

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ, ΤΜΗΜΑ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

ΑΝΑΠΤΥΞΗ ΕΞΥΠΝΩΝ ΔΙΕΠΑΦΩΝ ΚΑΙ ΠΑΙΧΝΙΔΙΩΝ

ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ - PROJECT

 **Zelda CSD:** *The Adventures of Link*



ΔΙΔΑΣΚΩΝ: ΑΝΤΩΝΙΟΣ ΣΑΒΒΙΔΗΣ
Χειμερινό Εξάμηνο 2022-2023

Περιεχόμενα

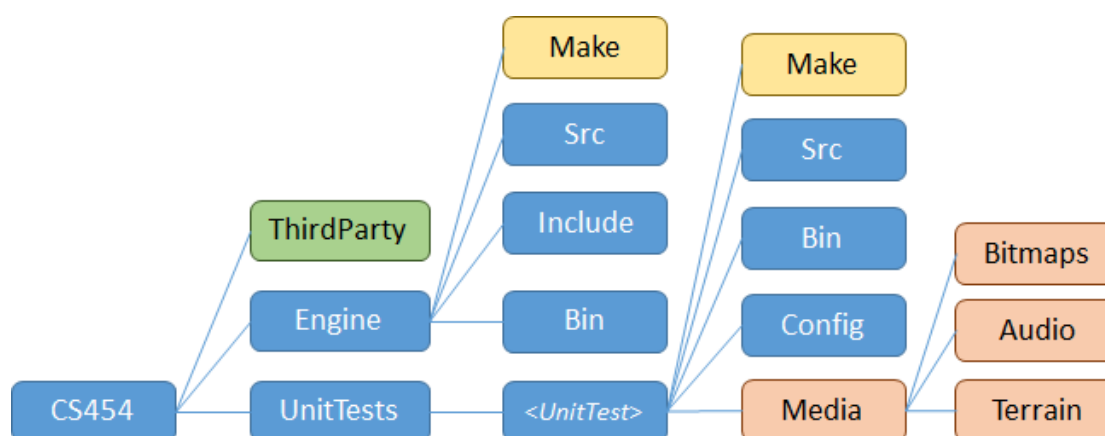
Περιεχόμενα	1
Εισαγωγή	2
Unit 1: Edit, Store, Load, Construct and Scroll a Tile Map	3
Unit 2: Edit, Store, Load, Construct and Use a Grid Map	4
Unit 3: Implement Layers, Animation and Sprites	5
Link character	7
Slash	7
Crouching Stance	7
Elevator and Scene Transition	8
The Full Project	8
Placing Powerups, Items and Bonuses	9
Placing NPCs	9
Supporting Pause / Resume	10
Characters	11
Link	11
Enemies	12
Power-ups	13
Spells	14
Optional Configuration Parameters	14
Optional Extras	15
Ομαδική εργασία -παράδοση	15

Εισαγωγή

Η εργασία υλοποιείται σε φάσεις, οι πρώτες από τις οποίες είναι προπαρασκευαστικές και σκοπό έχουν να εξασφαλίσουν την έγκαιρη και σωστή υλοποίηση της βασικής μηχανής (game engine).

Κυρίαρχο χαρακτηριστικό της υλοποίησης είναι αφενός η διαφοροποίηση της μηχανής ως ξεχωριστό library, αφετέρου η υλοποίηση των πρώτων φάσεων ως ανεξάρτητων unit tests (projects with executable) τα οποία τεστάρουν βασικά χαρακτηριστικά της μηχανής, και επίσης μπορεί να υλοποιούν μερικές λειτουργίες που θα τις χρειαστείτε στην κατασκευή του κυρίως παιχνιδιού.

Καθώς οι προπαρασκευαστικές φάσεις είναι πολύ μικρές, θα ελέγχονται κατά τη διάρκεια του μαθήματος, με 1 λεπτό για κάθε ομάδα. Ο κώδικας θα παραδίδεται επίσης. Η δομή του κώδικά σας (folder and project structure), πρέπει να είναι ως εξής.



Unit 1: Edit, Store, Load, Construct and Scroll a Tile Map

Τα βήματα που πρέπει να κάνετε είναι τα εξής:

1. Να βρείτε και να **εγκαταστήσετε κάποιον free tile map editor** ο οποίος μπορεί και αποθηκεύει τα maps σε κάποιο απλό text format. Ένας προτεινόμενος editor είναι ο Tiled, Version 1.9.2 (Stable)
 - <https://www.mapeditor.org/>
2. Να **βρείτε κάποια έτοιμα maps και tile sets** για αυτόν τον editor. Στην επιλογή σας σταθμίστε και τη διαθεσιμότητα τέτοιων maps. Ειδικά για την περίπτωση του Tiled, στο folder `UnitTest1/media` Project θα βρείτε ένα έτοιμο map καθώς και το αντίστοιχο tile set (πηγή <https://beast-pixels.itch.io/overworld-tilesheet-grass-biome>). Προσοχή, αυτό έχει δύο maps (layers). Μόνο το βασικό θα φορτώσετε (θα καταλάβετε ποιο είναι).
 - **Ωστόσο, είναι υποχρεωτικό να βρείτε και ένα ακόμη map και tile set.**
 - *Το μέγεθος του map τουλάχιστον σε μία από τις δύο διαστάσεις θα πρέπει να είναι 3-4 φορές το view window για να έχουμε επαρκές περιθώριο για το scrolling.*

