2 is used as a platform in this project. C/C++ is simultaneously used as a coding language preference in our project. The basic libraries such as stdio.h , stdbool.h, SDL2/SDL.h, SDL2/SDL_image.h, SDL2/SDL_ttf.h, SDL2/SDL_mixer.h, SDL2/SDL_timer.h & so on are used.

Other than basic library functions, we used customized library functions for the development of our software.

For the graphics part, we used free resources from the internet for our game character sprites, obstacles & background. We credited the owners of the images which were not free.

Limitations

Even though we are quite happy with the way our software turned out to be, there are still rooms for improvement. The UI/UX could have been more smooth. Proper usage of files could have been done so that different players can come back later to enjoy the game, despite pausing mid game. The game is competitive as it is but it can be made even more challenging if a battle for the top player were to commence. In other words, a scope for leaderboard where the players will try to upgrade their position is a limitation we are facing with this version of The Untamed.

Conclusions

While brainstorming for this project, we learnt a lot of new things. We also learnt how to incorporate SDL & C/C++ in our project properly so that the software runs smoothly. Many changes have been done with the original plan for our project. But despite all that, we are satisfied with this version of our software.

Future Plans

During this project work, we learnt lots of new things. As a beginner, we did not have a total grasp over the platforms, libraries & tools. But we have hope that if we decide to further develop the versions of The Untamed, we would very much like to surpass the limitations mentioned in this report. If we are not able to develop it further, proper documentation has been done so that interested developers can improvise it even further.

As this is our first ever project, this software still has lots of scopes to be developed. Hence, we decided to make the source codes of it, open to anyone who wants to further develop this work.

Link

1. GitHub Repository: https://github.com/MTanzeem/TheUntamed_project
2. YouTube Sample: <https://www.youtube.com/watch?v=8ycR-6bugCE>

Reference

1. <https://lazyfoo.net/tutorials/SDL/>
2. <https://wiki.libsdl.org/>
3. <https://www.youtube.com/watch?v=yFLa3ln16w0&t=1501s>
4. <https://www.youtube.com/watch?v=KsG6dJILBDw&list=PL2RPjWnJduNmXHRYwdtublIPdlqocBoLS>
5. [Animations with Sprite Sheets in SDL2 - Gigi Labs](#)
6. [SDLgamer Menu - SDL Gamer](#)
7. <https://stackoverflow.com/>
8. <https://youtube.com/playlist?list=PL949B30C9A609DEE8>