3. **Να κάνετε export όλο το map σε text.** Ειδικά για το Tiled, κάνετε export σε csv (Comma Separated Values), που επιτρέπει very fast and easy parsing. Μέσα στο csv format του Tiled εμφανίζονται indices από το tile set. Το Tiled αποδίδει ως indices συνεχόμενες αριθμούς που είναι η θέση ενός tile στο tile set, *αριστερά προς δεξιά, πάνω προς κάτω*.
4. **Υλοποιήστε ένα custom parsing function** για το text format του map editor (υλοποίηση της ReadTextMap(...), και χρησιμοποιήστε το ώστε στο unit test να κάνετε load το map.
5. **Υλοποιήστε όλα τα σχετικά με το map** (display function, scrolling, κλπ), πλην του grid.
6. **Κάνετε initialize graphics σε window mode, resolution 640x480**, και ορίστε ως display area όλο το window και ως map view window το ίδιο επίσης.
7. Υλοποιήστε ένα game loop όπως σας έχει δοθεί και ένα App class όπως σας δίδεται παρακάτω.

```
namespace app {
    class App {
        protected:
            Game game;

        public:
            virtual void    Initialise (void) = 0;
            virtual void    Load (void) = 0;
            virtual void    Run (void) { game.MainLoop(); }
            virtual void    RunIteration (void)
                                { game.MainLoopIteration(); }
            Game&            GetGame (void) { return game; }
            const Game&      GetGame (void) const { return game; }
            virtual void    Clear (void) = 0;
            void Main (void) {
                Initialise();
                Load();
                Run();
                Clear();
            }
    };
}

<say this is the main function your library requires> {
    App app;
    app.Main();
}
```

8. **Υλοποιήστε** input logic χρησιμοποιώντας πληκτρολόγιο ή και mouse για να κάνει **interactive scrolling** το map view window με ένα offset (από το πληκτρολόγιο το offset θα είναι fixed, από το mouse το $\langle dx, dy \rangle$ του mouse pointer).
 - Για το mouse να απαιτείται clicked left button για να είναι ενεργοποιημένο το scrolling.

- Από το πληκτρολόγιο τα home / end button θα κάνουν set to view window στο top-left / bottom-right position αντίστοιχα.

Unit 2: Edit, Store, Load, Construct and Use a Grid Map

Στο δεύτερο Unit Test θα υλοποιήσετε όλες τις λειτουργίες για το grid map και θα κάνετε έναν ενδελεχή έλεγχο όλων των λειτουργιών. Ειδικότερα, θα κάνετε τα εξής:

1. **Να παράγετε το grid απευθείας από τα tiles** (δοκιμάστε τόσο την *ComputeTileGridBlocks1* όσο και την *ComputeTileGridBlocks2*), επιλέγοντας επίσης έναν σύνολο από indices του tile set τα οποία θα θεωρηθούν ως empty (να παρέχονται με κάποιο τρόπο στον υπολογισμό όπως θέλετε, αποδεκτό να είναι *constant array μέσα στον κώδικα*). Υλοποιήστε αυτή την επεξεργασία το ως ένα ξεχωριστό εκτελέσιμο στο ίδιο folder με παρόν unit test το οποίο παράγει το grid (*name it as needed, it is now a development-time tool*).
2. **Ενσωματώστε στο unit test τη λειτουργικότητα του προηγούμενου App** με κατάλληλο τρόπο, αυτό σημαίνει ότι εάν δεν κάνετε τίποτε άλλο, αυτό το unit test τρέχει ακριβώς σαν να ήταν το προηγούμενο.
3. **Προσθέστε στο loading function και το loading του grid** (προγραμματιστικά είτε μέσω derivation από το App subclass του unit test 1 ή μέσω composition κάνοντας field το προηγούμενο App subclass - εσείς επιλέγετε).
4. **Για τον έλεγχο του grid θα υλοποιήσετε ένα movable rectangle**, με διαστάσεις τις επιλογής σας, το οποίο θα τοποθετήσετε σε μία αρχική θέση στο terrain που δεν αντιστοιχεί σε solid grid elements, και θα το μετακινείτε interactively με input control (επιλέξτε τα input bindings). Η μετακίνηση θα ελέγχεται μέσω των filter functions (*που πρέπει να συμπληρωθούν και για vertical direction*).
 - ο **Προσθέστε στο display function και τη ζωγραφική του rectangle** (με χρήση rectangle drawing του library, όχι ως bitmap)
 - ο **Προσοχή, δεν πρέπει να πειράξετε το αυθεντικό unit test**. Αυτό σημαίνει ότι θα πρέπει να τροποποιήσετε το Game interface ώστε να επιτρέπεται *list of functors*, σε όλες τις περιπτώσεις, με επιλογή για add first or add last.
5. Δοκιμάστε και ελέγξτε την κίνηση του rectangle σε όλα τα σημεία του terrain. Έτσι θα έχετε μία εικόνα για το πόσο ικανοποιητική είναι η ακρίβεια του grid.
 - ο **Να υποστηρίξετε interactive switching μέσω ενός toggle keystroke** (επιλογής σας), μεταξύ ενός rectangle outline και ενός filled rectangle, ώστε να έχετε καλύτερη εικόνα για τα όρια κίνησης.

Unit 3: Implement Layers, Animation and Sprites

Σε αυτό το Unit Test θα πρέπει να γίνει η υλοποίηση των tile layers ως ξεχωριστές κλάσεις καθώς και η υλοποίηση του animation system και των sprites. Η εργασία βασίζεται σε μέρος του παιχνιδιού [Zelda II: The Adventure of Link](#) και συγκεκριμένα διαδραματίζεται στο [Parapa Palace](#) (1ο palace του παιχνιδιού).

Τα βήματα είναι τα εξής:

1. **Υλοποίηση των *multiple layers*.** Σε αυτή τη φάση, για λόγους early testing, μπορείτε ακόμη να χρησιμοποιήσετε το top view terrain, αλλά να προσθέσετε και ένα foreground layer. λόγω της δομής του top view terrain, το circular background δεν έχει νόημα.
 - Στη συνέχεια **υλοποιήστε ένα κλασικό front view terrain**. Για το σκοπό αυτό σας δίδεται στο *UnitTest3Media* folder ένα μέρος από το tile set (sprite sheet) του Zelda II.
 - Προσθέστε είτε ένα circular background με τον τοίχο από τούβλα και του πυλώνες που να μοιάζει με το εσωτερικού του palace στο αυθεντικό παιχνίδι, ενώ να εφαρμόζετε scrolling ως ένα fraction (configurable) του action layer, ή ένα horizontal layer (επειδή είναι πολύ απλό το background του Zelda II και δε θυσιάζονται πολλά tiles – προτεινόμενη λύση, διαστάσεις ίδιες με το action layer).
2. **Υλοποίηση του animation system και των sprites.** Από το σημείο αυτό η υλοποίηση του test θα αφορά μόνο το Zelda II terrain.
 - Ειδικά για το animation system θα προσθέσετε και τον νέο κώδικα από το lecture για το Animation, κεφάλαιο Timing στο τέλος της διάλεξης.
3. **Υλοποίηση του terrain με τον tile editor και loading.** Το terrain του Zelda II δεν έχει συνεχόμενη ροή, αλλά αν και κατασκευάζεται ως ένα layer. Αυτό σημαίνει ότι υπάρχουν δύο σκηνές σε ένα layer, τα όρια των οποίων είναι γνωστά και απαγορεύεται κατά τη διάρκεια του gameplay να γίνεται scrolling εκτός των ορίων της παρούσας σκηνής. Οι παρακάτω εικόνες αποτυπώνουν τις ιδιαιτερότητες του terrain.
4. Υλοποίηση της νέας λογικής για scrolling που εξαρτάται μόνο από τη θέση του Link στο terrain. Για το λόγο αυτό θα πρέπει να χρησιμοποιήσετε τη θέση του Link για τον υπολογισμό της θέσης του view window. Επίσης θα πρέπει να προσθέσετε στους ελέγχους των *filter scroll methods* που εξαρτάται μόνο από τα layer bounds (σε pixel dimensions) με προαιρετικό έλεγχο από ένα scene rectangle το οποίο γίνεται set δυναμικά (precondition: το scene rectangle εμπεριέχεται στο tile layer και είναι πάντα τουλάχιστον όσο το view window).
5. Υλοποίηση του Link character, των pipes και της μετάβασης μεταξύ των σκηνών χωρίς να περιλαμβάνεται καθόλου η βασική δράση του παιχνιδιού.

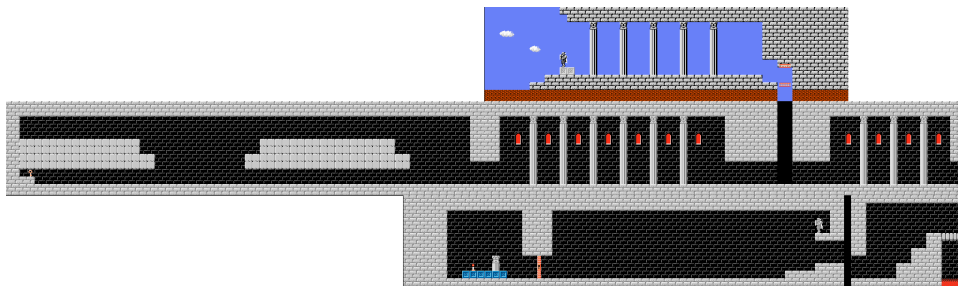


Figure 1: Part of the terrain, showing ground (bricks), platforms (floating bricks) and an elevator.

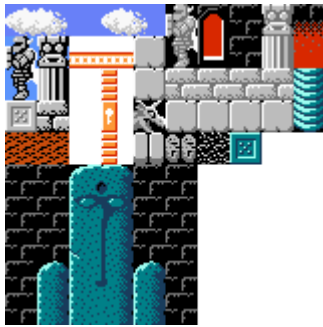


Figure 2: Basic tile set used for the terrain (Parapa palace).

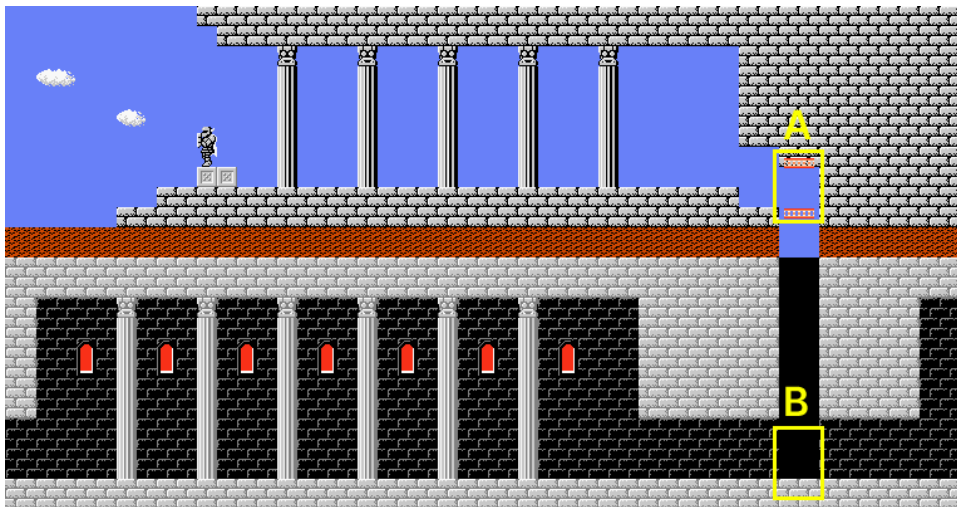


Figure 3: A part of the game terrain illusion as actually perceived by the player. Entering the elevator A and pressing the DOWN key, will lead Link into the palace (there are other points where the corresponding logic applies) at point B.

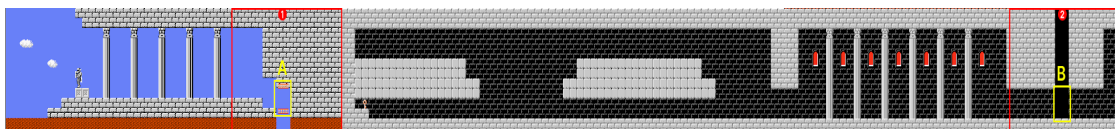


Figure 4: Game terrain as actually designed with scene transitions. When Link enters the elevator, at point A, the viewing window is at the position shown by the red rectangle 1. Entering the elevator will cause view window to scroll to position 2.

Link character

Ο Link, με την χρήση των πλήκτρων 'ΑΡΙΣΤΕΡΑ' και 'ΔΕΞΙΑ' είναι σε θέση να προχωρήσει προς τα αριστερά ή τα δεξιά, αντίστοιχα. Επίσης, με την χρήση του πλήκτρου 'Α', ο Link πραγματοποιεί ένα μικρό άλμα (ύψους 2 tiles) και είναι σε θέση να μετακινηθεί ενώ είναι στον αέρα. Παρακάτω, παρουσιάζονται οι δύο βασικές κινήσεις με τις οποίες επιτυγχάνεται η μάχη κατά την διάρκεια του παιχνιδιού.

Slash

Όταν ο παίκτης πατήσει το 'B' πλήκτρο, ο Link επιτίθεται χρησιμοποιώντας το ξίφος του. Το ξίφος θα εκφορτώσει μια ενεργειακή έκρηξη μικρής εμβέλειας **εάν η ζωή του Link είναι γεμάτη**.



Figure 5: **Link's Slash**. Link uses his sword to attack.

Crouching Stance

Όταν ο παίκτης **πατήσει και κρατήσει πατημένο** το 'ΚΑΤΩ' πλήκτρο, ο Link σκύβει και χρησιμοποιεί την ασπίδα του για να αποκρούσει επιθέσεις από χαμηλά. Να σημειωθεί ότι, σε αυτή την κατάσταση, ο Link παραμένει ευάλωτος σε επιθέσεις που δέχεται από εχθρούς που βρίσκονται πίσω του.



Figure 6: **Link's Crouching Stance**. Link uses his shield to block attacks from the front, but remains vulnerable to attacks from behind.

Να σημειωθεί ότι υποστηρίζεται και συνδυασμός των παραπάνω κινήσεων (π.χ. Crouching Slash).

Elevator and Scene Transition

1. Ο ανελκυστήρας (elevator) είναι φτιαγμένος από solid tiles.
2. Πρέπει να δημιουργηθεί ένα sprite για κάθε ανελκυστήρα, το οποίο να τοποθετηθεί ακριβώς πάνω από τον tile-based ανελκυστήρα (οπτικά πρέπει να μην είναι ορατή η διαφορά τους). Θα χρειαστούν προσαρμογές σε επίπεδο pixels προκειμένου να επιτευχθεί το επιθυμητό αποτέλεσμα.
3. Το παραπάνω αρκεί να υλοποιηθεί μόνο για το κάτω μέρος κάθε ανελκυστήρα, μιας και αυτό είναι το σημείο όπου στέκεται ο Link.
4. Να υπολοιπηθεί η λειτουργικότητα που ελέγχει αν ο Link είναι στο ανελκυστήρα όσο αυτός είναι στην κορυφή, στην μέση και στο τέλος όπως φαίνεται στο Figure 4.

5. Στην συνέχεια να γίνει η προσθήκη για να εισάγεται ένας ανελκυστήρας που να λειτουργεί ως εξής:
 - a. απενεργοποίηση κάθε κίνησης του Link και εφαρμογή απευθείας μετατόπισης
 - b. εφαρμογή animation με την κίνηση του ανελκυστήρα και του Link έτσι ώστε, στο τέλος την κίνησης, να ακουμπάει ο ανελκυστήρας στην άκρη του τρέχοντος παραθύρου
 - c. Προσθήκη ενός on finish functor για την εφαρμογή της απευθείας μετατόπισης στην επιθυμητή σκηνή

The Full Project

Υλοποιήστε το Zelda II The Adventure of Link και συγκεκριμένα την περιπέτεια του Link στο Parapa palace. Ακολουθούν επιπλέον οδηγίες ανά περίπτωση. Πρόκειται για γενικές οδηγίες που μπορούν να εφαρμοστούν σε πολλά παιχνίδια και δεν αφορούν αποκλειστικά το Zelda II.

Placing Powerups, Items and Bonuses

Στο παιχνίδι υπάρχει πληθώρα από αντικείμενα με τα οποία ο Link μπορεί να αλληλοεπιδράσει. Τα αντικείμενα αυτά επηρεάζουν τα στατιστικά του Link και για αυτό δεν μπορούν να υλοποιηθούν ως απλά tiles. Θα πρέπει να γίνει η υλοποίησή τους ως κανονικά sprites, έτσι ώστε να επιτρέπουν τον έλεγχο για collision και την αλληλεπίδραση με τον Link. Να σημειωθεί ότι ο Link πρέπει να χρησιμοποιήσει το ξίφος του για να συλλέξει ένα αντικείμενο και η επίδραση του αντικειμένου είναι άμεση (χωρίς ενδιάμεση αποθήκευση σε κάποιο inventory). Προκειμένου να τοποθετηθούν σωστά στον χώρο, μπορεί να εφαρμοστεί η παρακάτω τεχνική:

1. Δημιουργήστε το terrain κανονικά, χρησιμοποιώντας κάθε tile που θέλετε. Με άλλα λόγια, χρησιμοποιήστε tiles που αντιστοιχούν σε αντικείμενα τοποθετώντας τα στα σημεία που θα βρίσκονται κατά την διάρκεια του παιχνιδιού.
 - Με αυτό τον τρόπο θα βλέπετε το terrain ακριβώς με την μορφή που θα είναι και κατά την εκτέλεση του παιχνιδιού, κάτι που συμβαίνει οπτικά όταν τα αντικείμενα του παιχνιδιού έχουν διαστάσεις όπως τα tiles (όπως και στην περίπτωση του Zelda II).
2. Καταγράψτε την θέση ή τις συντεταγμένες αυτών των tiles έτσι ώστε να μπορείτε να τα αναγνωρίζετε μέσα στον tile map. Για παράδειγμα, καταγράψτε ότι το αντικείμενο 'Point Bag' βρίσκεται στην θέση x:0 και y:9 μέσα στον tile set.
3. Αμέσως μετά την ολοκληρωμένη φόρτωση του terrain:
 - αντικαταστήστε κάθε tile που αντιστοιχεί σε κάποιο αντικείμενο με το tile που χρησιμοποιείτε για background replace
 - καταγράψτε την τοποθεσία κάθε τέτοιου tile instance έτσι ώστε να τοποθετήσετε το αντίστοιχο sprite. Για παράδειγμα, αν το αντικείμενο 'Point Bag' έχει τοποθετηθεί στην θέση (row, col) θα αντικατασταθεί με κάποιο background non-solid tile και θα δημιουργηθεί ένα 'Point Bag' sprite στην συγκεκριμένη θέση
 - θα χρειαστείτε την υλοποίηση της συνάρτησης **SetGridTileBlock**

Με αυτόν τον τρόπο, το τελικό αποτέλεσμα θα μοιάζει ακριβώς με αυτό που σχεδιάσατε αλλά θα έχει αντικείμενα με τα οποία επιτρέπεται η αλληλεπίδραση αντί για απλά, στατικά, tiles σε όσα σημεία είναι απαραίτητο.

Placing NPCs

Προκειμένου να τοποθετηθούν οι εχθροί (NPCs) στο terrain, In order to orchestrate the enemies (NPCs) inside the terrain, σας συμβουλεύουμε να:

1. Έχετε ειδικά tiles που να αντιπροσωπεύουν τους NPCs (ακόμα και custom)
2. Για κάποιους εχθρούς, το sprite frame δεν τοποθετείτε ακριβώς μέσα σε ένα tile, για αυτό προτείνεται η παρακάτω μέθοδος:
 - a. Δημιουργήστε ένα extra tile που να αντιπροσωπεύει τον εχθρό και προσθέστε το στο tile set
 - b. Τοποθετήστε το tile στην κάτω-αριστερά τοποθεσία στο tile map όπου ο εχθρός πρέπει να βρίσκεται
 - c. Όταν θα δημιουργηθεί το αντίστοιχο sprite θα πρέπει να το επανατοποθετήσετε στην θέση που αναμένετε.
3. Χρησιμοποιήστε την τεχνική που παρουσιάστηκε προηγούμενος έτσι ώστε να αντικαταστήσετε κάθε NPC-representing tiles με ένα background non-solid tile κατά την φόρτωση και τοποθετήστε το πραγματικό sprite που αντιστοιχεί στον NPC.

Supporting Pause / Resume

1. Επεκτείνετε το `Game` έτσι ώστε να περιλαμβάνει pause/resume:

```
class Game {  
  
    Action pauseResume;  
    bool isPaused = false;  
    uint64_t pauseTime = 0;  
  
    void SetOnPauseResume (const Action& f)  
    { pauseResume = f; }  
    void Pause (uint64_t t)  
    { isPaused = true; pauseTime = t; Invoke(pauseResume); }  
    void Resume (void)  
    { isPaused = false; Invoke(pauseResume); pauseTime = 0; }  
    bool IsPaused (void) const  
    { return isPaused; }  
    uint64_t GetPauseTime (void) const  
    { return pauseTime; }  
};
```

2. Τροποποιήστε το game loop iteration για να λαμβάνει υπόψιν του το pause / resume:

```
Render();  
Input();  
if (!IsPaused()) {  
    ProgressAnimations();  
    AI();  
    Physics();  
}
```

```

        CollisionChecking();
        CommitDestructions();
        UserCode();
    }

```

3. Όταν το παιχνίδι είναι σε κατάσταση paused, να εμφανίζεται ένα επιπλέον μήνυμα που να ενημερώνει τον παίκτη κατάλληλα
4. Επεκτείνεται τον [AnimatorManager](#) για να περιλαμβάνει την παρακάτω μέθοδο (time shifting):

```

void TimeShift (unsigned dt) {
    for (auto* a : running)
        a->TimeShift(dt);
}

```

5. Προσθέστε μία εντολή εισόδου (όπως το *ESC* key) το οποίο έχει τον ρόλο της υλοποίησης του pause / resume ως εξής:

```

// initially
void InstallPauseResumeHandler (Game& game) {
    game.SetOnPauseResume(
        [&game](void) {
            if (!game.IsPaused()) // just resumed
                AnimatorManager::GetSingleton().TimeShift(
                    GetGameTime() - game.GetPauseTime()
                );
        }
    );
}

// in response to the pause / resume input key
void HandleTogglePauseResume (Game& game) {
    if (game.IsPaused())
        game.Resume();
    else
        game.Pause(GetGameTime());
}

```

Characters





Link

Life & magic system

Ο Link έχει πόντους ζωής που αναπαρίστανται μέσω μίας μπάρας. Όταν ο Link δεχτεί ένα χτύπημα, η ζωή του μειώνεται. Κάθε εχθρός είναι σε θέση να προκαλέσει διαφορετική μείωση της ζωής του Link. Όταν οι πόντοι ζωής του Link τελειώσουν, ο Link χάνει μία ζωή και το παιχνίδι ξεκινάει ξανά. Αξίζει να σημειωθεί ότι, μετά την απώλεια μίας ζωής, ο Link δεν ξεκινάει από την αρχή του level, αλλά από **checkpoints** που είναι τοποθετημένα σε διάφορα σημεία στο level και ενεργοποιούνται αυτόματα μόλις ο Link

φτάσει στο σημείο που έχουν τοποθετηθεί (δεν χρειάζεται αποθήκευση του state όταν ο παίκτης περάσει ένα checkpoint). Μετά την επαναφορά του παιχνιδιού στο checkpoint, όλοι οι εχθροί έχουν επανέλθει και το παιχνίδι συνεχίζει. Με την επαναφορά, πρέπει να πραγματοποιηθεί η σωστή τοποθέτηση του view window στο σημείο του checkpoint. Τέλος, τα checkpoint δεν είναι ορατά από τον παίκτη.

Enemies

Image	Name	Points	Health Point
	Great Palace Bot	10	16
	Wosu	0	8
	Guma	50	64
	Stalfos	30	32

- **Great Palace Bot:** Οι συγκεκριμένοι εχθροί προχωράνε σε ευθεία γραμμή με μικρά άλματα προς τον Link και είναι σε θέση να τον χτυπήσουν με την επαφή.
- **Wosu:** Οι Guma είναι αρκετά απλοί εχθροί. Προχωράνε προς τα μία κατεύθυνση αγνοώντας εντελώς την ύπαρξη του Link. Όταν ακουμπήσουν τον Link, προκύπτει “χτύπημα”.
- **Guma:** Οι συγκεκριμένοι εχθροί είναι σε θέση να απομακρυνθούν ελάχιστα από τον Link καθώς πετάνε μικρά όπλα προς το μέρος του. Ο Link, μπορεί να τραυματιστεί τόσο κατά την επαφή μαζί τους, όσο και από τα όπλα που πετάνε.
- **Stalfos:** Οι Stalfos είναι σκελετοί εξοπλισμένοι με πανοπλία οι οποίοι εμφανίζονται από το ταβάνι όταν πλησιάσει ο Link. Για την εξόντωσή τους, ο Link πρέπει να τους χτυπήσει από χαμηλά μιας και χρησιμοποιούν την ασπίδα τους για να προστατευτούν από επιθέσεις σε κανονική στάση.






Όταν ο Link δεχτεί επίθεση, προκύπτει ένα animation κατά το οποίο ο Link αλλάζει χρώμα και μεταφέρετε λίγο πιο πίσω. Επίσης, αξίζει να σημειωθεί ότι οι εχθροί έχουν μία πιθανότητα να αφήσουν κάποιο αντικείμενο στην θέση τους μόλις πεθάνουν, το οποίο μπορεί να συλλέξει ο Link με την χρήση του ξίφους του.

Power-ups

https://www.zeldadungeon.net/wiki/The_Adventure_of_Link_Items#Pick-Ups

Several power-ups can be found in the mushroom kingdom to help Mario achieve his goal. The minimum power ups that you will have to implement are the following:

Image	Name	Effect
-------	------	--------

	Blue Potion	The Blue Potion restores 16 Magic Points
	Red Potion	The Red Potion restores 128 Magic Points
	Point Bag	The Point Bag contains a number of experience points, depending on which enemy Link defeats. Basic Enemies: 50 Points Stronger Enemies: 200 Points
	Key	Keys are used to unlock any single door. Regular Keys are one-time use, so you will need to find one for every locked door.
	1-Up Doll	A 1-Up Doll will give Link one extra Life.

Το παιχνίδι προσφέρει στον Link την ικανότητα να χρησιμοποιεί διάφορα ξόρκια που μπορούν να τον βοηθήσουν κατά την περιπέτειά του. Για τον σκοπό αυτό, ο Link έχει πόντους μαγείας τους οποίους χρησιμοποιεί για την εφαρμογή των ξορκιών. Κάθε ξόρκι χρησιμοποιεί διαφορετικό πλήθος από πόντους μαγείας. Τέλος, αξίζει να σημειωθεί ότι, στο original παιχνίδι, για να χρησιμοποιήσει ο Link κάποιο ξόρκι έπρεπε πρώτα να το “μάθει” από σοφούς σε διάφορες πόλεις. Για τις ανάγκες που παιχνιδιού μας, ο Link μπορεί να γνωρίζει ήδη ένα σύνολο από ξόρκια (όσο επιθυμείτε εσείς με ελάχιστο πλήθος τα 2 ξόρκια) τα οποία μπορεί να εφαρμόσει με το πάτημα ενός πλήκτρου (διαφορετικού ανά ξόρκι). Παρακάτω παρουσιάζονται τα διαθέσιμα ξόρκια και είστε ελεύθεροι να εμπλουτίσετε την λίστα με δικές σας ιδέες:

Spells

Name	Abilities	Spell Cost
Shield	It reduces the damage Link takes by half.	32
Jump	Allows you to jump twice as high as normal.	48
Life	Heals Link for three squares of energy, provided you have the magic to use it.	70
Fairy	Allows you to reach high places and cross lengthy gaps.	80
Fire	Allows you to reflect magical attacks with the shield, as well as block previously unblockable projectiles like clubs, axes, and fireballs.	120
Reflect	Allows you to reflect magical attacks with the shield, as well as block previously unblockable projectiles like clubs, axes, and fireballs.	120

Thunder	Very expensive spell that destroys all non-boss enemies on screen.	120
---------	--	-----

Αξίζει να σημειωθεί ότι ξόρκια όπως το Shield και Jump, στο original παιχνίδι ήταν ενεργά (μετά την χρήση) για όσο διάστημα ο Link βρισκόταν στο συγκεκριμένο δωμάτιο. Στα πλαίσια αυτής της εργασίας και λόγω του ότι δεν υπάρχουν επιμέρους δωμάτια στην υλοποίηση που προτείνετε, μπορούν να είναι ενεργά για ένα μικρό χρονικό διάστημα (π.χ. 30 δευτερόλεπτα).



Figure 7: Screenshot during the game. Shown is the **magic bar**, **life bar** as well as Link's **hit points**.

Optional Configuration Parameters

Συστήνεται, ένα πλήθος των χαρακτηριστικών του παιχνιδιού, να μπορούν να καθοριστούν από παραμέτρους αποθηκευμένες σε configuration files. Τροποποιώντας τις συγκεκριμένες τιμές, το παιχνίδι μπορεί να επηρεαστεί σε σημείο που να μοιάζει τελείως διαφορετικό. Ακολουθούν μερικές από τις παραμέτρους που μπορείτε να υποστηρίξετε:

- Αρχικό πλήθος ζωών του χαρακτήρα
- Πλήθος πόντων ζωής του χαρακτήρα
- Πλήθος πόντων μαγείας του χαρακτήρα
- Μέγεθος επίθεσης του χαρακτήρα
- Αξία σε πόντους κάθε εχθρού
- Πλήθος πόντων ζωής ανά εχθρό
- Μέγεθος επίθεσης ανά εχθρό
- Πλήθος από εχθρούς
- Ταχύτητα των εχθρών
- Διάρκεια κάθε ξορκιού του Link
- Sound effect files και background music theme

Optional Extras

- Επέκταση του terrain territory

- Επιπλέον πλατφόρμες και εμπόδια
- Damage effects
- More detailed animation films (more frames and better visuals) for Link
- Extra modes for Link (both visually and in terms of actions, like flying)
- Οτιδήποτε επιπλέον μπορεί να εμπλουτίσει την εμπειρία του παιχνιδιού

Resources

- https://en.wikipedia.org/wiki/Zelda_II:_The_Adventure_of_Link
- https://www.zeldadungeon.net/wiki/Zelda_II:_The_Adventure_of_Link
- https://www.zeldadungeon.net/wiki/Parapa_Palace
- https://www.ign.com/wikis/zelda-ii-the-adventure-of-link/Basic_Controls

Ομαδική εργασία -παράδοση

Η εργασία μπορεί να περατωθεί από ομάδα τριών (3) το πολύ ατόμων. Η υλοποίηση της εργασίας είναι υποχρεωτικό να γίνει με τη μέθοδο που διδάσκεται στο μάθημα σε C++, οτιδήποτε άλλο δε θα γίνει δεκτό.

Το ποσοστό της τελικής βαθμολογίας που καταλαμβάνει η εργασία σας είναι 100%. Η παράδοση και εξέταση της εργασίας θα γίνει σε τρεις φάσεις:

Η εξέταση των εργασιών θα γίνει με κοινή παρουσίαση μπροστά στις υπόλοιπες ομάδες, με περίπου 15-20 λεπτά διαθέσιμα ανά ομάδα. Αυτό θα γίνει προς το τέλος της εξεταστικής περιόδου και θα ανακοινωθεί μέσα στον Ιανουάριο.

Επίσης κατά την εξέταση θα παραδώσετε όλο το project όπως έχει φτιαχτεί στο IDE.

Επίσης θα προηγηθεί ξεχωριστή εξέταση των ομάδων με έλεγχο του κώδικα όπου θα φανεί η συμμετοχή των μελών και θα βαθμολογηθούν αντίστοιχα. Όσοι αποδειχθεί ότι δε συμμετείχαν θα λάβουν μηδέν και θα υπάρχει σχετικό penalty στο συνολικό βαθμό της ομάδας.

Φροντίστε η αρχική οθόνη του παιχνιδιού να περιέχει: τα ονόματα της ομάδας σας με λατινικούς χαρακτήρες και κεφάλαια γράμματα καθώς και την ένδειξη “University of Crete \n Department of Computer Science \n CS-454. Development of Intelligent Interfaces and Games \n Term Project, Fall Semester 2022”.